CycloBranch

Generated by Doxygen 1.9.4

1 Introduction 1

2 User's Guide 3

2.1 Main Window and Toolbar . . . 5
2.1.1 Main Window . . . 5
2.1.2 File . . . 5
2.1.3 Search . . . 6
2.1.4 Tools . . . 7
2.1.5 Bookmarks . . . 7
2.1.6 View . . . 7
2.1.7 Help . . . 9
2.1.8 Text Filter . . . 9
2.2 Settings . . . 9
2.2.1 Settings - Dereplication (MS) . . . 9
2.2.1.1 Basic Buttons . . . 10
2.2.2 Search . . . 10
2.2.2.1 Mode . . . 10
2.2.2.2 Maximum Number of Threads . . . 10
2.2.3 Experimental Spectrum/Spectra . . . 11
2.2.3.1 File (CTRL + P) . . . 11
2.2.3.2 Charge . . . 13
2.2.3.3 m/z Error Tolerance . . . 13
2.2.3.4 Minimum Threshold of Relative Intensity . . . 13
2.2.3.5 Minimum Threshold of Absolute Intensity . . . 13
2.2.3.6 m/z Ratio . . . 13
2.2.3.7 Retention Time . . . 13
2.2.3.8 FWHM . . . 13
2.2.4 Isotope Ratios . . . 13
2.2.4.1 54Fe/56Fe Ratio . . . 13
2.2.4.2 60Ni/58Ni Ratio . . . 14
2.2.4.3 65Cu/63Cu Ratio . . . 14
2.2.4.4 66Zn/64Zn Ratio . . . 14
2.2.4.5 68Zn/64Zn Ratio . . . 14
2.2.5 Theoretical Spectrum/Spectra . . . 14
2.2.5.1 Sequence/Compound Database File (CTRL + T) . . . 14
2.2.5.2 Ion Types . . . 14
2.2.5.3 Neutral Losses . . . 15
2.2.5.4 Maximum Number of Combined Losses . . . 16
2.2.5.5 Report Unmatched Theoretical Peaks . . . 16
2.2.5.6 Generate Full Isotope Patterns . . . 16
2.2.5.7 Minimum Number of Isotopic Peaks . . . 16
2.2.5.8 Minimum Number of Spectra . . . 16

2.2.5.9 Minimum Number of Ion Types . . . 16
2.2.5.10 Basic Formula Check . . . 16
2.2.5.11 Calculate FDRs . . . 16
2.2.5.12 Minimum Intensity of Highest Peak in Isotopic Pattern . . . 16
2.2.5.13 Isotope m/z Tolerance . . . 17
2.2.5.14 Isotope Intensity Tolerance . . . 17
2.3 Output Report . . . 17
2.4 Linear Peptides . . . 20
2.4.1 Linear Peptide Detail Window . . . 20
2.4.2 Toolbar for Linear Peptides . . . 21
2.4.3 Linear Sequence Detection . . . 22
2.5 Cyclic Peptides . . . 23
2.5.1 Cyclic Peptide Detail Window . . . 23
2.5.2 Toolbar for Cyclic Peptides . . . 24
2.5.3 Cyclic Sequence Detection . . . 24
2.5.4 Nomenclature of Cyclic Series . . . 24
2.6 Branched Peptides . . . 25
2.6.1 Branched Peptide Detail Window . . . 25
2.6.2 Toolbar for Branched Peptides . . . 26
2.6.3 Branched Sequence Detection . . . 27
2.6.4 Nomenclature of Branched Series . . . 28
2.6.5 Fragmentation Example . . . 29
2.6.6 Branched Siderophores . . . 30
2.7 Branch-cyclic Peptides . . . 31
2.7.1 Branch-cyclic Peptide Detail Window . . . 31
2.7.2 Toolbar for Branch-cyclic Peptides . . . 31
2.7.3 Branch-cyclic Sequence Detection . . . 31
2.7.4 Nomenclature of Branch-cyclic Series . . . 33
2.7.4.1 Linearized Sequences (Derived from a Ring Opening) . . . 33
2.7.4.2 Linearized Sequences (Derived from a Branched Fragment) . . . 34
2.7.4.3 Example of Linearized Sequences of a Branched Fragment . . . 35
2.7.4.4 Example of Linearized Sequences of a Linear Fragment . . . 36
2.7.5 Branch-cyclic Siderophores . . . 36
2.8 Linear Siderophores . . . 36
2.8.1 Linear Nonribosomal Peptide Siderophores . . . 36
2.8.2 Linear Polyketide Detail Window . . . 36
2.8.3 Nomenclature of Linear Polyketide Series . . . 38
2.8.4 Linear Polyketide Sequence Detection . . . 39
2.9 Cyclic Siderophores . . . 40
2.9.1 Cyclic Nonribosomal Peptide Siderophores . . . 40
2.9.2 Cyclic Polyketide Detail Window . . . 40
2.9.3 Nomenclature of Cyclic Polyketide Series . . . 41

2.9.4 Cyclic Polyketide Sequence Detection . . . 42
2.10 Dereplication . . . 42
2.10.1 Dereplication in MS Data . . . 42
2.10.2 Dereplication in LC-MS data . . . 44
2.10.3 Dereplication in MSI Data . . . 47
2.10.4 Databases of Compounds . . . 49
2.11 Building Blocks Editor . . . 49
2.11.1 Building Blocks Editor . . . 49
2.11.1.1 Toolbar . . . 49
2.11.1.2 Text Filter . . . 50
2.11.1.3 Columns . . . 50
2.11.2 Format of Building Block Databases . . . 50
2.11.3 Format of Ketide Building Blocks . . . 51
2.11.4 List of Available Building Block Databases . . . 51
2.12 Sequence/Compound Database Editor . . . 52
2.12.1 Sequence/Compound Database Editor . . . 52
2.12.1.1 Toolbar . . . 53
2.12.1.2 Text Filter . . . 54
2.12.1.3 Columns . . . 54
2.12.2 Format of Sequence Databases . . . 54
2.12.3 List of Available Sequence/Compound Databases . . . 55
2.13 Modifications Editor . . . 57
2.13.1 Modifications Editor . . . 57
2.13.1.1 Toolbar . . . 57
2.13.1.2 Text Filter . . . 58
2.13.1.3 Columns . . . 58
2.13.2 Format of Modification Databases . . . 58
2.13.3 List of Available Modification Databases . . . 58
2.14 Draw Peptide Tool . . . 59
2.14.1 Draw Peptide Tool . . . 59
2.14.1.1 Controls . . . 60
2.15 Summary Table of Matched Peaks . . . 61
2.15.1 Summary Table of Matched Peaks . . . 61
2.15.1.1 Toolbar . . . 61
2.15.1.2 Text Filter . . . 62
2.15.1.3 Columns . . . 62
2.16 Annotated Features Overview . . . 63
2.16.1 Annotated Features Overview . . . 63
2.16.1.1 Toolbar . . . 63
2.16.1.2 Columns . . . 70
2.17 Chromatogram . . . 70
2.17.1 Chromatogram . . . 70

2.17.1.1 Toolbar . . . 71
2.18 Image Fusion . . . 72
2.18.1 Image Fusion . . . 72
2.18.1.1 Toolbar . . . 73
2.18.1.2 Layers . . . 74
2.18.1.3 Vendor-specific Details . . . 78
2.19 Simulate Isotopic Pattern . . . 79
2.19.1 Simulate Isotopic Pattern . . . 79
2.19.1.1 Toolbar . . . 79
2.20 Calculate Molecular Formula . . . 81
2.20.1 Calculate Molecular Formula . . . 81
2.20.1.1 Toolbar . . . 81
2.21 Text Filter . . . 82
2.21.1 Text Filter . . . 82
2.22 Preferences . . . 83
2.22.1 Preferences Dialog . . . 83
2.22.1.1 Ion Types . . . 84
2.22.1.2 Data Processing . . . 87
2.22.1.3 Directories . . . 87
2.22.1.4 Bookmarks . . . 88

3 Tutorials **89**

4 Download **91**

5 Datasets **101**
5.1 List of Datasets . . . 101
5.2 Dataset 1: MS/MS of Nonribosomal Peptides . . . 101
5.3 Dataset 2: MALDI-MS of Siderophores . . . 107
5.4 Dataset 3: MALDI-MSI and DESI-MSI of Siderophores . . . 107
5.5 Dataset 4: LC-MS of Siderophores . . . 108
5.6 Dataset 5: MS/MS of Siderophores . . . 109
5.7 Dataset 6: Correlation of MALDI-MSI with Histology Image . . . 129

6 Publications **131**
6.1 Publication to Cite . . . 131
6.2 Other Publications to Cite . . . 131
6.3 MassSpecBlocks . . . 131
6.4 Other Related Publications . . . 132

7 Settings **135**
7.1 Settings - Compound Search (MS) . . . 135
7.2 Theoretical Spectrum/Spectra . . . 135
7.2.1 Chemical Elements . . . 135

7.2.2 Maximum Number of Combined Elements . . . 136
7.2.3 Basic Formula Check . . . 136
7.2.4 Advanced Formula Check . . . 136
7.2.5 N/O Ratio Check . . . 136
7.2.6 Calculate FDRs . . . 136
7.2.7 Minimum Intensity of Highest Peak in Isotopic Pattern . . . 136
7.2.8 Isotope m/z Tolerance . . . 136
7.2.9 Isotope Intensity Tolerance . . . 136

8 Settings **137**
8.1 Settings - Database Search (MS/MS) . . . 137
8.2 Miscellaneous . . . 137
8.2.1 Disable Precursor Mass Filter . . . 137
8.3 Theoretical Spectrum/Spectra . . . 138
8.3.1 Sequence/Compound Database File (CTRL + T) . . . 138

9 Settings **139**
9.1 Settings - De Novo Sequencing (MS/MS) . . . 139
9.2 Experimental Spectrum/Spectra . . . 139
9.2.1 Peptide Type . . . 139
9.2.2 Scan no. . . . 140
9.2.3 Precursor m/z Ratio . . . 140
9.2.4 Precursor Ion Adduct . . . 140
9.2.5 Charge . . . 140
9.2.6 Precursor m/z Error Tolerance . . . 140
9.2.7 m/z Error Tolerance . . . 140
9.3 Database of Building Blocks . . . 141
9.3.1 Building Blocks Database File (CTRL + B) . . . 141
9.3.2 Maximum Number of Combined Blocks (start / middle / end) . . . 141
9.3.3 Incomplete Paths in De Novo Graph . . . 141
9.3.4 Maximum Cumulative Mass of Blocks . . . 142
9.3.5 N-/C-terminal Modifications File (CTRL + M) . . . 142
9.4 Miscellaneous . . . 142
9.4.1 Disable Precursor Mass Filter . . . 142
9.4.2 Internal Fragments . . . 142
9.4.3 Enable Scrambling . . . 142
9.4.4 Cyclic N-terminus . . . 142
9.4.5 Cyclic C-terminus . . . 142
9.4.6 Regular Order of Ketide Blocks . . . 142
9.5 Theoretical Spectrum/Spectra . . . 142
9.5.1 Score Type . . . 142
9.5.2 Maximum Number of Sequence Candidates Reported . . . 143
9.5.3 Peptide Sequence Tag . . . 143

9.5.4 Ion Types . . . 143
9.5.5 Neutral Losses . . . 143
9.5.6 Maximum Number of Combined Losses . . . 144
9.6 Searched Sequence/Compound . . . 144
9.6.1 Sequence . . . 144
9.6.2 N-terminal Modification . . . 145
9.6.3 C-terminal Modification . . . 145
9.6.4 Branch Modification . . . 145

10 Settings **147**
10.1 Settings - Direct Comparison of Experimental and Theoretical Spectrum (MS/MS) . . . 147
10.2 Searched Sequence/Compound . . . 147
10.2.1 Sequence . . . 147
10.2.2 N-terminal Modification . . . 148
10.2.3 C-terminal Modification . . . 148
10.2.4 Branch Modification . . . 148
10.2.5 Formula . . . 148

11 Tutorials (CycloBranch 1.x) **149**
11.1 List of Tutorials . . . 149
11.2 Tutorial 1: Does a peptide have a cycle ? . . . 149
11.3 Tutorial 2: How to determine a tag ? . . . 149
11.4 Tutorial 3: How to determine a complete sequence ? . . . 150
11.5 Tutorial 4: How to determine a branched sequence ? . . . 150
11.6 Tutorial 5: How to determine terminal modifications ? . . . 150
11.7 Tutorial 6: How to compare a theoretical spectrum with a peaklist ? . . . 150
11.8 Tutorial 7: How to search in a database of MS/MS spectra ? . . . 150
11.9 Tutorial 8: How to search MS spectrum against a database of compounds ? . . . 150
11.10 Tutorial 9: Draw Peptide Tool . . . 150
11.11 Tutorial 10: Identification of Branch-Cyclic Peptide with a Long-sized Branch . . . 151
11.12 Tutorial 11: Identification of Peptide Family using Similarity Search . . . 151
11.13 Tutorial 12: Identification of Siderophores in MS/MS data . . . 151
11.14 Tutorial 13: Identification of Siderophores in MALDI-MS and LC-MS data . . . 151
11.15 Tutorial 14: Identification of Siderophores in Imaging Mass Spectra . . . 151

12 Tutorials (CycloBranch 2.x) **153**
12.1 List of Tutorials . . . 153
12.2 Tutorial 1: Dereplication in MS1 data . . . 153
12.3 Tutorial 2: Dereplication in LC-MS data . . . 156
12.4 Tutorial 3: Identification of unknown compounds in LC-MS data . . . 159
12.5 Tutorial 4: Dereplication in MSI data and fusion with histology image . . . 161
12.6 Tutorial 5: Identification of unknown compounds in MSI data . . . 165
12.7 Tutorial 6: Annotate MS/MS spectrum of a metabolite . . . 167

13 Hierarchical Index 171
13.1 Class Hierarchy . . . 171

14 Class Index 175
14.1 Class List . . . 175

15 File Index 181
15.1 File List . . . 181

16 Class Documentation 185
16.1 cAboutWidget Class Reference . . . 185
16.1.1 Detailed Description . . . 185
16.1.2 Constructor & Destructor Documentation . . . 185
16.1.2.1 cAboutWidget() . . . 185
16.1.3 Member Function Documentation . . . 186
16.1.3.1 closeEvent() . . . 186
16.1.3.2 keyPressEvent() . . . 186
16.2 cBranchCyclicWidget Class Reference . . . 186
16.2.1 Detailed Description . . . 187
16.2.2 Member Function Documentation . . . 187
16.2.2.1 exportToPDF() . . . 187
16.2.2.2 exportToPNG() . . . 187
16.2.2.3 exportToSVG() . . . 187
16.2.2.4 hidePeakLabels() . . . 188
16.2.2.5 initialize() . . . 188
16.2.2.6 paintEvent() . . . 188
16.2.2.7 setReportIsomers() . . . 188
16.3 cBranchedWidget Class Reference . . . 189
16.3.1 Detailed Description . . . 189
16.3.2 Member Function Documentation . . . 189
16.3.2.1 exportToPDF() . . . 189
16.3.2.2 exportToPNG() . . . 190
16.3.2.3 exportToSVG() . . . 190
16.3.2.4 hidePeakLabels() . . . 190
16.3.2.5 initialize() . . . 190
16.3.2.6 paintEvent() . . . 190
16.3.2.7 setReportIsomers() . . . 191
16.4 cBrick Class Reference . . . 191
16.4.1 Detailed Description . . . 192
16.4.2 Member Function Documentation . . . 192
16.4.2.1 empty() . . . 192
16.4.2.2 explodeToIntComposition() . . . 193
16.4.2.3 explodeToStringComposition() . . . 193

16.4.2.4 getAcronyms() . . . 193
16.4.2.5 getAcronymsAsString() . . . 193
16.4.2.6 getAcronymsWithReferencesAsHTMLString() . . . 193
16.4.2.7 getComposition() . . . 194
16.4.2.8 getFirstAcronymAsString() . . . 194
16.4.2.9 getLosses() . . . 194
16.4.2.10 getLossIDs() . . . 194
16.4.2.11 getMass() . . . 194
16.4.2.12 getName() . . . 195
16.4.2.13 getReferences() . . . 195
16.4.2.14 getReferencesAsString() . . . 195
16.4.2.15 getResidueLossType() . . . 195
16.4.2.16 getReverseComposition() . . . 195
16.4.2.17 getSummary() . . . 196
16.4.2.18 getSummaryMap() . . . 196
16.4.2.19 isArtificial() . . . 196
16.4.2.20 load() . . . 196
16.4.2.21 setAcronyms() . . . 196
16.4.2.22 setArtificial() . . . 197
16.4.2.23 setComposition() . . . 197
16.4.2.24 setLosses() . . . 197
16.4.2.25 setLossIDs() . . . 197
16.4.2.26 setMass() . . . 197
16.4.2.27 setName() . . . 198
16.4.2.28 setReferences() . . . 198
16.4.2.29 setResidueLossType() . . . 198
16.4.2.30 setSummary() . . . 198
16.4.2.31 store() . . . 198
16.5 cBricksDatabase Class Reference . . . 199
16.5.1 Detailed Description . . . 200
16.5.2 Member Function Documentation . . . 200
16.5.2.1 calculateSummaryFromComposition() . . . 200
16.5.2.2 checkKetideBlocks() . . . 200
16.5.2.3 getAcronymName() . . . 201
16.5.2.4 getAcronymNameOfTPeptide() . . . 201
16.5.2.5 getMapOfNeutralLosses() . . . 201
16.5.2.6 getMassOfComposition() . . . 202
16.5.2.7 getMassOfNeutralLosses() . . . 202
16.5.2.8 getRealName() . . . 202
16.5.2.9 getSummaryFormulaOfNeutralLosses() . . . 203
16.5.2.10 getTagName() . . . 203
16.5.2.11 getTagNameOfTPeptide() . . . 203

16.5.2.12 load() . . . 204
16.5.2.13 loadFromPlainTextStream() . . . 204
16.5.2.14 nextCombination() . . . 204
16.5.2.15 nextCombinationFast() . . . 205
16.5.2.16 nextCombinationFastLimited() . . . 205
16.5.2.17 operator[]() . . . 206
16.5.2.18 push_back() . . . 206
16.5.2.19 replaceAcronymsByIDs() . . . 206
16.5.2.20 size() . . . 207
16.5.2.21 store() . . . 207
16.5.2.22 storeToPlainTextStream() . . . 207
16.6 cBricksDatabaseProxyModel Class Reference . . . 207
16.6.1 Detailed Description . . . 208
16.6.2 Constructor & Destructor Documentation . . . 208
16.6.2.1 cBricksDatabaseProxyModel() . . . 208
16.6.3 Member Function Documentation . . . 208
16.6.3.1 filterAcceptsRow() . . . 208
16.6.3.2 initialize() . . . 209
16.6.3.3 lessThan() . . . 209
16.6.3.4 setWholeWord() . . . 209
16.7 cBricksDatabaseWidget Class Reference . . . 210
16.7.1 Detailed Description . . . 210
16.7.2 Constructor & Destructor Documentation . . . 210
16.7.2.1 cBricksDatabaseWidget() . . . 210
16.7.3 Member Function Documentation . . . 210
16.7.3.1 applyGlobalPreferences() . . . 211
16.7.3.2 closeEvent() . . . 211
16.7.3.3 keyPressEvent() . . . 211
16.8 cCalibrationChart Class Reference . . . 211
16.8.1 Detailed Description . . . 212
16.8.2 Constructor & Destructor Documentation . . . 212
16.8.2.1 cCalibrationChart() . . . 212
16.8.3 Member Function Documentation . . . 212
16.8.3.1 applyGlobalPreferences() . . . 213
16.8.3.2 closeEvent() . . . 213
16.8.3.3 createTable() . . . 213
16.8.3.4 emitXValueInterval . . . 213
16.8.3.5 keyPressEvent() . . . 214
16.8.3.6 setData() . . . 214
16.8.3.7 setLineParameters() . . . 214
16.9 cCalibrationChartProxyModel Class Reference . . . 215
16.9.1 Detailed Description . . . 215

16.9.2 Constructor & Destructor Documentation . . . 215
16.9.2.1 cCalibrationChartProxyModel() . . . 215
16.9.3 Member Function Documentation . . . 216
16.9.3.1 lessThan() . . . 216
16.10 cCalibrationChartScene Class Reference . . . 216
16.10.1 Detailed Description . . . 217
16.10.2 Constructor & Destructor Documentation . . . 217
16.10.2.1 cCalibrationChartScene() . . . 217
16.10.3 Member Function Documentation . . . 217
16.10.3.1 exportToPDF() . . . 218
16.10.3.2 exportToPNG() . . . 218
16.10.3.3 exportToSVG() . . . 218
16.10.3.4 getMaximumCollectionTime() . . . 218
16.10.3.5 getMaximumConcentration() . . . 218
16.10.3.6 hideLabels() . . . 219
16.10.3.7 mouseMoveEvent() . . . 219
16.10.3.8 mousePressEvent() . . . 219
16.10.3.9 mouseReleaseEvent() . . . 219
16.10.3.10 resizeEvent() . . . 219
16.10.3.11 setData() . . . 220
16.10.3.12 setGraphType() . . . 220
16.10.3.13 setLineParameters() . . . 220
16.10.3.14 showEvent() . . . 221
16.10.3.15 updateXValueInterval . . . 221
16.10.3.16 wheelEvent() . . . 221
16.11 cCandidate Class Reference . . . 221
16.11.1 Detailed Description . . . 224
16.11.2 Constructor & Destructor Documentation . . . 224
16.11.2.1 cCandidate() [1/2] . . . 224
16.11.2.2 cCandidate() [2/2] . . . 225
16.11.3 Member Function Documentation . . . 225
16.11.3.1 calculatePrecursorMassFromBricks() . . . 225
16.11.3.2 calculateSummaryFormulaFromBricks() . . . 225
16.11.3.3 checkKetideBlocks() . . . 226
16.11.3.4 checkKetideSequence() . . . 226
16.11.3.5 compare() . . . 227
16.11.3.6 getAcronymPeptideNameWithHTMLReferences() . . . 227
16.11.3.7 getAcronyms() . . . 227
16.11.3.8 getAcronymsTComposition() . . . 227
16.11.3.9 getBackboneAcronyms() . . . 228
16.11.3.10 getBranchAcronyms() . . . 228
16.11.3.11 getBranchCyclicRotations() . . . 228

16.11.3.12 getBranchEnd() . . . 228
16.11.3.13 getBranchSize() . . . 228
16.11.3.14 getBranchStart() . . . 229
16.11.3.15 getComposition() . . . 229
16.11.3.16 getEndModifID() . . . 229
16.11.3.17 getLeftResidueType() . . . 229
16.11.3.18 getMiddleModifID() . . . 229
16.11.3.19 getName() . . . 230
16.11.3.20 getPath() . . . 230
16.11.3.21 getPathAsString() . . . 230
16.11.3.22 getPermutations() . . . 230
16.11.3.23 getPermutationsOfBranches() . . . 230
16.11.3.24 getRealNameTComposition() . . . 231
16.11.3.25 getRealPeptideName() . . . 231
16.11.3.26 getRevertedTComposition() . . . 231
16.11.3.27 getRightResidueType() . . . 231
16.11.3.28 getRotations() . . . 232
16.11.3.29 getStartIonType() . . . 232
16.11.3.30 getStartModifID() . . . 232
16.11.3.31 getSummaryFormula() . . . 232
16.11.3.32 getTComposition() . . . 232
16.11.3.33 hasEqualTPermutations() . . . 233
16.11.3.34 hasFirstBrickArtificial() . . . 233
16.11.3.35 hasLastBrickArtificial() . . . 233
16.11.3.36 hasLastBrickInvalid() . . . 234
16.11.3.37 hasOnlyArtificialBricks() . . . 234
16.11.3.38 isEqualTo() . . . 234
16.11.3.39 load() . . . 234
16.11.3.40 prepareBranchedCandidates() . . . 235
16.11.3.41 setAcronymPeptideNameWithHTMLReferences() . . . 235
16.11.3.42 setAcronyms() . . . 235
16.11.3.43 setBackboneAcronyms() . . . 235
16.11.3.44 setBranchAcronyms() . . . 236
16.11.3.45 setCandidate() [1/2] . . . 236
16.11.3.46 setCandidate() [2/2] . . . 236
16.11.3.47 setEndModifID() . . . 237
16.11.3.48 setMiddleModifID() . . . 237
16.11.3.49 setName() . . . 237
16.11.3.50 setPath() . . . 237
16.11.3.51 setRealPeptideName() . . . 238
16.11.3.52 setStartIonType() . . . 238
16.11.3.53 setStartModifID() . . . 238

16.11.3.54 setSummaryFormula() . . . 238
16.11.3.55 store() . . . 239
16.12 cCandidateSet Class Reference . . . 239
16.12.1 Detailed Description . . . 239
16.12.2 Member Function Documentation . . . 239
16.12.2.1 getSet() . . . 239
16.12.2.2 size() . . . 240
16.13 cCheckBoxDelegate Class Reference . . . 240
16.13.1 Detailed Description . . . 240
16.13.2 Constructor & Destructor Documentation . . . 240
16.13.2.1 cCheckBoxDelegate() . . . 240
16.13.3 Member Function Documentation . . . 241
16.13.3.1 editorEvent() . . . 241
16.13.3.2 paint() . . . 241
16.14 cChromatogramWindow Class Reference . . . 241
16.14.1 Detailed Description . . . 242
16.14.2 Constructor & Destructor Documentation . . . 242
16.14.2.1 cChromatogramWindow() . . . 242
16.14.3 Member Function Documentation . . . 243
16.14.3.1 applyGlobalPreferences() . . . 243
16.14.3.2 calculateAvgSpectrum . . . 243
16.14.3.3 closeEvent() . . . 243
16.14.3.4 doubleClickedScanIDSignal . . . 243
16.14.3.5 emitRetentionTimeInterval . . . 243
16.14.3.6 emitScanIDInterval . . . 244
16.14.3.7 keyPressEvent() . . . 244
16.14.3.8 recalculateTICChromatogram() . . . 244
16.14.3.9 setAbsoluteIntensityEnabled() . . . 244
16.14.3.10 setPeakShape() . . . 245
16.15 cChromatogramWindowWidget Class Reference . . . 245
16.15.1 Detailed Description . . . 246
16.15.2 Constructor & Destructor Documentation . . . 246
16.15.2.1 cChromatogramWindowWidget() . . . 246
16.15.3 Member Function Documentation . . . 246
16.15.3.1 chromatogramWidgetDoubleClicked . . . 246
16.15.3.2 exportToPDF() . . . 247
16.15.3.3 exportToPNG() . . . 247
16.15.3.4 exportToSVG() . . . 247
16.15.3.5 mouseDoubleClickEvent() . . . 247
16.15.3.6 mouseMoveEvent() . . . 247
16.15.3.7 mousePressEvent() . . . 248
16.15.3.8 mouseReleaseEvent() . . . 248

16.15.3.9 recalculateTICChromatogram() . . . 248
16.15.3.10 resizeEvent() . . . 248
16.15.3.11 showEvent() . . . 248
16.15.3.12 updateRetentionTimeInterval . . . 249
16.15.3.13 updateScanIDInterval . . . 249
16.15.3.14 wheelEvent() . . . 249
16.16 cComboBoxDelegate Class Reference . . . 249
16.16.1 Detailed Description . . . 250
16.16.2 Constructor & Destructor Documentation . . . 250
16.16.2.1 cComboBoxDelegate() . . . 250
16.16.3 Member Function Documentation . . . 250
16.16.3.1 createEditor() . . . 250
16.16.3.2 paint() . . . 250
16.16.3.3 setEditorData() . . . 251
16.16.3.4 setModelData() . . . 251
16.17 cCoordinateInfo Struct Reference . . . 251
16.17.1 Detailed Description . . . 252
16.18 cCyclicWidget Class Reference . . . 252
16.18.1 Detailed Description . . . 253
16.18.2 Member Function Documentation . . . 253
16.18.2.1 exportToPDF() . . . 253
16.18.2.2 exportToPNG() . . . 253
16.18.2.3 exportToSVG() . . . 253
16.18.2.4 hidePeakLabels() . . . 253
16.18.2.5 initialize() . . . 254
16.18.2.6 paintEvent() . . . 254
16.18.2.7 setReportIsomers() . . . 254
16.19 cDefaultNeutralLosses Class Reference . . . 254
16.19.1 Detailed Description . . . 254
16.19.2 Member Function Documentation . . . 255
16.19.2.1 getNeutralLosses() . . . 255
16.20 cDefineCalibrationWidget Class Reference . . . 255
16.20.1 Detailed Description . . . 256
16.20.2 Member Function Documentation . . . 256
16.20.2.1 closeEvent() . . . 256
16.20.2.2 getData() . . . 256
16.20.2.3 keyPressEvent() . . . 257
16.20.2.4 load() . . . 257
16.20.2.5 prepareWidget() . . . 257
16.20.2.6 store() . . . 257
16.21 cDeNovoGraph Class Reference . . . 258
16.21.1 Detailed Description . . . 258

16.21.2 Member Function Documentation . . . 258
16.21.2.1 connectEdgesFormingPathsNotFinishingInPrecursor() . . . 259
16.21.2.2 connectEdgesFormingPathsNotStartingFromFirstNode() . . . 259
16.21.2.3 createGraph() . . . 259
16.21.2.4 getBrickDatabaseWithCombinations() . . . 259
16.21.2.5 getNumberOfEdges() . . . 260
16.21.2.6 initialize() . . . 260
16.21.2.7 operator[]() . . . 260
16.21.2.8 printGraph() . . . 260
16.21.2.9 printPaths() . . . 261
16.21.2.10 removeEdgesFormingPathsNotFinishingInPrecursor() . . . 261
16.21.2.11 removeEdgesFormingPathsNotStartingFromFirstNode() . . . 261
16.21.2.12 removePathsWhichCanBeSubstitutedByLongerPath() . . . 261
16.21.2.13 startGraphReader() . . . 262
16.22 cDeNovoGraphNode Class Reference . . . 262
16.22.1 Detailed Description . . . 263
16.22.2 Member Function Documentation . . . 263
16.22.2.1 addIonAnnotation() . . . 263
16.22.2.2 checkIonAnnotation() . . . 263
16.22.2.3 getIntensity() . . . 264
16.22.2.4 getMZRatio() . . . 264
16.22.2.5 insertEdge() . . . 264
16.22.2.6 insertTempEdge() . . . 264
16.22.2.7 operator[]() . . . 264
16.22.2.8 removeEdge() . . . 265
16.22.2.9 setIntensity() . . . 265
16.22.2.10 setMZRatio() . . . 265
16.22.2.11 size() . . . 265
16.23 cDrawPeptideWidget Class Reference . . . 266
16.23.1 Detailed Description . . . 266
16.23.2 Constructor & Destructor Documentation . . . 266
16.23.2.1 cDrawPeptideWidget() . . . 266
16.23.3 Member Function Documentation . . . 267
16.23.3.1 closeEvent() . . . 267
16.23.3.2 keyPressEvent() . . . 267
16.23.3.3 sendInsert . . . 267
16.23.3.4 sendSequence . . . 267
16.23.3.5 sendTag . . . 268
16.23.3.6 setSequence() . . . 268
16.24 cEdge Struct Reference . . . 268
16.24.1 Detailed Description . . . 269
16.25 cEnvelopeSummaryTableKeyMS_comp Struct Reference . . . 269

16.25.1 Detailed Description . . . 269
16.25.2 Member Function Documentation . . . 269
16.25.2.1 operator()() . . . 269
16.26 cEventFilter Class Reference . . . 270
16.26.1 Detailed Description . . . 270
16.26.2 Member Function Documentation . . . 270
16.26.2.1 eventFilter() . . . 270
16.27 cExportDialog Class Reference . . . 270
16.27.1 Detailed Description . . . 271
16.27.2 Constructor & Destructor Documentation . . . 271
16.27.2.1 cExportDialog() . . . 271
16.28 cFindDialog Class Reference . . . 271
16.28.1 Detailed Description . . . 272
16.28.2 Constructor & Destructor Documentation . . . 272
16.28.2.1 cFindDialog() . . . 272
16.29 cFormulaPredictor Class Reference . . . 272
16.29.1 Detailed Description . . . 272
16.29.2 Constructor & Destructor Documentation . . . 273
16.29.2.1 cFormulaPredictor() . . . 273
16.29.3 Member Function Documentation . . . 273
16.29.3.1 closeEvent() . . . 273
16.29.3.2 keyPressEvent() . . . 273
16.30 cFormulaPredictorRowItem Struct Reference . . . 273
16.30.1 Detailed Description . . . 274
16.31 cFormulaPredictorThread Class Reference . . . 274
16.31.1 Detailed Description . . . 274
16.31.2 Member Function Documentation . . . 274
16.31.2.1 initialize() . . . 275
16.32 cFragmentIons Class Reference . . . 275
16.32.1 Detailed Description . . . 276
16.32.2 Constructor & Destructor Documentation . . . 276
16.32.2.1 cFragmentIons() . . . 276
16.32.3 Member Function Documentation . . . 276
16.32.3.1 load() . . . 276
16.32.3.2 operator[]() . . . 276
16.32.3.3 recalculateFragments() . . . 277
16.32.3.4 store() . . . 277
16.33 cFragmentIonsListWidget Class Reference . . . 277
16.33.1 Detailed Description . . . 278
16.33.2 Constructor & Destructor Documentation . . . 278
16.33.2.1 cFragmentIonsListWidget() . . . 278
16.33.3 Member Function Documentation . . . 278

16.33.3.1 getList() . . . 278
16.34 cFragmentIonType Struct Reference . . . 278
16.34.1 Detailed Description . . . 279
16.34.2 Member Function Documentation . . . 279
16.34.2.1 load() . . . 279
16.34.2.2 store() . . . 279
16.35 cGlobalPreferences Class Reference . . . 279
16.35.1 Detailed Description . . . 281
16.36 cGraphReaderThread Class Reference . . . 281
16.36.1 Detailed Description . . . 281
16.36.2 Member Function Documentation . . . 281
16.36.2.1 initialize() . . . 281
16.37 cGraphWidget Class Reference . . . 282
16.37.1 Detailed Description . . . 283
16.37.2 Member Function Documentation . . . 283
16.37.2.1 closeEvent() . . . 283
16.37.2.2 getHTML() . . . 283
16.37.2.3 keyPressEvent() . . . 283
16.37.2.4 load() . . . 283
16.37.2.5 setHTML() . . . 283
16.37.2.6 store() . . . 284
16.37.2.7 updateView() . . . 284
16.38 cHTMLDelegate Class Reference . . . 284
16.38.1 Detailed Description . . . 285
16.38.2 Constructor & Destructor Documentation . . . 285
16.38.2.1 cHTMLDelegate() . . . 285
16.38.3 Member Function Documentation . . . 285
16.38.3.1 paint() . . . 285
16.39 cHTMLExportDialog Class Reference . . . 285
16.39.1 Detailed Description . . . 286
16.39.2 Constructor & Destructor Documentation . . . 286
16.39.2.1 cHTMLExportDialog() . . . 286
16.40 cImageWindow Class Reference . . . 286
16.40.1 Detailed Description . . . 287
16.40.2 Constructor & Destructor Documentation . . . 287
16.40.2.1 cImageWindow() . . . 287
16.40.3 Member Function Documentation . . . 288
16.40.3.1 activeLayerChanged . . . 288
16.40.3.2 addLayer() . . . 288
16.40.3.3 applyGlobalPreferences() . . . 288
16.40.3.4 closeEvent() . . . 288
16.40.3.5 doubleClickedSpectrumIDSignal . . . 289

16.40.3.6 keyPressEvent() . . . 289
16.40.3.7 keyReleaseEvent() . . . 289
16.40.3.8 layerChanged . . . 289
16.40.3.9 setDefaultValues() . . . 290
16.40.3.10 updateSummaryPeaksTableFilter . . . 290
16.41 cImageWindowImportDialog Class Reference . . . 290
16.41.1 Detailed Description . . . 291
16.41.2 Constructor & Destructor Documentation . . . 291
16.41.2.1 cImageWindowImportDialog() . . . 291
16.41.3 Member Function Documentation . . . 291
16.41.3.1 getFormValues() . . . 291
16.41.3.2 setFormValues() . . . 292
16.42 cImageWindowTabElectron Class Reference . . . 292
16.42.1 Detailed Description . . . 292
16.42.2 Constructor & Destructor Documentation . . . 292
16.42.2.1 cImageWindowTabElectron() . . . 293
16.42.3 Member Function Documentation . . . 293
16.42.3.1 getFormValues() . . . 293
16.42.3.2 setFormValues() . . . 293
16.43 cImageWindowTabOptical Class Reference . . . 293
16.43.1 Detailed Description . . . 294
16.43.2 Constructor & Destructor Documentation . . . 294
16.43.2.1 cImageWindowTabOptical() . . . 294
16.43.3 Member Function Documentation . . . 294
16.43.3.1 getFormValues() . . . 294
16.43.3.2 setFormValues() . . . 294
16.44 cImageWindowWidget Class Reference . . . 295
16.44.1 Detailed Description . . . 296
16.44.2 Member Function Documentation . . . 296
16.44.2.1 getImage() . . . 297
16.44.2.2 goToMicroscopyPosition() . . . 297
16.44.2.3 imageWidgetDoubleClicked . . . 297
16.44.2.4 mouseDoubleClickEvent() . . . 297
16.44.2.5 mouseMoveEvent() . . . 297
16.44.2.6 mousePressEvent() . . . 298
16.44.2.7 mouseReleaseEvent() . . . 298
16.44.2.8 resizeEvent() . . . 298
16.44.2.9 setCorrelationValues() . . . 298
16.44.2.10 setDefaultMaxXY() . . . 299
16.44.2.11 setFilterOptions() . . . 299
16.44.2.12 setHistologyImage() . . . 299
16.44.2.13 setHistologyPosition() . . . 300

16.44.2.14 setKeepAspectRatio() . . . 300
16.44.2.15 setMicroscopyImage() . . . 300
16.44.2.16 setMicroscopyPosition() . . . 300
16.44.2.17 setNavigationLayer() . . . 301
16.44.2.18 setOpticalImage() . . . 301
16.44.2.19 setSelectedRegion() . . . 301
16.44.2.20 updateCorrelation . . . 302
16.44.2.21 updateFilter . . . 302
16.44.2.22 updateHistologyPosition . . . 302
16.44.2.23 updateMicroscopyPosition . . . 303
16.44.2.24 updateRuler . . . 303
16.44.2.25 wheelEvent() . . . 303
16.45 cImzML Class Reference . . . 304
16.45.1 Detailed Description . . . 304
16.45.2 Member Function Documentation . . . 304
16.45.2.1 getItems() . . . 304
16.45.2.2 hasProfileSpectra() . . . 304
16.45.2.3 parse() . . . 305
16.45.2.4 updateRawDataToPeakList() . . . 305
16.45.2.5 use64BitIntensityPrecision() . . . 305
16.45.2.6 use64BitMzPrecision() . . . 306
16.45.2.7 write() . . . 306
16.46 cImzMLItem Struct Reference . . . 306
16.46.1 Detailed Description . . . 307
16.47 cIonLabel Struct Reference . . . 307
16.47.1 Detailed Description . . . 307
16.48 cIonSummaryTableKeyMS_comp Struct Reference . . . 307
16.48.1 Detailed Description . . . 307
16.48.2 Member Function Documentation . . . 307
16.48.2.1 operator()() . . . 308
16.49 cIonType Struct Reference . . . 308
16.49.1 Detailed Description . . . 308
16.49.2 Member Function Documentation . . . 309
16.49.2.1 load() . . . 309
16.49.2.2 store() . . . 309
16.50 cIsotopeMap Class Reference . . . 309
16.50.1 Detailed Description . . . 309
16.50.2 Member Function Documentation . . . 309
16.50.2.1 count() . . . 309
16.50.2.2 operator[]() . . . 310
16.51 cIsotopePatternCache Class Reference . . . 310
16.51.1 Detailed Description . . . 311

16.51.2 Constructor & Destructor Documentation . . . 311
16.51.2.1 cIsotopePatternCache() . . . 311
16.51.3 Member Function Documentation . . . 311
16.51.3.1 get() . . . 311
16.51.3.2 put() . . . 311
16.52 cLifMetadata Struct Reference . . . 312
16.52.1 Detailed Description . . . 313
16.53 cLinearWidget Class Reference . . . 313
16.53.1 Detailed Description . . . 313
16.53.2 Member Function Documentation . . . 314
16.53.2.1 exportToPDF() . . . 314
16.53.2.2 exportToPNG() . . . 314
16.53.2.3 exportToSVG() . . . 314
16.53.2.4 hidePeakLabels() . . . 314
16.53.2.5 initialize() . . . 314
16.53.2.6 paintEvent() . . . 315
16.53.2.7 setReportIsomers() . . . 315
16.54 cMainThread Class Reference . . . 315
16.54.1 Detailed Description . . . 316
16.54.2 Constructor & Destructor Documentation . . . 316
16.54.2.1 cMainThread() . . . 317
16.54.3 Member Function Documentation . . . 317
16.54.3.1 addToParallelOutputState() . . . 317
16.54.3.2 enableButtonsHandlingResults . . . 317
16.54.3.3 enableRunButtonAndSettings . . . 317
16.54.3.4 enableStopButton . . . 318
16.54.3.5 isGraphReaderWorking() . . . 318
16.54.3.6 message . . . 318
16.54.3.7 sendParameters . . . 318
16.54.3.8 setGraph . . . 318
16.54.3.9 setGraphReaderIsWorking() . . . 319
16.55 cMainWindow Class Reference . . . 319
16.55.1 Detailed Description . . . 319
16.55.2 Member Function Documentation . . . 319
16.55.2.1 keyPressEvent() . . . 319
16.56 cMainWindowProxyModel Class Reference . . . 320
16.56.1 Detailed Description . . . 320
16.56.2 Constructor & Destructor Documentation . . . 320
16.56.2.1 cMainWindowProxyModel() . . . 320
16.56.3 Member Function Documentation . . . 321
16.56.3.1 filterAcceptsRow() . . . 321
16.56.3.2 initialize() . . . 321

16.56.3.3 lessThan() . . . 321
16.56.3.4 setWholeWord() . . . 322
16.57 cModificationsProxyModel Class Reference . . . 322
16.57.1 Detailed Description . . . 322
16.57.2 Constructor & Destructor Documentation . . . 323
16.57.2.1 cModificationsProxyModel() . . . 323
16.57.3 Member Function Documentation . . . 323
16.57.3.1 filterAcceptsRow() . . . 323
16.57.3.2 initialize() . . . 323
16.57.3.3 lessThan() . . . 324
16.57.3.4 setWholeWord() . . . 324
16.58 cModificationsWidget Class Reference . . . 324
16.58.1 Detailed Description . . . 325
16.58.2 Constructor & Destructor Documentation . . . 325
16.58.2.1 cModificationsWidget() . . . 325
16.58.3 Member Function Documentation . . . 325
16.58.3.1 applyGlobalPreferences() . . . 325
16.58.3.2 closeEvent() . . . 325
16.58.3.3 keyPressEvent() . . . 326
16.59 cMultipleButtonDelegate Class Reference . . . 326
16.59.1 Detailed Description . . . 326
16.59.2 Constructor & Destructor Documentation . . . 326
16.59.2.1 cMultipleButtonDelegate() . . . 326
16.59.3 Member Function Documentation . . . 327
16.59.3.1 displayText() . . . 327
16.59.3.2 editorEvent() . . . 327
16.59.3.3 paint() . . . 327
16.60 cMultipleDatasetsTableProxyModel Class Reference . . . 328
16.60.1 Detailed Description . . . 328
16.60.2 Constructor & Destructor Documentation . . . 328
16.60.2.1 cMultipleDatasetsTableProxyModel() . . . 328
16.60.3 Member Function Documentation . . . 328
16.60.3.1 lessThan() . . . 329
16.61 cMultipleDatasetsTableWidget Class Reference . . . 329
16.61.1 Detailed Description . . . 330
16.61.2 Constructor & Destructor Documentation . . . 330
16.61.2.1 cMultipleDatasetsTableWidget() . . . 330
16.61.3 Member Function Documentation . . . 330
16.61.3.1 applyGlobalPreferences() . . . 330
16.61.3.2 closeEvent() . . . 331
16.61.3.3 getCalibrationChartWidget() . . . 331
16.61.3.4 getDefineCalibrationWidget() . . . 331

16.61.3.5 getPubChemSearchWidget() . . . 331
16.61.3.6 keyPressEvent() . . . 331
16.61.3.7 multipleDatasetsTableRowClicked . . . 332
16.61.3.8 prepareToShow() . . . 332
16.62 cMzML Class Reference . . . 332
16.62.1 Detailed Description . . . 333
16.62.2 Member Function Documentation . . . 333
16.62.2.1 hasProfileSpectra() . . . 333
16.62.2.2 parse() . . . 333
16.63 cNeutralLossesListWidget Class Reference . . . 334
16.63.1 Detailed Description . . . 334
16.63.2 Constructor & Destructor Documentation . . . 334
16.63.2.1 cNeutralLossesListWidget() . . . 334
16.63.3 Member Function Documentation . . . 334
16.63.3.1 addItem() . . . 335
16.63.3.2 getList() . . . 335
16.64 compareStringBySize Struct Reference . . . 335
16.64.1 Detailed Description . . . 335
16.64.2 Member Function Documentation . . . 335
16.64.2.1 operator()() . . . 335
16.65 cParameters Class Reference . . . 336
16.65.1 Detailed Description . . . 341
16.65.2 Member Function Documentation . . . 341
16.65.2.1 calculateNeutralLosses() . . . 341
16.65.2.2 checkAndPrepare() . . . 341
16.65.2.3 checkModifications() . . . 341
16.65.2.4 generateCompounds() . . . 342
16.65.2.5 load() . . . 342
16.65.2.6 prepareLossesAndCompounds() . . . 343
16.65.2.7 printToString() . . . 343
16.65.2.8 setOutputStream() . . . 343
16.65.2.9 store() . . . 343
16.65.3 Member Data Documentation . . . 343
16.65.3.1 reportunmatchedtheoreticalpeaks . . . 343
16.66 cParametersWidget Class Reference . . . 344
16.66.1 Detailed Description . . . 345
16.66.2 Constructor & Destructor Documentation . . . 345
16.66.2.1 cParametersWidget() . . . 345
16.66.3 Member Function Documentation . . . 345
16.66.3.1 applyGlobalPreferences() . . . 345
16.66.3.2 closeEvent() . . . 345
16.66.3.3 getParameters() . . . 345

16.66.3.4 keyPressEvent() . . . 346
16.66.3.5 sendSequenceLine . . . 346
16.66.3.6 setAndRestoreParameters() . . . 346
16.66.3.7 setSequence() . . . 346
16.66.3.8 setTag() . . . 346
16.67 cPatternSimulatorSceneWidget Class Reference . . . 347
16.67.1 Detailed Description . . . 348
16.67.2 Constructor & Destructor Documentation . . . 348
16.67.2.1 cPatternSimulatorSceneWidget() . . . 348
16.67.3 Member Function Documentation . . . 348
16.67.3.1 exportToPDF() . . . 348
16.67.3.2 exportToPNG() . . . 348
16.67.3.3 exportToSVG() . . . 349
16.67.3.4 hidePeakLabels() . . . 349
16.67.3.5 initialize() . . . 349
16.67.3.6 mouseMoveEvent() . . . 349
16.67.3.7 mousePressEvent() . . . 349
16.67.3.8 mouseReleaseEvent() . . . 350
16.67.3.9 resizeEvent() . . . 350
16.67.3.10 showEvent() . . . 350
16.67.3.11 updateMZInterval . . . 350
16.67.3.12 wheelEvent() . . . 350
16.68 cPatternSimulatorWidget Class Reference . . . 351
16.68.1 Detailed Description . . . 351
16.68.2 Constructor & Destructor Documentation . . . 351
16.68.2.1 cPatternSimulatorWidget() . . . 352
16.68.3 Member Function Documentation . . . 352
16.68.3.1 applyGlobalPreferences() . . . 352
16.68.3.2 closeEvent() . . . 352
16.68.3.3 emitMZInterval . . . 352
16.68.3.4 keyPressEvent() . . . 352
16.68.3.5 rawDataStateChangedSignal . . . 353
16.69 cPeak Struct Reference . . . 353
16.69.1 Detailed Description . . . 354
16.69.2 Constructor & Destructor Documentation . . . 355
16.69.2.1 cPeak() . . . 355
16.69.3 Member Function Documentation . . . 355
16.69.3.1 empty() . . . 355
16.69.3.2 equals() . . . 355
16.69.3.3 load() . . . 355
16.69.3.4 operator=() . . . 356
16.69.3.5 store() . . . 356

16.70 cPeakListSeries Class Reference . . . 356
16.70.1 Detailed Description . . . 357
16.70.2 Constructor & Destructor Documentation . . . 357
16.70.2.1 cPeakListSeries() . . . 357
16.70.3 Member Function Documentation . . . 357
16.70.3.1 addPeakList() . . . 357
16.70.3.2 load() . . . 357
16.70.3.3 loadFromBAFStream() . . . 358
16.70.3.4 loadFromIMZMLStream() . . . 358
16.70.3.5 loadFromMGFStream() . . . 359
16.70.3.6 loadFromMZMLStream() . . . 359
16.70.3.7 loadFromPlainTextStream() . . . 359
16.70.3.8 loadFromProfileApexStream() . . . 360
16.70.3.9 loadSpotList() . . . 360
16.70.3.10 operator=() . . . 360
16.70.3.11 operator[]() . . . 360
16.70.3.12 resize() . . . 361
16.70.3.13 size() . . . 361
16.70.3.14 store() . . . 361
16.71 cPeaksList Class Reference . . . 361
16.71.1 Detailed Description . . . 364
16.71.2 Constructor & Destructor Documentation . . . 364
16.71.2.1 cPeaksList() . . . 364
16.71.3 Member Function Documentation . . . 364
16.71.3.1 add() . . . 364
16.71.3.2 attach() . . . 364
16.71.3.3 cropAbsoluteIntenzity() . . . 365
16.71.3.4 cropMaximumMZRatio() . . . 365
16.71.3.5 cropMinimumMZRatio() . . . 365
16.71.3.6 cropRelativeIntenzity() . . . 365
16.71.3.7 equals() . . . 366
16.71.3.8 find() . . . 366
16.71.3.9 getCoordinateX() . . . 366
16.71.3.10 getCoordinateY() . . . 366
16.71.3.11 getMaximumAbsoluteIntensity() . . . 367
16.71.3.12 getMaximumAbsoluteIntensityFromMZInterval() . . . 367
16.71.3.13 getMaximumMZRatio() . . . 367
16.71.3.14 getMaximumRelativeIntensity() . . . 367
16.71.3.15 getMaximumRelativeIntensityFromMZInterval() . . . 368
16.71.3.16 getMinimumMZRatio() . . . 368
16.71.3.17 getRetentionTime() . . . 368
16.71.3.18 getRetentionTimeUnit() . . . 369

16.71.3.19 getTitle() . . . 369
16.71.3.20 load() . . . 369
16.71.3.21 loadFromBAFStream() . . . 369
16.71.3.22 loadFromIBDStream() . . . 369
16.71.3.23 loadFromMGFStream() . . . 370
16.71.3.24 loadFromPlainTextStream() . . . 370
16.71.3.25 maxHighestPeaksInWindow() . . . 370
16.71.3.26 normalizeIntenzity() . . . 370
16.71.3.27 normalizeIntenzityByValue() . . . 371
16.71.3.28 operator=() . . . 371
16.71.3.29 operator[]() . . . 371
16.71.3.30 print() . . . 371
16.71.3.31 reduceIsotopeFormulaDescriptions() . . . 372
16.71.3.32 reducePeakDescriptions() . . . 372
16.71.3.33 remove() . . . 372
16.71.3.34 removeChargeVariants() . . . 372
16.71.3.35 removeIsotopes() . . . 373
16.71.3.36 removeNeutralLoss() . . . 373
16.71.3.37 resize() . . . 373
16.71.3.38 setCoordinates() . . . 373
16.71.3.39 setIsotopeFlags() . . . 374
16.71.3.40 setRetentionTime() . . . 374
16.71.3.41 setRetentionTimeUnit() . . . 374
16.71.3.42 setTitle() . . . 374
16.71.3.43 size() . . . 374
16.71.3.44 sortbyMass() . . . 375
16.71.3.45 store() . . . 375
16.71.3.46 storeToIBDStream() . . . 375
16.72 cPeriodicTableMap Class Reference . . . 375
16.72.1 Detailed Description . . . 376
16.72.2 Member Function Documentation . . . 376
16.72.2.1 count() . . . 376
16.72.2.2 getAdductElements() . . . 376
16.72.2.3 getElements() . . . 376
16.72.2.4 operator[]() . . . 376
16.73 cPreferencesDialog Class Reference . . . 377
16.73.1 Detailed Description . . . 377
16.73.2 Constructor & Destructor Documentation . . . 377
16.73.2.1 cPreferencesDialog() . . . 377
16.73.3 Member Function Documentation . . . 378
16.73.3.1 keyPressEvent() . . . 378
16.73.3.2 resetPreferences() . . . 378

16.73.3.3 setPreferences() . . . 378
16.74 cPreferencesIonTypeDialog Class Reference . . . 378
16.74.1 Detailed Description . . . 379
16.74.2 Constructor & Destructor Documentation . . . 379
16.74.2.1 cPreferencesIonTypeDialog() . . . 379
16.74.3 Member Function Documentation . . . 379
16.74.3.1 getValues() . . . 379
16.74.3.2 setValues() . . . 379
16.75 cPreferencesTabDataProcessing Class Reference . . . 380
16.75.1 Detailed Description . . . 380
16.75.2 Constructor & Destructor Documentation . . . 380
16.75.2.1 cPreferencesTabDataProcessing() . . . 380
16.75.3 Member Function Documentation . . . 381
16.75.3.1 resetPreferences() . . . 381
16.75.3.2 setPreferences() . . . 381
16.76 cPreferencesTabDirectories Class Reference . . . 381
16.76.1 Detailed Description . . . 381
16.76.2 Constructor & Destructor Documentation . . . 382
16.76.2.1 cPreferencesTabDirectories() . . . 382
16.76.3 Member Function Documentation . . . 382
16.76.3.1 resetPreferences() . . . 382
16.76.3.2 setPreferences() . . . 382
16.77 cPreferencesTabIonTypes Class Reference . . . 382
16.77.1 Detailed Description . . . 383
16.77.2 Constructor & Destructor Documentation . . . 383
16.77.2.1 cPreferencesTabIonTypes() . . . 383
16.77.3 Member Function Documentation . . . 383
16.77.3.1 resetPreferences() . . . 383
16.77.3.2 setPreferences() . . . 384
16.78 cPreferencesTabURLs Class Reference . . . 384
16.78.1 Detailed Description . . . 384
16.78.2 Constructor & Destructor Documentation . . . 384
16.78.2.1 cPreferencesTabURLs() . . . 384
16.78.3 Member Function Documentation . . . 385
16.78.3.1 resetPreferences() . . . 385
16.78.3.2 setPreferences() . . . 385
16.79 cPubchemCIDReader Struct Reference . . . 385
16.79.1 Detailed Description . . . 385
16.79.2 Member Function Documentation . . . 386
16.79.2.1 attachVectorOfCIDs() . . . 386
16.79.2.2 getCIDList() . . . 386
16.79.2.3 getCommaSeparatedCIDs() . . . 386

16.79.2.4 getNumberOfCIDs() . . . 386
16.79.2.5 operator()() . . . 387
16.80 cPubchemNameReader Struct Reference . . . 387
16.80.1 Detailed Description . . . 387
16.80.2 Member Function Documentation . . . 387
16.80.2.1 getNameList() . . . 387
16.80.2.2 getReducedNames() . . . 388
16.80.2.3 operator()() . . . 388
16.81 cPubChemSearchWidget Class Reference . . . 388
16.81.1 Detailed Description . . . 389
16.81.2 Member Function Documentation . . . 389
16.81.2.1 closeEvent() . . . 389
16.81.2.2 getHTML() . . . 389
16.81.2.3 keyPressEvent() . . . 389
16.81.2.4 load() . . . 390
16.81.2.5 setHTML() . . . 390
16.81.2.6 store() . . . 390
16.82 cRowItemInfo Struct Reference . . . 390
16.82.1 Detailed Description . . . 391
16.83 cSequence Class Reference . . . 391
16.83.1 Detailed Description . . . 392
16.83.2 Member Function Documentation . . . 392
16.83.2.1 getBranchModification() . . . 392
16.83.2.2 getCTterminalModification() . . . 392
16.83.2.3 getMaximumRetentionTime() . . . 392
16.83.2.4 getMinimumRetentionTime() . . . 393
16.83.2.5 getName() . . . 393
16.83.2.6 getNameWithReferenceAsHTMLString() . . . 393
16.83.2.7 getNTterminalModification() . . . 393
16.83.2.8 getPeptideType() . . . 393
16.83.2.9 getReference() . . . 394
16.83.2.10 getSequence() . . . 394
16.83.2.11 getSummaryFormula() . . . 394
16.83.2.12 isDecoy() . . . 394
16.83.2.13 load() . . . 394
16.83.2.14 setBranchModification() . . . 395
16.83.2.15 setCTterminalModification() . . . 395
16.83.2.16 setDecoy() . . . 395
16.83.2.17 setName() . . . 395
16.83.2.18 setNTterminalModification() . . . 396
16.83.2.19 setPeptideType() . . . 396
16.83.2.20 setReference() . . . 396

16.83.2.21 setRetentionTime() . . . 396
16.83.2.22 setSequence() . . . 396
16.83.2.23 setSummaryFormula() . . . 397
16.83.2.24 store() . . . 397
16.84 cSequenceDatabase Class Reference . . . 397
16.84.1 Detailed Description . . . 398
16.84.2 Member Function Documentation . . . 398
16.84.2.1 attachDecoys() . . . 398
16.84.2.2 erase() . . . 398
16.84.2.3 load() . . . 398
16.84.2.4 loadFromPlainTextStream() . . . 398
16.84.2.5 operator[]() . . . 399
16.84.2.6 push_back() . . . 399
16.84.2.7 size() . . . 399
16.84.2.8 store() . . . 399
16.84.2.9 storeToPlainTextStream() . . . 400
16.85 cSequenceDatabaseProxyModel Class Reference . . . 400
16.85.1 Detailed Description . . . 400
16.85.2 Constructor & Destructor Documentation . . . 401
16.85.2.1 cSequenceDatabaseProxyModel() . . . 401
16.85.3 Member Function Documentation . . . 401
16.85.3.1 filterAcceptsRow() . . . 401
16.85.3.2 initialize() . . . 401
16.85.3.3 lessThan() . . . 402
16.85.3.4 setWholeWord() . . . 402
16.86 cSequenceDatabaseWidget Class Reference . . . 402
16.86.1 Detailed Description . . . 403
16.86.2 Constructor & Destructor Documentation . . . 403
16.86.2.1 cSequenceDatabaseWidget() . . . 403
16.86.3 Member Function Documentation . . . 403
16.86.3.1 applyGlobalPreferences() . . . 403
16.86.3.2 closeEvent() . . . 403
16.86.3.3 importSequence() . . . 404
16.86.3.4 keyPressEvent() . . . 404
16.87 cSpectrumAnalyzerThreadMS1 Class Reference . . . 404
16.87.1 Detailed Description . . . 405
16.87.2 Member Function Documentation . . . 405
16.87.2.1 initialize() . . . 405
16.88 cSpectrumComparatorThreadMS1 Class Reference . . . 405
16.88.1 Detailed Description . . . 406
16.88.2 Member Function Documentation . . . 406
16.88.2.1 initialize() . . . 406

16.89 cSpectrumComparatorThreadMS2 Class Reference . . . 406
16.89.1 Detailed Description . . . 407
16.89.2 Member Function Documentation . . . 407
16.89.2.1 initialize() . . . 407
16.90 cSpectrumDetailProxyModel Class Reference . . . 407
16.90.1 Detailed Description . . . 408
16.90.2 Constructor & Destructor Documentation . . . 408
16.90.2.1 cSpectrumDetailProxyModel() . . . 408
16.90.3 Member Function Documentation . . . 408
16.90.3.1 filterAcceptsRow() . . . 408
16.90.3.2 lessThan() . . . 409
16.90.3.3 setFlags() . . . 409
16.91 cSpectrumDetailWidget Class Reference . . . 409
16.91.1 Detailed Description . . . 411
16.91.2 Constructor & Destructor Documentation . . . 411
16.91.2.1 cSpectrumDetailWidget() . . . 411
16.91.3 Member Function Documentation . . . 412
16.91.3.1 absoluteIntensityStateChangedSignal . . . 412
16.91.3.2 closeEvent() . . . 412
16.91.3.3 emitMZInterval . . . 412
16.91.3.4 exportImage() . . . 412
16.91.3.5 findAll() . . . 413
16.91.3.6 getDetailsAsHTMLString() . . . 413
16.91.3.7 getLocalNeutralLosses() . . . 413
16.91.3.8 getPartialPeaksTableAsHTMLString() . . . 413
16.91.3.9 getPeaksTableAsHTMLString() . . . 414
16.91.3.10 getTheoreticalSpectrum() . . . 414
16.91.3.11 hasAbsoluteIntensityEnabled() . . . 414
16.91.3.12 hasHideMatchedPeaksEnabled() . . . 414
16.91.3.13 hasHidePeakLabelsEnabled() . . . 415
16.91.3.14 hasHideScrambledPeaksEnabled() . . . 415
16.91.3.15 hasHideUnmatchedPeaksEnabled() . . . 415
16.91.3.16 hasMouseMzSelectionEnabled() . . . 415
16.91.3.17 hasProfileSpectrumEnabled() . . . 415
16.91.3.18 initialize() . . . 415
16.91.3.19 keyPressEvent() . . . 416
16.91.3.20 lastActiveDetail . . . 416
16.91.3.21 moveEvent() . . . 416
16.91.3.22 operator=() . . . 417
16.91.3.23 prepareToShow() . . . 417
16.91.3.24 rawDataStateChangedSignal . . . 417
16.91.3.25 setAbsoluteIntensityEnabled() . . . 417

16.91.3.26 setHideMatchedPeaksEnabled() . . . 418
16.91.3.27 setHidePeakLabelsEnabled() . . . 418
16.91.3.28 setHideScrambledPeaksEnabled() . . . 418
16.91.3.29 setHideUnmatchedPeaksEnabled() . . . 418
16.91.3.30 setMouseMzSelectionEnabled() . . . 418
16.91.3.31 setProfileSpectrumEnabled() . . . 419
16.91.3.32 zoomToPeak() . . . 419
16.92 cSpectrumGeneratorThreadMS1 Class Reference . . . 419
16.92.1 Detailed Description . . . 419
16.92.2 Member Function Documentation . . . 420
16.92.2.1 initialize() . . . 420
16.93 cSpectrumSceneWidget Class Reference . . . 420
16.93.1 Detailed Description . . . 421
16.93.2 Constructor & Destructor Documentation . . . 421
16.93.2.1 cSpectrumSceneWidget() . . . 421
16.93.3 Member Function Documentation . . . 421
16.93.3.1 exportToPDF() . . . 422
16.93.3.2 exportToPNG() . . . 422
16.93.3.3 exportToSVG() . . . 422
16.93.3.4 hideMatchedPeaks() . . . 422
16.93.3.5 hidePeakLabels() . . . 422
16.93.3.6 hideScrambledPeaks() . . . 423
16.93.3.7 hideUnmatchedPeaks() . . . 423
16.93.3.8 initialize() . . . 423
16.93.3.9 mouseMoveEvent() . . . 423
16.93.3.10 mousePressEvent() . . . 424
16.93.3.11 mouseReleaseEvent() . . . 424
16.93.3.12 resizeEvent() . . . 424
16.93.3.13 showEvent() . . . 424
16.93.3.14 updateMZInterval . . . 424
16.93.3.15 wheelEvent() . . . 425
16.94 cSummaryFormula Class Reference . . . 425
16.94.1 Detailed Description . . . 426
16.94.2 Constructor & Destructor Documentation . . . 426
16.94.2.1 cSummaryFormula() . . . 426
16.94.3 Member Function Documentation . . . 426
16.94.3.1 addFormula() . . . 426
16.94.3.2 equals() . . . 426
16.94.3.3 getFancySummary() . . . 427
16.94.3.4 getIsotopePattern() . . . 427
16.94.3.5 getMass() . . . 427
16.94.3.6 getSummary() . . . 428

16.94.3.7 hasAllElementsPositive() . . . 428
16.94.3.8 isEmpty() . . . 428
16.94.3.9 isPartial() . . . 428
16.94.3.10 isValid() . . . 428
16.94.3.11 load() . . . 429
16.94.3.12 setFormula() . . . 429
16.94.3.13 setFromMap() . . . 429
16.94.3.14 setPartial() . . . 429
16.94.3.15 store() . . . 430
16.95 cSummaryPeaksTableProxyModel Class Reference . . . 430
16.95.1 Detailed Description . . . 431
16.95.2 Constructor & Destructor Documentation . . . 431
16.95.2.1 cSummaryPeaksTableProxyModel() . . . 431
16.95.3 Member Function Documentation . . . 431
16.95.3.1 filterAcceptsRow() . . . 431
16.95.3.2 initialize() . . . 431
16.95.3.3 lessThan() . . . 432
16.95.3.4 setRegion() . . . 432
16.95.3.5 setWholeWord() . . . 433
16.96 cSummaryPeaksTableWidget Class Reference . . . 433
16.96.1 Detailed Description . . . 434
16.96.2 Constructor & Destructor Documentation . . . 434
16.96.2.1 cSummaryPeaksTableWidget() . . . 434
16.96.3 Member Function Documentation . . . 434
16.96.3.1 applyGlobalPreferences() . . . 434
16.96.3.2 closeEvent() . . . 435
16.96.3.3 filterCompound() . . . 435
16.96.3.4 getPubChemSearchWidget() . . . 435
16.96.3.5 keyPressEvent() . . . 435
16.96.3.6 prepareToShow() . . . 435
16.96.3.7 sendFilterOptionsToChromatogram . . . 436
16.96.3.8 sendFilterOptionsToImageWindow . . . 436
16.96.3.9 summaryPeaksTableRowDoubleClicked . . . 437
16.96.3.10 updateFilterBySelectedRegion() . . . 437
16.97 cSummaryTableKeyMS Struct Reference . . . 437
16.97.1 Detailed Description . . . 438
16.98 cSummaryTableKeyMSMS Struct Reference . . . 438
16.98.1 Detailed Description . . . 439
16.99 cSummaryTableKeyMSMS_comp Struct Reference . . . 439
16.99.1 Detailed Description . . . 439
16.99.2 Member Function Documentation . . . 439
16.99.2.1 operator()() . . . 439

16.100 cTablePosition Struct Reference . . . 439
16.100.1 Detailed Description . . . 440
16.101 cTheoreticalSpectrum Class Reference . . . 440
16.101.1 Detailed Description . . . 443
16.101.2 Constructor & Destructor Documentation . . . 443
16.101.2.1 cTheoreticalSpectrum() . . . 443
16.101.3 Member Function Documentation . . . 443
16.101.3.1 attach() . . . 443
16.101.3.2 clear() . . . 443
16.101.3.3 compareAverageMSSpectrum() . . . 443
16.101.3.4 compareBranchCyclic() . . . 444
16.101.3.5 compareBranched() . . . 444
16.101.3.6 compareCyclic() . . . 445
16.101.3.7 compareCyclicPolyketide() . . . 445
16.101.3.8 compareLinear() . . . 446
16.101.3.9 compareLinearPolyketide() . . . 446
16.101.3.10 compareMSSpectrum() . . . 447
16.101.3.11 compareOther() . . . 447
16.101.3.12 equals() . . . 448
16.101.3.13 finalizeMSSpectrum() . . . 448
16.101.3.14 generateCTerminalFragmentIons() . . . 448
16.101.3.15 generateFineMSSpectrum() . . . 449
16.101.3.16 generateMSSpectrum() . . . 449
16.101.3.17 generateNTerminalFragmentIons() . . . 450
16.101.3.18 getCandidate() . . . 450
16.101.3.19 getCosineSimilarity() . . . 451
16.101.3.20 getCoverageBySeries() . . . 451
16.101.3.21 getDecoyScores() . . . 451
16.101.3.22 getExperimentalMatches() . . . 451
16.101.3.23 getExperimentalSpectrum() . . . 451
16.101.3.24 getHintsIndex() . . . 452
16.101.3.25 getLabelsOfRotations() . . . 452
16.101.3.26 getNumberOfCompletedSeries() . . . 452
16.101.3.27 getNumberOfMatchedPeaks() [1/2] . . . 452
16.101.3.28 getNumberOfMatchedPeaks() [2/2] . . . 453
16.101.3.29 getNumberOfMatchedPeaksB() . . . 453
16.101.3.30 getNumberOfMatchedPeaksY() . . . 453
16.101.3.31 getNumberOfMatchedPeaksYB() . . . 453
16.101.3.32 getNumberOfPeaks() . . . 454
16.101.3.33 getNumberOfScrambledPeaks() . . . 454
16.101.3.34 getPathId() . . . 454
16.101.3.35 getRatioOfMatchedPeaks() . . . 454

16.101.3.36 getReverseValidPosition() . . . 454
16.101.3.37 getSumOfRelativeIntensities() . . . 455
16.101.3.38 getTargetPatternFDR() . . . 455
16.101.3.39 getTargetPatternScore() . . . 455
16.101.3.40 getTargetScores() . . . 455
16.101.3.41 getTheoreticalPeaks() . . . 456
16.101.3.42 getUnmatchedExperimentalPeaksCount() . . . 456
16.101.3.43 getValidPosition() . . . 456
16.101.3.44 getWeightedRatioOfMatchedPeaks() . . . 456
16.101.3.45 isValid() . . . 456
16.101.3.46 load() . . . 456
16.101.3.47 operator[]() . . . 457
16.101.3.48 resizePeakList() . . . 457
16.101.3.49 setCandidate() . . . 457
16.101.3.50 setExperimentalSpectrum() . . . 458
16.101.3.51 setFDRs() . . . 458
16.101.3.52 setNumberOfCompletedSeries() . . . 458
16.101.3.53 setParameters() . . . 458
16.101.3.54 setPathId() . . . 458
16.101.3.55 setValidSequence() . . . 459
16.101.3.56 sortByMass() . . . 459
16.101.3.57 store() . . . 459
16.102 cTheoreticalSpectrumList Class Reference . . . 459
16.102.1 Detailed Description . . . 460
16.102.2 Member Function Documentation . . . 460
16.102.2.1 add() . . . 460
16.102.2.2 clear() . . . 460
16.102.2.3 get() . . . 461
16.102.2.4 initialize() . . . 461
16.102.2.5 parallelCompareAndStore() . . . 461
16.102.2.6 resize() [1/2] . . . 462
16.102.2.7 resize() [2/2] . . . 462
16.102.2.8 size() [1/2] . . . 462
16.102.2.9 size() [2/2] . . . 462
16.102.2.10 sortAndFitSize() . . . 462
16.102.2.11 updatekNNList() . . . 463
16.103 cTileInfo Struct Reference . . . 463
16.103.1 Detailed Description . . . 464
16.104 cViewButtonDelegate Class Reference . . . 464
16.104.1 Detailed Description . . . 464
16.104.2 Constructor & Destructor Documentation . . . 464
16.104.2.1 cViewButtonDelegate() . . . 464

16.104.3 Member Function Documentation . . . 464
16.104.3.1 displayText() . . . 465
16.104.3.2 editorEvent() . . . 465
16.104.3.3 paint() . . . 465
16.105 hash_cCandidate Struct Reference . . . 466
16.105.1 Detailed Description . . . 466
16.105.2 Member Function Documentation . . . 466
16.105.2.1 operator()() . . . 466
16.106 hash_cEdge Struct Reference . . . 466
16.106.1 Detailed Description . . . 466
16.106.2 Member Function Documentation . . . 466
16.106.2.1 operator()() . . . 467
16.107 hash_cIonLabel Struct Reference . . . 467
16.107.1 Detailed Description . . . 467
16.107.2 Member Function Documentation . . . 467
16.107.2.1 operator()() . . . 467
16.108 hintStructure Struct Reference . . . 468
16.108.1 Detailed Description . . . 468
16.109 isotope Struct Reference . . . 468
16.109.1 Detailed Description . . . 468
16.110 layerInfo Struct Reference . . . 468
16.110.1 Detailed Description . . . 469
16.111 matchedSeries Struct Reference . . . 470
16.111.1 Detailed Description . . . 470
16.111.2 Member Function Documentation . . . 470
16.111.2.1 load() . . . 470
16.111.2.2 store() . . . 470
16.112 microscopyLayerInfo Struct Reference . . . 471
16.112.1 Detailed Description . . . 472
16.113 neutralLoss Struct Reference . . . 472
16.113.1 Detailed Description . . . 472
16.113.2 Member Function Documentation . . . 472
16.113.2.1 load() . . . 472
16.113.2.2 store() . . . 473
16.114 nodeEdge Struct Reference . . . 473
16.114.1 Detailed Description . . . 473
16.114.2 Member Function Documentation . . . 473
16.114.2.1 load() . . . 473
16.114.2.2 store() . . . 474
16.115 splitSite Struct Reference . . . 474
16.115.1 Detailed Description . . . 474
16.116 TRotationInfo Struct Reference . . . 474

16.116.1 Detailed Description . . . 475
16.116.2 Member Function Documentation . . . 475
16.116.2.1 endsWithBracket() . . . 475
16.116.2.2 startsWithBracket() . . . 475

17 File Documentation **477**
17.1 core/cBrick.h File Reference . . . 477
17.1.1 Detailed Description . . . 477
17.1.2 Function Documentation . . . 477
17.1.2.1 getNumberOfBricks() . . . 477
17.2 cBrick.h . . . 478
17.3 core/cBricksDatabase.h File Reference . . . 480
17.3.1 Detailed Description . . . 480
17.3.2 Function Documentation . . . 480
17.3.2.1 compareBrickMasses() . . . 480
17.3.2.2 generateBricksPermutations() . . . 481
17.3.2.3 getNameOfCompositionFromIntVector() . . . 481
17.3.2.4 getReversedNameOfCompositionFromStringVector() . . . 481
17.4 cBricksDatabase.h . . . 481
17.5 core/cCandidate.h File Reference . . . 483
17.5.1 Detailed Description . . . 484
17.5.2 Function Documentation . . . 484
17.5.2.1 operator==() . . . 484
17.6 cCandidate.h . . . 484
17.7 core/cCandidateSet.h File Reference . . . 488
17.7.1 Detailed Description . . . 488
17.8 cCandidateSet.h . . . 488
17.9 core/cDeNovoGraph.h File Reference . . . 489
17.9.1 Detailed Description . . . 489
17.9.2 Function Documentation . . . 489
17.9.2.1 compareNodes() . . . 489
17.10 cDeNovoGraph.h . . . 490
17.11 core/cDeNovoGraphNode.h File Reference . . . 491
17.11.1 Detailed Description . . . 492
17.11.2 Function Documentation . . . 492
17.11.2.1 compareEdgesByPPMError() . . . 492
17.11.2.2 compareEdgesByTargetNodeIntensityDesc() . . . 492
17.11.2.3 operator==() [1/2] . . . 492
17.11.2.4 operator==() [2/2] . . . 493
17.12 cDeNovoGraphNode.h . . . 493
17.13 core/cFragmentIons.h File Reference . . . 495
17.13.1 Detailed Description . . . 510

17.13.2 Function Documentation . . . 510
17.13.2.1 charge() . . . 511
17.13.2.2 initializeFragmentIonsForDeNovoGraphOfBranchCyclicPeptides() . . . 511
17.13.2.3 initializeFragmentIonsForDeNovoGraphOfCyclicPeptides() . . . 511
17.13.2.4 initializeFragmentIonsForDeNovoGraphOfCyclicPolyketide() . . . 511
17.13.2.5 initializeFragmentIonsForDeNovoGraphOfLinearPeptides() . . . 511
17.13.2.6 initializeFragmentIonsForDeNovoGraphOfLinearPolyketide() . . . 512
17.13.2.7 initializeFragmentIonsForDeNovoGraphOfTPeptides() . . . 512
17.13.2.8 loadModificationsFromPlainTextStream() . . . 512
17.13.2.9 storeModificationsToPlainTextStream() . . . 513
17.13.2.10 uncharge() . . . 513
17.14 cFragmentIons.h . . . 513
17.15 core/cGlobalPreferences.h File Reference . . . 525
17.15.1 Detailed Description . . . 526
17.16 cGlobalPreferences.h . . . 526
17.17 core/cImzML.h File Reference . . . 527
17.17.1 Detailed Description . . . 528
17.18 cImzML.h . . . 528
17.19 core/cIsotopePatternCache.h File Reference . . . 529
17.19.1 Detailed Description . . . 529
17.20 cIsotopePatternCache.h . . . 529
17.21 core/cMzML.h File Reference . . . 530
17.21.1 Detailed Description . . . 530
17.21.2 Function Documentation . . . 531
17.21.2.1 compareElementTagName() . . . 531
17.21.2.2 getAttribute() . . . 531
17.21.2.3 setAttribute() . . . 531
17.22 cMzML.h . . . 532
17.23 core/cParameters.h File Reference . . . 532
17.23.1 Detailed Description . . . 533
17.23.2 Function Documentation . . . 533
17.23.2.1 checkAdvancedFilteringRules() . . . 533
17.23.2.2 checkSeniorRules() . . . 534
17.23.2.3 searchHintForDeNovo() . . . 534
17.24 cParameters.h . . . 534
17.25 core/cPeak.h File Reference . . . 540
17.25.1 Detailed Description . . . 540
17.26 cPeak.h . . . 540
17.27 core/cPeakListSeries.h File Reference . . . 541
17.27.1 Detailed Description . . . 542
17.28 cPeakListSeries.h . . . 542
17.29 core/cPeaksList.h File Reference . . . 543

17.29.1 Detailed Description . . . 543
17.29.2 Function Documentation . . . 543
17.29.2.1 compareRelativePeakIntensitiesDesc() . . . 543
17.29.2.2 isInPpmMassErrorTolerance() . . . 544
17.29.2.3 ppmError() . . . 544
17.29.2.4 searchHint() . . . 544
17.30 cPeaksList.h . . . 545
17.31 core/cSequence.h File Reference . . . 548
17.31.1 Detailed Description . . . 548
17.32 cSequence.h . . . 548
17.33 core/cSequenceDatabase.h File Reference . . . 549
17.33.1 Detailed Description . . . 550
17.34 cSequenceDatabase.h . . . 550
17.35 core/cSummaryFormula.h File Reference . . . 551
17.35.1 Detailed Description . . . 551
17.35.2 Function Documentation . . . 551
17.35.2.1 addStringFormulaToMap() . . . 551
17.35.2.2 mergeMaps() . . . 551
17.35.2.3 rechargeMap() . . . 552
17.35.2.4 reduceMapToStringFormula() . . . 552
17.36 cSummaryFormula.h . . . 552
17.37 core/cTheoreticalSpectrum.h File Reference . . . 553
17.37.1 Detailed Description . . . 554
17.37.2 Function Documentation . . . 554
17.37.2.1 getNumberOfScoreHits() . . . 554
17.38 cTheoreticalSpectrum.h . . . 554
17.39 core/cTheoreticalSpectrumList.h File Reference . . . 560
17.39.1 Detailed Description . . . 560
17.40 cTheoreticalSpectrumList.h . . . 560
17.41 core/utilities.h File Reference . . . 561
17.41.1 Detailed Description . . . 564
17.41.2 Function Documentation . . . 564
17.41.2.1 checkFile() . . . 564
17.41.2.2 checkRegex() . . . 565
17.41.2.3 compareDoubles() . . . 565
17.41.2.4 compareMapsIntDouble() . . . 565
17.41.2.5 compareStringVectors() . . . 566
17.41.2.6 convertStringIntUnorderedMapToStringVector() . . . 566
17.41.2.7 cropDecimalsByteArray() . . . 566
17.41.2.8 cropPrecisionToSixDecimals() . . . 567
17.41.2.9 cropPrecisionToSixDecimalsByteArray() . . . 567
17.41.2.10 fact() . . . 567

17.41.2.11 fixLabelCharge() . . . 567
17.41.2.12 getPeptideTypeFromString() . . . 568
17.41.2.13 getShortFileName() . . . 568
17.41.2.14 getStringFromPeptideType() . . . 568
17.41.2.15 isCompatibleVersion() . . . 569
17.41.2.16 isWhiteSpace() . . . 569
17.41.2.17 isWhiteSpaceExceptSpace() . . . 569
17.41.2.18 loadDoubleVector() . . . 570
17.41.2.19 loadIntStringMap() . . . 570
17.41.2.20 loadIntVector() . . . 570
17.41.2.21 loadString() . . . 570
17.41.2.22 loadStringIntMap() . . . 571
17.41.2.23 loadStringIntMapVector() . . . 571
17.41.2.24 loadStringVector() . . . 571
17.41.2.25 median() . . . 571
17.41.2.26 parseBranch() . . . 572
17.41.2.27 proxyModelCheckDouble() . . . 572
17.41.2.28 proxyModelCheckInt() . . . 573
17.41.2.29 proxyModelCheckString() . . . 573
17.41.2.30 removeWhiteSpacesExceptSpaces() . . . 574
17.41.2.31 storeDoubleVector() . . . 574
17.41.2.32 storeIntStringMap() . . . 574
17.41.2.33 storeIntVector() . . . 574
17.41.2.34 storeString() . . . 575
17.41.2.35 storeStringIntMap() . . . 575
17.41.2.36 storeStringIntMapVector() . . . 575
17.41.2.37 storeStringVector() . . . 575
17.41.2.38 stripHTML() . . . 576
17.41.2.39 stripIsomers() . . . 576
17.41.2.40 stripIsomersFromStringVector() . . . 576
17.42 utilities.h . . . 576
17.43 gui/cAboutWidget.h File Reference . . . 580
17.43.1 Detailed Description . . . 580
17.44 cAboutWidget.h . . . 580
17.45 gui/cBranchCyclicWidget.h File Reference . . . 581
17.45.1 Detailed Description . . . 581
17.45.2 Function Documentation . . . 581
17.45.2.1 generateBranchLabelsDown() . . . 581
17.45.2.2 generateBranchLabelsUp() . . . 582
17.46 cBranchCyclicWidget.h . . . 583
17.47 gui/cBranchedWidget.h File Reference . . . 584
17.47.1 Detailed Description . . . 584

17.48 cBranchedWidget.h . . . 584
17.49 gui/cBricksDatabaseProxyModel.h File Reference . . . 585
17.49.1 Detailed Description . . . 585
17.50 cBricksDatabaseProxyModel.h . . . 585
17.51 gui/cBricksDatabaseWidget.h File Reference . . . 586
17.51.1 Detailed Description . . . 587
17.51.2 Function Documentation . . . 587
17.51.2.1 numberOfOccurrences() . . . 587
17.52 cBricksDatabaseWidget.h . . . 587
17.53 gui/cCalibrationChart.h File Reference . . . 589
17.53.1 Detailed Description . . . 590
17.54 cCalibrationChart.h . . . 590
17.55 gui/cCalibrationChartProxyModel.h File Reference . . . 592
17.55.1 Detailed Description . . . 592
17.56 cCalibrationChartProxyModel.h . . . 592
17.57 gui/cCalibrationChartScene.h File Reference . . . 592
17.57.1 Detailed Description . . . 593
17.57.2 Function Documentation . . . 593
17.57.2.1 calculateLOD() . . . 593
17.57.2.2 calculateLOQ() . . . 593
17.57.2.3 calculateR2Value() . . . 594
17.58 cCalibrationChartScene.h . . . 594
17.59 gui/cCheckBoxDelegate.h File Reference . . . 597
17.59.1 Detailed Description . . . 597
17.60 cCheckBoxDelegate.h . . . 597
17.61 gui/cChromatogramWindow.h File Reference . . . 598
17.61.1 Detailed Description . . . 598
17.62 cChromatogramWindow.h . . . 598
17.63 gui/cChromatogramWindowWidget.h File Reference . . . 600
17.63.1 Detailed Description . . . 601
17.63.2 Function Documentation . . . 601
17.63.2.1 calculateBaseTau() . . . 602
17.63.2.2 calculateCaruana() . . . 602
17.63.2.3 calculateExponentialParameters() . . . 602
17.63.2.4 calculateGaussianParameters() . . . 603
17.63.2.5 calculateGuo() . . . 603
17.63.2.6 calculateTauUsingWeightedAverage() . . . 604
17.63.2.7 computeExponentialFunction() . . . 604
17.63.2.8 computeExponentiallyModifiedGaussFunction() . . . 604
17.63.2.9 erfcx() . . . 605
17.63.2.10 erfcxinv() . . . 605
17.63.2.11 getExponentiallyModifiedGaussianArea() . . . 606

17.63.2.12 getExponentiallyModifiedGaussianAreas() . . . 606
17.63.2.13 getGaussianArea() . . . 607
17.63.2.14 getGaussianAreas() . . . 607
17.64 cChromatogramWindowWidget.h . . . 607
17.65 gui/cComboBoxDelegate.h File Reference . . . 610
17.65.1 Detailed Description . . . 611
17.66 cComboBoxDelegate.h . . . 611
17.67 gui/cCyclicWidget.h File Reference . . . 611
17.67.1 Detailed Description . . . 612
17.67.2 Function Documentation . . . 612
17.67.2.1 generateCyclicLabelsToLeft() . . . 612
17.67.2.2 generateCyclicLabelsToRight() . . . 613
17.67.2.3 paintCircle() . . . 614
17.68 cCyclicWidget.h . . . 615
17.69 gui/cDefineCalibrationWidget.h File Reference . . . 616
17.69.1 Detailed Description . . . 616
17.70 cDefineCalibrationWidget.h . . . 616
17.71 gui/cDrawPeptideWidget.h File Reference . . . 618
17.71.1 Detailed Description . . . 618
17.72 cDrawPeptideWidget.h . . . 618
17.73 gui/cEventFilter.h File Reference . . . 620
17.73.1 Detailed Description . . . 620
17.74 cEventFilter.h . . . 620
17.75 gui/cExportDialog.h File Reference . . . 621
17.75.1 Detailed Description . . . 621
17.76 cExportDialog.h . . . 621
17.77 gui/cFindDialog.h File Reference . . . 621
17.77.1 Detailed Description . . . 622
17.78 cFindDialog.h . . . 622
17.79 gui/cFormulaPredictor.h File Reference . . . 623
17.79.1 Detailed Description . . . 623
17.80 cFormulaPredictor.h . . . 623
17.81 gui/cFragmentIonsListWidget.h File Reference . . . 625
17.81.1 Detailed Description . . . 625
17.82 cFragmentIonsListWidget.h . . . 626
17.83 gui/cGraphWidget.h File Reference . . . 626
17.83.1 Detailed Description . . . 627
17.84 cGraphWidget.h . . . 627
17.85 gui/cHTMLDelegate.h File Reference . . . 628
17.85.1 Detailed Description . . . 628
17.86 cHTMLDelegate.h . . . 628
17.87 gui/cHTMLExportDialog.h File Reference . . . 628

17.87.1 Detailed Description . . . 629
17.88 cHTMLExportDialog.h . . . 629
17.89 gui/cImageWindow.h File Reference . . . 630
17.89.1 Detailed Description . . . 630
17.90 cImageWindow.h . . . 630
17.91 gui/cImageWindowImportDialog.h File Reference . . . 636
17.91.1 Detailed Description . . . 637
17.92 cImageWindowImportDialog.h . . . 637
17.93 gui/cImageWindowTabElectron.h File Reference . . . 638
17.93.1 Detailed Description . . . 638
17.94 cImageWindowTabElectron.h . . . 638
17.95 gui/cImageWindowTabOptical.h File Reference . . . 639
17.95.1 Detailed Description . . . 640
17.95.2 Function Documentation . . . 640
17.95.2.1 parseLifMetadata() . . . 640
17.95.2.2 parseLifMetadataSubTree() . . . 640
17.95.2.3 readLifMetadata() . . . 640
17.96 cImageWindowTabOptical.h . . . 640
17.97 gui/cImageWindowWidget.h File Reference . . . 644
17.97.1 Detailed Description . . . 644
17.98 cImageWindowWidget.h . . . 644
17.99 gui/cLinearWidget.h File Reference . . . 649
17.99.1 Detailed Description . . . 649
17.99.2 Function Documentation . . . 649
17.99.2.1 drawLabels() . . . 649
17.99.2.2 insertLabel() . . . 650
17.99.2.3 operator==() . . . 650
17.100 cLinearWidget.h . . . 650
17.101 gui/cMainThread.h File Reference . . . 652
17.101.1 Detailed Description . . . 652
17.102 cMainThread.h . . . 652
17.103 gui/cMainWindow.h File Reference . . . 654
17.103.1 Detailed Description . . . 655
17.104 cMainWindow.h . . . 655
17.105 gui/cMainWindowProxyModel.h File Reference . . . 659
17.105.1 Detailed Description . . . 659
17.106 cMainWindowProxyModel.h . . . 660
17.107 gui/cModificationsProxyModel.h File Reference . . . 660
17.107.1 Detailed Description . . . 661
17.108 cModificationsProxyModel.h . . . 661
17.109 gui/cModificationsWidget.h File Reference . . . 661
17.109.1 Detailed Description . . . 662

17.110 cModificationsWidget.h . . . 662
17.111 gui/cMultipleButtonDelegate.h File Reference . . . 664
17.111.1 Detailed Description . . . 664
17.112 cMultipleButtonDelegate.h . . . 664
17.113 gui/cMultipleDatasetsTableProxyModel.h File Reference . . . 665
17.113.1 Detailed Description . . . 665
17.114 cMultipleDatasetsTableProxyModel.h . . . 665
17.115 gui/cMultipleDatasetsTableWidget.h File Reference . . . 666
17.115.1 Detailed Description . . . 666
17.115.2 Function Documentation . . . 666
17.115.2.1 calculateLineParameterA() . . . 667
17.115.2.2 calculateLineParameters() . . . 667
17.115.2.3 calculateLineParametersWithIterations() . . . 667
17.116 cMultipleDatasetsTableWidget.h . . . 667
17.117 gui/cNeutralLossesListWidget.h File Reference . . . 670
17.117.1 Detailed Description . . . 671
17.118 cNeutralLossesListWidget.h . . . 671
17.119 gui/cParametersWidget.h File Reference . . . 672
17.119.1 Detailed Description . . . 672
17.120 cParametersWidget.h . . . 672
17.121 gui/cPatternSimulatorSceneWidget.h File Reference . . . 677
17.121.1 Detailed Description . . . 677
17.121.2 Function Documentation . . . 677
17.121.2.1 computeGaussFunction() . . . 677
17.121.2.2 computeGaussHeight() . . . 677
17.122 cPatternSimulatorSceneWidget.h . . . 678
17.123 gui/cPatternSimulatorWidget.h File Reference . . . 680
17.123.1 Detailed Description . . . 681
17.124 cPatternSimulatorWidget.h . . . 681
17.125 gui/cPreferencesDialog.h File Reference . . . 683
17.125.1 Detailed Description . . . 683
17.126 cPreferencesDialog.h . . . 683
17.127 gui/cPreferencesIonTypeDialog.h File Reference . . . 684
17.127.1 Detailed Description . . . 685
17.128 cPreferencesIonTypeDialog.h . . . 685
17.129 gui/cPreferencesTabDataProcessing.h File Reference . . . 685
17.129.1 Detailed Description . . . 686
17.130 cPreferencesTabDataProcessing.h . . . 686
17.131 gui/cPreferencesTabDirectories.h File Reference . . . 687
17.131.1 Detailed Description . . . 687
17.132 cPreferencesTabDirectories.h . . . 687
17.133 gui/cPreferencesTabIonTypes.h File Reference . . . 689

17.133.1 Detailed Description . . . 689
17.134 cPreferencesTabIonTypes.h . . . 689
17.135 gui/cPreferencesTabURLs.h File Reference . . . 690
17.135.1 Detailed Description . . . 690
17.136 cPreferencesTabURLs.h . . . 691
17.137 gui/cPubChemSearchWidget.h File Reference . . . 692
17.137.1 Detailed Description . . . 692
17.138 cPubChemSearchWidget.h . . . 692
17.139 gui/cSequenceDatabaseProxyModel.h File Reference . . . 695
17.139.1 Detailed Description . . . 696
17.140 cSequenceDatabaseProxyModel.h . . . 696
17.141 gui/cSequenceDatabaseWidget.h File Reference . . . 697
17.141.1 Detailed Description . . . 697
17.142 cSequenceDatabaseWidget.h . . . 697
17.143 gui/cSpectrumDetailProxyModel.h File Reference . . . 699
17.143.1 Detailed Description . . . 699
17.144 cSpectrumDetailProxyModel.h . . . 700
17.145 gui/cSpectrumDetailWidget.h File Reference . . . 700
17.145.1 Detailed Description . . . 701
17.146 cSpectrumDetailWidget.h . . . 701
17.147 gui/cSpectrumSceneWidget.h File Reference . . . 705
17.147.1 Detailed Description . . . 706
17.148 cSpectrumSceneWidget.h . . . 706
17.149 gui/cSummaryPeaksTableProxyModel.h File Reference . . . 708
17.149.1 Detailed Description . . . 709
17.150 cSummaryPeaksTableProxyModel.h . . . 709
17.151 gui/cSummaryPeaksTableWidget.h File Reference . . . 710
17.151.1 Detailed Description . . . 710
17.151.2 Function Documentation . . . 710
17.151.2.1 addEICPeak() . . . 711
17.152 cSummaryPeaksTableWidget.h . . . 711
17.153 gui/cViewButtonDelegate.h File Reference . . . 715
17.153.1 Detailed Description . . . 715
17.154 cViewButtonDelegate.h . . . 715
17.155 parallel/cFormulaPredictorThread.h File Reference . . . 716
17.155.1 Detailed Description . . . 716
17.156 cFormulaPredictorThread.h . . . 716
17.157 parallel/cGraphReaderThread.h File Reference . . . 717
17.157.1 Detailed Description . . . 717
17.158 cGraphReaderThread.h . . . 718
17.159 parallel/cSpectrumAnalyzerThreadMS1.h File Reference . . . 718
17.159.1 Detailed Description . . . 719

17.160 cSpectrumAnalyzerThreadMS1.h . . . 719
17.161 parallel/cSpectrumComparatorThreadMS1.h File Reference . . . 719
17.161.1 Detailed Description . . . 720
17.162 cSpectrumComparatorThreadMS1.h . . . 720
17.163 parallel/cSpectrumComparatorThreadMS2.h File Reference . . . 720
17.163.1 Detailed Description . . . 721
17.163.2 Function Documentation . . . 721
17.163.2.1 compareBandAllIonsDesc() . . . 721
17.163.2.2 compareCosineSimilarityDesc() . . . 721
17.163.2.3 compareNumberOfMatchedPeaksDesc() . . . 722
17.163.2.4 comparePaths() . . . 722
17.163.2.5 compareSumOfRelIntDesc() . . . 722
17.163.2.6 compareWeightedRatioDesc() . . . 723
17.163.2.7 compareYandAllIonsDesc() . . . 723
17.163.2.8 compareYBandAllIonsDesc() . . . 723
17.164 cSpectrumComparatorThreadMS2.h . . . 724
17.165 parallel/cSpectrumGeneratorThreadMS1.h File Reference . . . 725
17.165.1 Detailed Description . . . 725
17.166 cSpectrumGeneratorThreadMS1.h . . . 725

Index **727**

Chapter 1

Introduction

CycloBranch

(v. 2.1.35)

Availability & Usability	Dereplication & Compound Search (MS, LC-MS, and MSI)	*De Novo* Sequencing & Database Search (MS/MS)
• Open-Source • Graphical User Interface • User-Friendly • Cross-Platform • Stand-Alone	• MALDI-MS • LC-MS • MALDI-MSI • DESI-MSI • Simulate Isotopic Pattern • Calculate Molecular Formula	• Linear Peptides • Cyclic Peptides • Branch-Cyclic Peptides • Siderophores • Polyketides
Supported File Formats	**Databases**	**Visualization**
• XML-based: mzML, mzXML, imzML • Native: baf (Bruker), raw (Thermo), raw (Waters) • Plain Text: mgf, txt	• Building Blocks (MS/MS) • Sequences (MS/MS) • Compounds (MS, LC-MS and MSI) • `Custom`	• Mass Spectra (MS, MSI, and MS/MS) • Draw Peptide Tool (MS/MS) • Chromatogram (LC-MS) • Image Fusion (MSI, histology and microscopy images)

If you use CycloBranch in your work, please cite us using

- *Anal. Chem.* (2020), 92(10):6844-6849. DOI: 10.1021/acs.analchem.0c00170
  Download citation: [**ris**], [**bib**]

or another related reference.

Chapter 2

User's Guide

- Main Window and Toolbar
  - Main Window
  - File
  - Search
  - Tools
  - View
  - Help
  - Text Filter
- Settings
  - Dereplication (MS)
  - Compound Search (MS)
  - De Novo Sequencing (MS/MS)
  - Database Search (MS/MS)
  - Direct Comparison of Experimental and Theoretical Spectrum (MS/MS)
- Output Report
- Linear Peptides
  - Linear Peptide Detail Window
  - Toolbar for Linear Peptides
  - Linear Sequence Detection
- Cyclic Peptides
  - Cyclic Peptide Detail Window
  - Toolbar for Cyclic Peptides
  - Cyclic Sequence Detection
  - Nomenclature of Cyclic Series
- Branched Peptides
  - Branched Peptide Detail Window
  - Toolbar for Branched Peptides
  - Branched Sequence Detection
  - Nomenclature of Branched Series
  - Fragmentation Example
  - Branched Siderophores

- Branch-cyclic Peptides
  - Branch-cyclic Peptide Detail Window
  - Toolbar for Branch-cyclic Peptides
  - Branch-cyclic Sequence Detection
  - Nomenclature of Branch-cyclic Series
  - Branch-cyclic Siderophores
- Linear Siderophores
  - Linear Nonribosomal Peptide Siderophores
  - Linear Polyketide Detail Window
  - Nomenclature of Linear Polyketide Series
  - Linear Polyketide Sequence Detection
- Cyclic Siderophores
  - Cyclic Nonribosomal Peptide Siderophores
  - Cyclic Polyketide Detail Window
  - Nomenclature of Cyclic Polyketide Series
  - Cyclic Polyketide Sequence Detection
- Dereplication
  - Dereplication in MS Data
  - Dereplication in LC-MS data
  - Dereplication in MSI Data
  - Databases of Compounds
- Building Blocks Editor
  - Building Blocks Editor
  - Format of Building Block Databases
  - Format of Ketide Building Blocks
  - List of Available Building Block Databases
- Sequence/Compound Database Editor
  - Sequence/Compound Database Editor
  - Format of Sequence Databases
  - List of Available Sequence/Compound Databases
- Modifications Editor
  - Modifications Editor
  - Format of Modification Databases
  - List of Available Modification Databases
- Draw Peptide Tool
- Summary Table of Matched Peaks
- Annotated Features Overview
- Chromatogram
- Image Fusion
- Simulate Isotopic Pattern

- Calculate Molecular Formula
- Text Filter
- Preferences

2.1 Main Window and Toolbar

2.1.1 Main Window

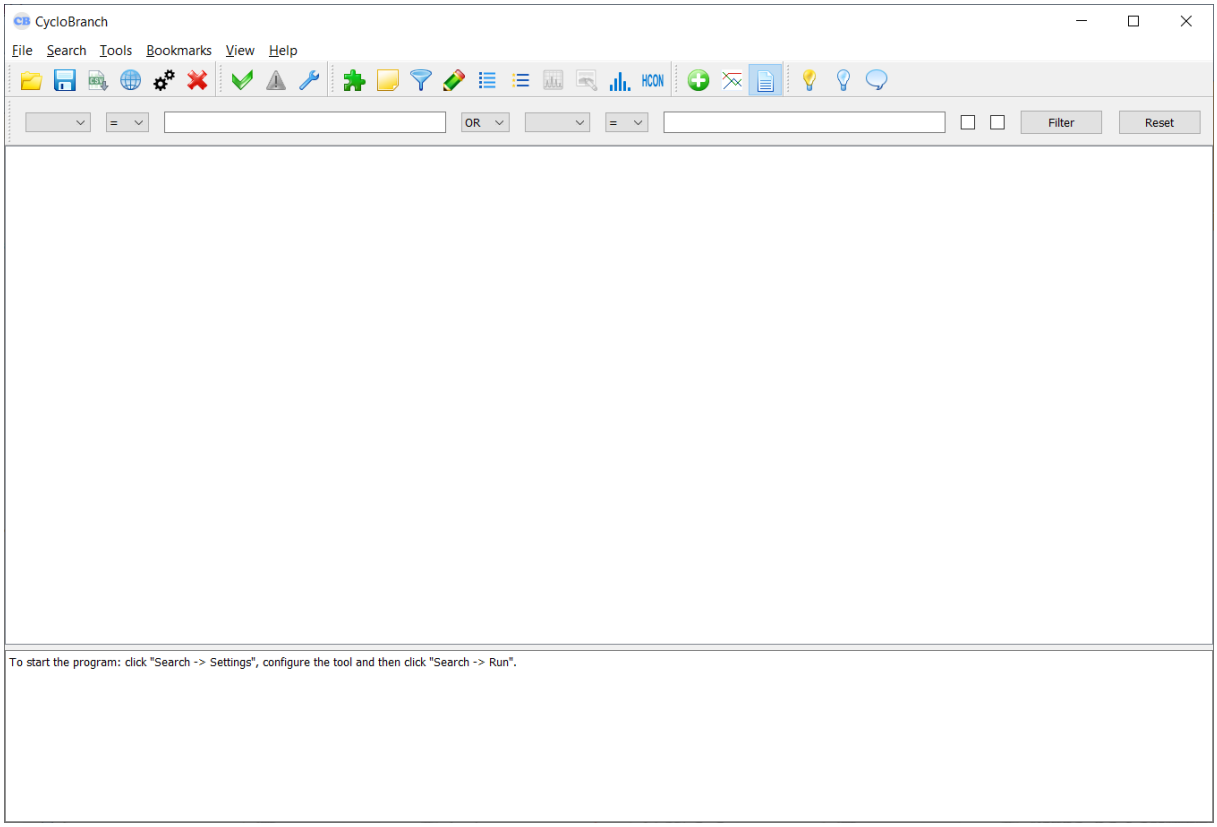


Figure 2.1 Main window of the application.

2.1.2 File

Open Results... (CTRL + O) 
Open an output report (a file ∗.res).

Save Results... (CTRL + S) 
Save an output report (a file ∗.res). Files created using an older version of the software cannot be opened in a newer version.

Export to CSV (CTRL + Shift + C) 
An output report is exported into a csv (comma separated values) file and it can be imported, e.g., to MS Excel.

Export to HTML (CTRL + Shift + H) 
An output report is exported into a HTML page.

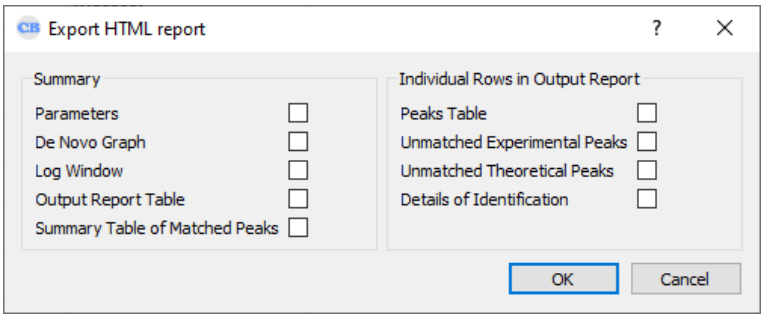


Figure 2.2 HTML Export Dialog.

Summary:

- Parameters - export the search parameters
- De Novo Graph - export the de novo graph (if "De Novo Search Engine" Mode was used)
- Log Window - export the content of Log Window
- Output Report Table - export the Output Report in the main window
- Summary Table of Matched Peaks - export the summary table of matched peaks; see Summary Table of Matched Peaks

Individual Rows in Output Report (the following data are exported for each row in the table):

- Peaks Table - export a table of peaks
- Unmatched Experimental Peaks - export unmatched experimental peaks
- Unmatched Theoretical Peaks - export unmatched theoretical peaks (MS/MS data only)
- Details of Identification - export identification details (MS/MS data only)

The tables in the output report can be sorted using any column. This feature uses `jQuery` and requires an on-line connection. If the feature does not work, the following paths in a source code of the generated HTML report must be fixed:

```
https://cdnjs.cloudflare.com/ajax/libs/jquery/2.1.4/jquery.min.js
https://cdnjs.cloudflare.com/ajax/libs/jquery.tablesorter/2.23.3/js/jquery.tablesorter.min.js
https://cdnjs.cloudflare.com/ajax/libs/jquery.tablesorter/2.23.3/css/theme.green.min.css
```

Preferences (CTRL + Shift + P)
Global preferences of the application. See details here.

Quit (CTRL + Q)
Quit the application.

2.1.3 Search

Run (F5)
Run the application using predefined settings.

Stop (F6)
Stop the computation if running.

Settings... (F7)
Settings of the application. See details here.

2.1.4 Tools

Building Blocks Editor (CTRL + B) 
An editor of a database of building blocks is opened. See details here.

Sequence/Compound Database Editor (CTRL + D) 
An editor of a database of sequences is opened. See details here.

Modifications Editor (CTRL + M) 
An editor of a list of modifications is opened. See details here.

Draw Peptide (CTRL + P) 
The "Draw Peptide Tool" is opened. See details here.

Summary Table of Matched Peaks (F8) 
Show a summary table of matched peaks reported in all rows which are currently listed in the main window. See details here.

Chromatogram (F9) 
Show a chromatogram for LC-MS data. See details here.

Image Fusion (F10) 
Open the image fusion tool. See details here.

Simulate Isotopic Pattern (F11) 
Open the simulate isotopic pattern tool. See details here.

Calculate Molecular Formula (F12)
Open the calculate isotopic formula tool. See details here.

2.1.5 Bookmarks

A custom list of up to 10 bookmarks can be created in global Preferences dialog. The bookmarks can then be opened in an external web browser using CTRL + 1, CTRL + 2, ..., CTRL + 9, and CTRL + 0. See details here.

2.1.6 View

Show Isomers (CTRL + I) 
Show/hide isomers of building blocks.

De Novo Graph (CTRL + G) 
Open a window with the de novo graph. The graph is updated dynamically when the engine is running in "De Novo Search Engine" mode.

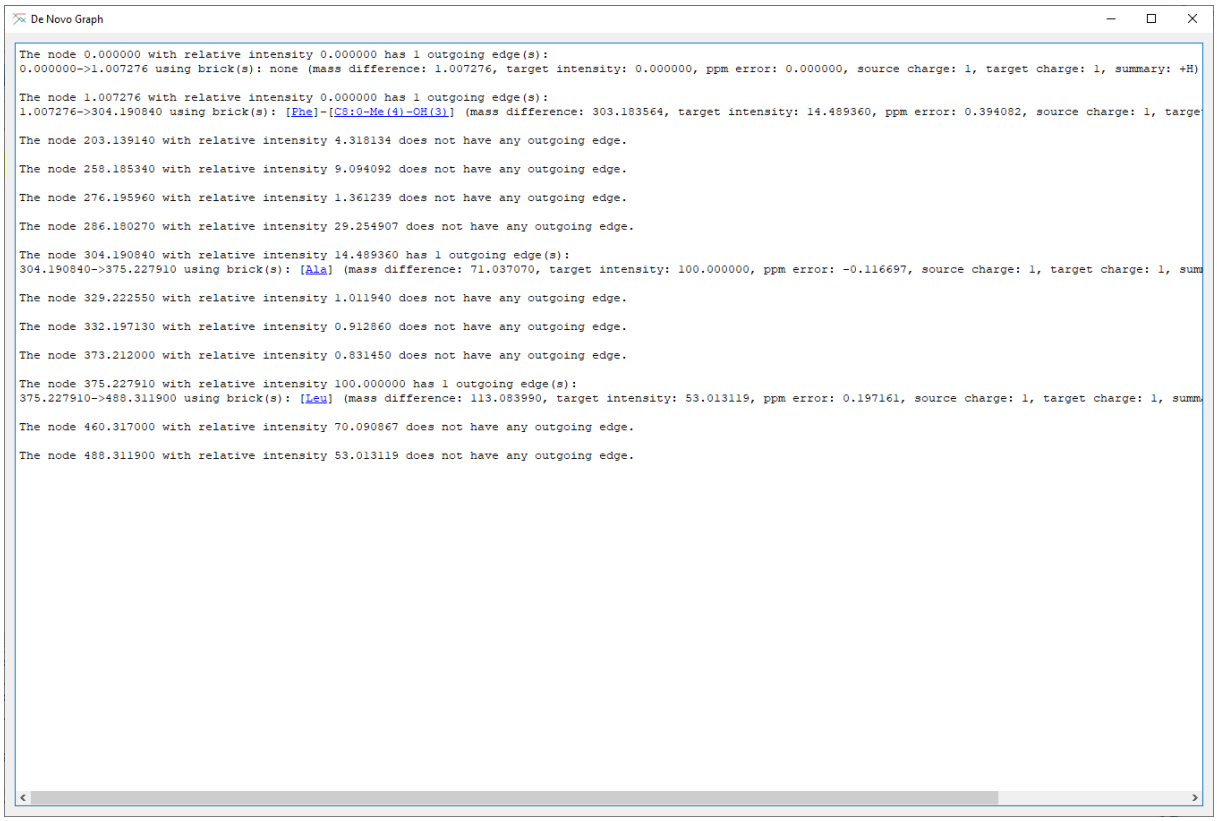


Figure 2.3 A de novo graph.

Log Window (CTRL + L) 

Enable/disable the log window with the current status of the application.

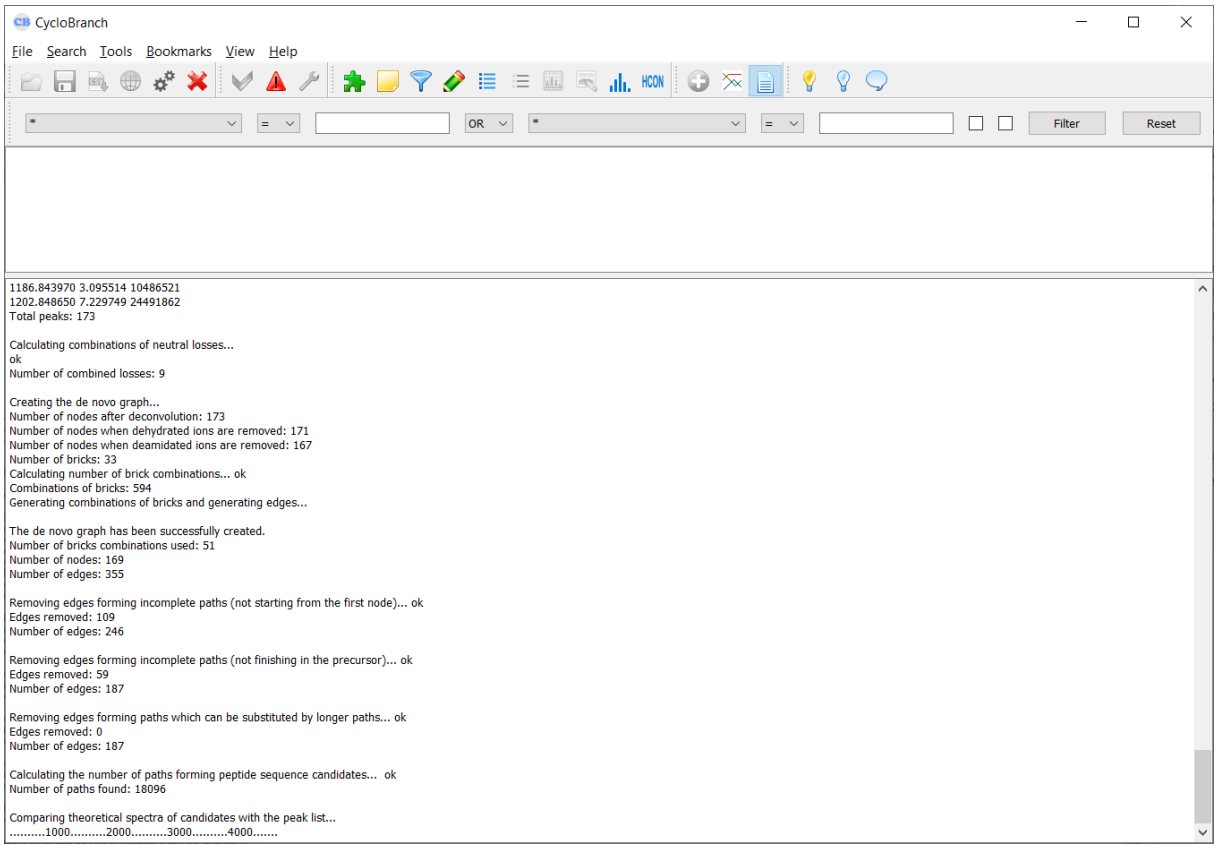


Figure 2.4 Log window.

2.1.7 Help

HTML Documentation (F1)
This documentation is opened from the local file 'docs/html/index.html'.

PDF Manual (F2)
The PDF manual is opened from the local file 'docs/refman.pdf'.

About (F3)
An about dialog of the application is shown (a version number, licence information, etc.).

2.1.8 Text Filter

See details here.

2.2 Settings

2.2.1 Settings - Dereplication (MS)

The settings dialog is opened using the command "Search -> Settings... ". Use F1 to open this help.
The application works in five different modes. This page summarizes the parameters used for dereplication. The differences in the configuration for other modes are described on extra pages, see the parameter Mode.
See also the following tutorials: Tutorial 1: Dereplication in MS1 data, Tutorial 2: Dereplication in LC-MS data, Tutorial 4: Dereplication in MSI data and fusion with histology image, and the page Dereplication.

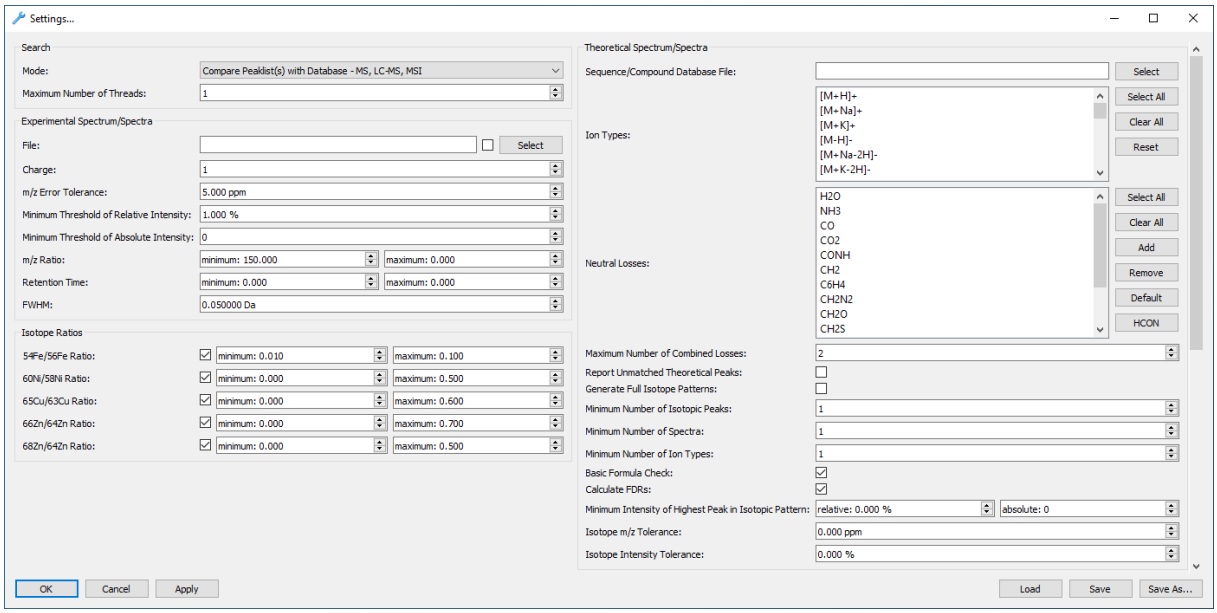


Figure 2.5 Settings dialog - dereplication (MS).

2.2.1.1 Basic Buttons

- OK (Enter) - Accept changes and hide window.
- Cancel (Esc) - Drop changes and hide window.
- Apply - Accept changes and keep window opened.
- Load (CTRL + L) - Load settings from a file (∗.ini).
- Save (CTRL + S) - Save settings in the current file (∗.ini). When a file has not been loaded yet, the "Save As ..." file dialog is opened.
- Save As... (CTRL + D) - Save settings into a file (∗.ini).

2.2.2 Search

2.2.2.1 Mode

- **'Compare Peaklist(s) with Database - MS, LC-MS, MSI'** - dereplication; the experimental peaklist(s) are compared with a theoretical peaklist generated from an input database of compounds.
- **'Compound Search - MS, LC-MS, MSI'** - the experimental peaklist(s) are compared with a theoretical peaklist generated from a list of compounds generated from an input list chemical elements (a database-free search method).
- **'De Novo Search Engine - MS/MS'** - determine a sequence/tag of a nonribosomal peptide, polyketide, or siderophore using the *de novo* sequencing of an input product ion mass spectrum.
- **'Compare Peaklist with Database - MS/MS'** - an experimental peaklist (product ion mass spectrum) is compared with theoretical spectra generated from a database of sequences.
- **'Compare Peaklist(s) with Spectrum of Searched Sequence - MS/MS'** - a theoretical spectrum is generated from the input field "Searched Sequence" and compared with the experimental peaklist(s), i.e., product ion mass spectrum/spectra.

2.2.2.2 Maximum Number of Threads

The maximum number of threads used during a search process.

2.2.3 Experimental Spectrum/Spectra

2.2.3.1 File (CTRL + P)

Select an input file with experimental spectra. Supported file formats:

File format	Supported types of spectra (data processing)	Supported types of spectra (visualization)	Description	Notes
txt	centroid	centroid	plain text file	Mass-to-charge ratio and intensity on each line separated by tabulator; multiple peaklists must be separated by an empty line.
mgf	centroid	centroid	Mascot Generic File format	
mzML	profile and centroid	profile and centroid	standard data format	m/z values and intensities must be stored as 64-bit/32-bit floats; no compression and zlib compression are supported. OpenMS 2.x must be installed.
mzXML	centroid	centroid	standard data format	OpenMS 2.x must be installed; the FileConverter tool is used to automatically convert this file format into mzML.
imzML	profile and centroid	profile and centroid	standard data format	Processed or continuos data file format; m/z values and intensities must be stored as 64-bit/32-bit floats; no compression. OpenMS 2.x must be installed.
baf	profile and centroid	profile and centroid	Bruker's Analysis File	CompassXport 3.0.13.1 or Compass↩ Xtract 3.2.↩ 201 (64-bit) must be installed; see details here; Windows only.
raw	profile and centroid	profile and centroid	Thermo raw file	OpenMS 2.x including Proteo↩ Wizard must be installed; Windows only.
raw	profile and centroid	profile and centroid	Waters raw directory	Select a ∗.dat file in the raw directory; Windows only.

File format	Supported types of spectra (data processing)	Supported types of spectra (visualization)	Description	Notes
mis (deprecated)	profile and centroid	centroid	Bruker's flex↩ Imaging File (old data format)	CompassXport 3.0 must be installed; Windows only. If the file "[name].mis" is selected, all the "analysis.baf" files in the subfolders of "[name]" directory are searched and processed.
ser (deprecated)	profile and centroid	centroid	Apex file format	CompassXport 3.0 must be installed; Windows only.

Multiple files can be selected in the modes 'Compare Peaklist(s) with Database - MS, LC-MS, MSI' and 'Compound Search - MS, LC-MS, MSI'. The feature is not available for imzML file format. If a directory name is specified, it is searched recursively and all found files are processed - the feature is currently available for baf files (Bruker) and raw directories (Waters) only.
The visualization of profile mass spectra is enabled if the checkbox next to the file name is checked. This feature is available for the following file formats:

- baf files (Bruker; Windows only)
- raw directories (Waters; Windows only)
- raw files (Thermo; Windows only)
- mzML
- imzML

CycloBranch supports the standard imzML data format for imaging mass spectra (see imzml.org). If the file includes profile spectra, they are automatically converted to centroided spectra using the OpenMS pipeline Noise↩ FilterGaussian >> BaseLineFilter >> PeakPickerHiRes (requires OpenMS 2.x installed). An external script is used to convert the raw data thus the parameters of the pipeline (e.g., S/N ratio) can be adjusted (edit the file "↩ External/windows/raw2peaks.bat", "External/linux/raw2peaks.sh" or "External/macosx/raw2peaks.sh" depending on your data requirements and the platform). Since the data conversion may be very time-consuming (hrs/days) due to the size of a dataset (e.g., tens of gigabytes), CycloBranch stores the converted files with centroided spectra for further use. The new filenames are "filename_converted_fwhm_value.imzML" and "filename_converted_fwhm_↩ value.ibd". CycloBranch automatically detects these files when the search process is repeated and opens a dialog to recommend their use. The "filename_converted_fwhm_value.imzML" can be used directly as a peaklist file in Settings to skip the recommendation dialog.

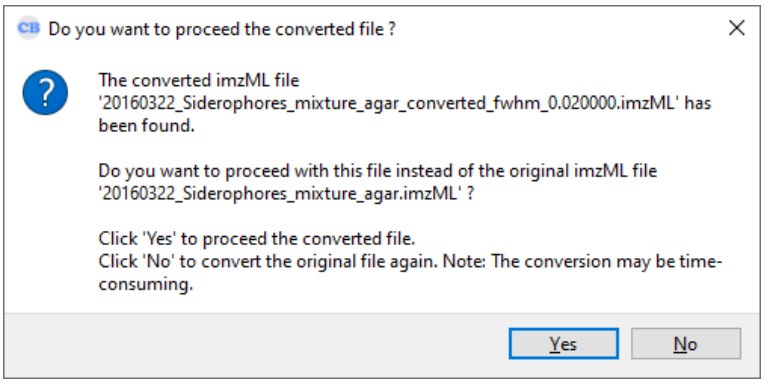


Figure 2.6 The dialog which recommends the use of centroided spectra instead of profile spectra.

2.2.3.2 Charge

MS mode - the value determines the maximum charge of generated theoretical peaks; a negative value is allowed. For example, the value 3 means that theoretical peaks of compouds are generated with charges 1+, 2+, and 3+. The value -1 means that theoretical peaks of compounds are generated with charge 1-.
MS/MS mode - the value defines the charge of precursor ion; a negative value is allowed. The charge of precursor ion in an input peaklist file is ignored.

2.2.3.3 m/z Error Tolerance

Enter the m/z error tolerance in MS mode or the fragment m/z error tolerance in MS/MS mode [ppm].

2.2.3.4 Minimum Threshold of Relative Intensity

Enter the minimum threshold of relative intensity in %. Peaks with relative intensities below the threshold are removed from an input experimental peaklist.

2.2.3.5 Minimum Threshold of Absolute Intensity

Enter the minimum threshold of absolute intensity. Peaks with absolute intensities below the threshold are removed from an input experimental peaklist.
The relative and absolute thresholds are used simultaneously.

2.2.3.6 m/z Ratio

Enter the minimum m/z ratio and maximum m/z ratio. Experimental peaks with m/z ratios below/above the thresholds are removed from the input experimental peaklist(s). For maximum m/z, 0 = disabled.

2.2.3.7 Retention Time

Limit the range of retention time in which the compounds are searched in 'Compare Peaklist(s) with Database - MS, LC-MS, MSI' and 'Compound Search - MS, LC-MS, MSI' modes if LC-MS data are processed.

2.2.3.8 FWHM

Full width at half maximum. The value is used if the profile spectra are converted into peaklists (mzML and imzML input files) and if the full isotope patterns of compounds are generated.

2.2.4 Isotope Ratios

2.2.4.1 54Fe/56Fe Ratio

Enter the minimum and maximum intensity ratios of peaks corresponding to 54Fe and 56Fe. If the option 'Generate Full Isotope Patterns' is enabled, the value of 'Minimum Number of Isotopic Peaks' must be at least 2 to enable this feature. The feature is enabled only if the checkbox is checked.

2.2.4.2 60Ni/58Ni Ratio

Enter the minimum and maximum intensity ratios of peaks corresponding to 60Ni and 58Ni. If the option 'Generate Full Isotope Patterns' is enabled, the value of 'Minimum Number of Isotopic Peaks' must be at least 2 to enable this feature. The feature is enabled only if the checkbox is checked. The default maximum value is 0.5 (for FWHM <= 0.001 only). The minimum and maximum values must be estimated empirically for FWHM > 0.001, because the intesity of 60Ni may be accumulated with 13C2, etc.

2.2.4.3 65Cu/63Cu Ratio

Enter the minimum and maximum intensity ratios of peaks corresponding to 65Cu and 63Cu. If the option 'Generate Full Isotope Patterns' is enabled, the value of 'Minimum Number of Isotopic Peaks' must be at least 2 to enable this feature. The feature is enabled only if the checkbox is checked. The default maximum value is 0.6 (for FWHM <= 0.001 only). The minimum and maximum values must be estimated empirically for FWHM > 0.001, because the intesity of 65Cu may be accumulated with 13C2, etc.

2.2.4.4 66Zn/64Zn Ratio

Enter the minimum and maximum intensity ratios of peaks corresponding to 66Zn and 64Zn. If the option 'Generate Full Isotope Patterns' is enabled, the value of 'Minimum Number of Isotopic Peaks' must be at least 2 to enable this feature. The feature is enabled only if the checkbox is checked. The default maximum value is 0.7 (for FWHM <= 0.001 only). The minimum and maximum values must be estimated empirically for FWHM > 0.001, because the intesity of 66Zn may be accumulated with 13C2, etc.

2.2.4.5 68Zn/64Zn Ratio

Enter the minimum and maximum intensity ratios of peaks corresponding to 68Zn and 64Zn. If the option 'Generate Full Isotope Patterns' is enabled, the value of 'Minimum Number of Isotopic Peaks' must be at least 3 to enable this feature. The feature is enabled only if the checkbox is checked. The default maximum value is 0.5 (for FWHM <= 0.001 only). The minimum and maximum values must be estimated empirically for FWHM > 0.001, because the intesity of 68Zn may be accumulated with 13C4, etc.

2.2.5 Theoretical Spectrum/Spectra

2.2.5.1 Sequence/Compound Database File (CTRL + T)

A text file containing a database of sequences/compounds. See Format of Sequence/Compound Databases.

2.2.5.2 Ion Types

The types of ions for MS, LC-MS, and MSI data can be defined in global Preferences dialog. The default list of ions is shown here.
If a metal ion is selected, the isotopes having natural abundance greater than 3% are generated automatically in theoretical peaklists next to the most abundant isotopes. If the option Generate Full Isotope Patterns is selected, this feature is disabled because full isotopic patterns are calculated.
The following types of fragment ions are defined for MS/MS spectra:

- A (a-ion)
- B (b-ion)
- C (c-ion)
- X (x-ion)
- Y (y-ion)
- Z (z-ion)

Note that y-ions are not supported when cyclic peptides are searched. Fragment ions series of linear and cyclic polyketides are described on extra pages. See Nomenclature of Linear Polyketide Series and Nomenclature of Cyclic Polyketide Series.
Buttons:

- "Select All" - select all fragment ions in the list.
- "Clear All" - unselect all fragment ions in the list.
- "Reset" - b-ion is selected automatically if a cyclic peptide is choosen as "Peptide Type". B-ion and y-ion are selected if a linear, branched or branch-cyclic peptide is choosen.

2.2.5.3 Neutral Losses

Define and select the types of neutral losses which will be used when generating theoretical spectra.
Buttons:

- "Select All" - select all neutral losses in the list.
- "Clear All" - unselect all neutral losses in the list.
- "Add" - add a new item at the end of the list - a valid molecular formula must be entered.
- "Remove" - remove selected items from the list.
- "Default" - clear the list and define a list of default neutral losses.
- "HCON" - clear the list and set the default chemical elements H, C, O, N, S, and P.

The default neutral losses are defined as follows:

Formula	Description
H2O	water
NH3	ammonia
CO	formyl group
CO2	carboxyl group
CONH	carbamyl group
CH2	shortened carbon chain
C6H4	benzene ring removal
CH2N2	part of Arg side chain
CH2O	Ser side chain
CH2S	Cys side chain
C4H9N	Lys side chain
C4H4N2	His side chain
C4H9N3	Arg side chain
C9H7N	Trp side chain

In MS/MS mode, the molecular formulas of proteinogenic amino acids side chains can be optionally combined from the default values as follows:

Amino acid	Side chain		Amino acid	Side chain
Gly	-		Asp	C2H2O2 (CH2+CO2)
Ala	CH2		Gln	C3H5NO (CH2+CH2+CONH)
Ser	CH2O		Lys	C4H9N
Pro	C3H4 (not defined)		Glu	C3H4O2 (CH2+CH2+CO2)
Val	C3H6 (CH2+CH2+CH2)		Met	C3H6S (CH2+CH2+CH2S)
Thr	C2H4O (CH2+CH2O)		His	C4H4N2
Cys	CH2S		Phe	C7H6 (CH2+C6H4)
Leu	C4H8 (CH2+CH2+CH2+CH2)		Arg	C4H9N3
Ile	C4H8 (CH2+CH2+CH2+CH2)		Tyr	C7H6O (CH2O+C6H4)
Asn	C2H3NO (CH2+CONH)		Trp	C9H7N

In some cases, the list of neutral losses can be replaced by a list of chemical elements (e.g. H, C, O, N, S, and P). This is useful if an experimental spectrum contains peaks corresponding to ions with unknown neutral losses, if a metabolite (i.e. a non-peptidic compound) is fragmented or if the mode 'Compound Search - MS, LC-MS, MSI' is used.

2.2.5.4 Maximum Number of Combined Losses

Maximum number of combined neutral losses.

2.2.5.5 Report Unmatched Theoretical Peaks

If checked, all unmatched theoretical peaks are reported. If unchecked, unmatched theoretical peaks are reported only if a corresponding isotope pattern has been matched. This feature may spend a lot of main memory, keep it disabled if possible.

2.2.5.6 Generate Full Isotope Patterns

The full isotope patterns of compounds are generated in theoretical spectra. The FWHM value is used for this purpose. If checked, the deisotoping is disabled automatically.

2.2.5.7 Minimum Number of Isotopic Peaks

The minimum number of peaks which must be annotated in an isotopic pattern. The option "Generate Full Isotope Patterns" must be enabled.

2.2.5.8 Minimum Number of Spectra

The minimum number of spectra in which a compound must be identified to be reported.
LC-MS data = the minimum number of consecutive scans;
MSI data = the minimum number of pixels.

2.2.5.9 Minimum Number of Ion Types

The minimum number of ion types which must be matched to report a given compound. Use e.g. Ion Types: [M+H]+, [M+Na]+; Charge: 2; Neutral Losses: H2O; Maximum Number of Combined Losses: 1. If Minimum Ion Types: 2, then any pair of ions from the set of ions [M+H]+, [M+Na]+, [M+2H]2+, [M+Na+H]2+, [M+H-H2O]+, [M+Na-H2O]+, [M+2H-H2O]2+, [M+Na+H-H2O]2+ must be matched to report a given compound.

2.2.5.10 Basic Formula Check

Apply Senior's filtering rules. See https://doi.org/10.1186/1471-2105-8-105 for more details.
In 'Compound Search - MS, LC-MS, MSI' mode, the rules are applied to generated compounds. In other modes, the rules are applied to combinations of neutral losses, see https://doi.org/10.1021/acs.analchem.0c00170 for more details.

2.2.5.11 Calculate FDRs

Calculate FDRs for LC-MS and MSI data (experimental feature). It can optionally be disabled to cut the processing time in half.

2.2.5.12 Minimum Intensity of Highest Peak in Isotopic Pattern

Enter the minimum relative/absolute intensity of the most intense peak in an isotopic pattern. Isotopic patterns with relative/absolute intensities below this value will be kept in the spectrum but not annotated.

2.2.5.13 Isotope m/z Tolerance

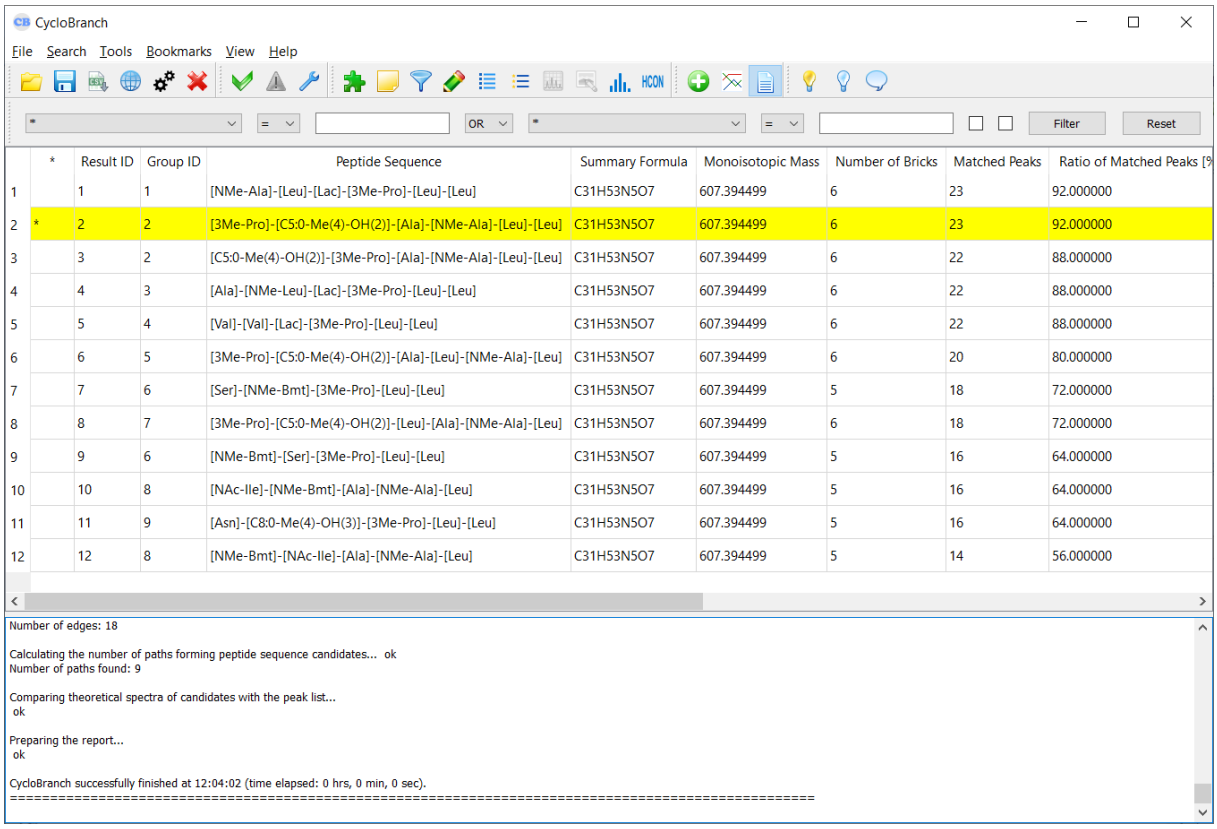
The maximum m/z error of difference between an isotopic peak and the most intense peak in an experimental and theoretical isotopic pattern (0 = disabled) [ppm].
Example, let's have two matched peaks in an isotopic pattern of a compound, an isotopic peak with m/z error 7 ppm and the most intense peak with m/z error 4 ppm:
if m/z Error Tolerance = 10 ppm (i.e. <-10,10>), Isotope m/z Tolerance = 0 ppm (i.e. disabled) => the compound is reported as a correct hit;
if m/z Error Tolerance = 10 ppm (i.e. <-10,10>), Isotope m/z Tolerance = 2 ppm (i.e. <0,2>) => the compound is discarded because 7-4=3 is bigger than 2.

2.2.5.14 Isotope Intensity Tolerance

The maximum error tolerance of intensities of matched isotopes (0% or 100% = disabled) [in % of relative intensity of the most intense peak].
Example:
Isotope Intensity Tolerance = 10%, Relative Intensity of the Most Intense Peak = 100% => the tolerance of relative intensities of isotopes is 10%;
Isotope Intensity Tolerance = 10%, Relative Intensity of the Most Intense Peak = 50% => the tolerance of relative intensities of isotopes is 5%; etc.

2.3 Output Report

If the search is finished, an output report is shown. The report for MS/MS data is shown below. For other data, see Dereplication in MS Data, Dereplication in LC-MS Data, and Dereplication in MSI Data.
Peptide sequence candidates in yellow rows contain a regular expression used in Searched Sequence. Note that the regular expression may be matched in any linearized sequence of a cyclic, branched, branch-cyclic peptide sequence or cyclic polyketide.
The table can be exported into a csv file using the command 'File -> Export to CSV' and then imported, e.g., to MS Excel. It can be also exported into a HTML page using the command 'File -> Export to HTML' including details about peptide sequence candidates. The report can be saved using the command 'File -> Save Results...' and re-opened using the command 'File -> Open Results...' (the files created using an older version of the software cannot be opened in a newer version).



	*	Result ID	Group ID	Peptide Sequence	Summary Formula	Monoisotopic Mass	Number of Bricks	Matched Peaks	Ratio of Matched Peaks [%
1		1	1	[NMe-Ala]-[Leu]-[Lac]-[3Me-Pro]-[Leu]-[Leu]	C31H53N5O7	607.394499	6	23	92.000000
2	*	2	2	[3Me-Pro]-[C5:0-Me(4)-OH(2)]-[Ala]-[NMe-Ala]-[Leu]-[Leu]	C31H53N5O7	607.394499	6	23	92.000000
3		3	2	[C5:0-Me(4)-OH(2)]-[3Me-Pro]-[Ala]-[NMe-Ala]-[Leu]-[Leu]	C31H53N5O7	607.394499	6	22	88.000000
4		4	3	[Ala]-[NMe-Leu]-[Lac]-[3Me-Pro]-[Leu]-[Leu]	C31H53N5O7	607.394499	6	22	88.000000
5		5	4	[Val]-[Val]-[Lac]-[3Me-Pro]-[Leu]-[Leu]	C31H53N5O7	607.394499	6	22	88.000000
6		6	5	[3Me-Pro]-[C5:0-Me(4)-OH(2)]-[Ala]-[Leu]-[NMe-Ala]-[Leu]	C31H53N5O7	607.394499	6	20	80.000000
7		7	6	[Ser]-[NMe-Bmt]-[3Me-Pro]-[Leu]-[Leu]	C31H53N5O7	607.394499	5	18	72.000000
8		8	7	[3Me-Pro]-[C5:0-Me(4)-OH(2)]-[Leu]-[Ala]-[NMe-Ala]-[Leu]	C31H53N5O7	607.394499	6	18	72.000000
9		9	6	[NMe-Bmt]-[Ser]-[3Me-Pro]-[Leu]-[Leu]	C31H53N5O7	607.394499	5	16	64.000000
10		10	8	[NAc-Ile]-[NMe-Bmt]-[Ala]-[NMe-Ala]-[Leu]	C31H53N5O7	607.394499	5	16	64.000000
11		11	9	[Asn]-[C8:0-Me(4)-OH(3)]-[3Me-Pro]-[Leu]-[Leu]	C31H53N5O7	607.394499	5	16	64.000000
12		12	8	[NMe-Bmt]-[NAc-Ile]-[Ala]-[NMe-Ala]-[Leu]	C31H53N5O7	607.394499	5	14	56.000000

Figure 2.7 De novo sequencing - example of output report.

The report contains the following columns when a peptide is being searched:

- $*$ - it indicates that a row is highlighted; you can sort the list by this column to see all highlighted rows at the top
- **Result ID** - a rank of a peptide sequence candidate according to a score defined in Score Type (if the mode "De Novo Search Engine" or "Compare Peaklist with Database - MS/MS data" is used; see Mode)
- **Group ID** - an identifier of a group of peptide sequence candidates generated from the same path in a de novo graph (if the mode "De Novo Search Engine" is used; see Mode)
- **Name** - a name of a peptide (if the mode "Compare Peaklist with Database - MS/MS data" is used; see Mode)
- **Scan ID** - an id of a scan which is used as Scan no. (if the mode "Compare Peaklist(s) with Spectrum of Searched Sequence" is used; see Mode)
- **Title** - a title of an experimental spectrum (if the mode "Compare Peaklist(s) with Spectrum of Searched Sequence" is used; see Mode)
- **Peptide Sequence** - a peptide sequence candidate
- **Summary Formula** - a summary formula of a neutral molecule. When a sequence tag is reported, the formula is incomplete. For example, a formula "C38H69N7O7 (partial)" of a sequence tag "[198.136914]-[↩ Val/Ival]-[Me-Leu]-[Ala/bAla/Sar]-[Ala/bAla/Sar]-[Me-Leu]-[Me-Leu]-[Leu/Ile/NMe-Val]-[268.179095]" does not include the unknown blocks [198.136914] and [268.179095].
- **Monoisotopic mass** - a monoisotopic mass of a neutral modlecule
- **Number of Bricks** - a number of building blocks in a peptide sequence candidate
- **Matched Peaks** - a number of matched peaks between an experimental spectrum and a theoretical spectrum of a peptide sequence candidate
- **Ratio of Matched Peaks [%]** - a ratio of matched peaks to all peaks in an experimental spectrum

- **Sum of Relative Intensities** - a sum of relative intensities of matched peaks
- **Weighted Ratio of Matched Peaks** - a sum of relative intensities of matched peaks / a sum of relative intensities of matched and unmatched peaks
- **Cosine Similarity**
- **A** - the number of a-ions
- **B** - the number of b-ions
- **C** - the number of c-ions
- **X** - the number of x-ions
- **Y** - the number of y-ions
- **Z** - the number of z-ions

Note that the numbers of fragment ions are reported if a desired type of a fragment ion has been selected in Ion Types in Theoretical Spectra. The different types of fragment ions are reported for linear and cyclic polyketides, see Nomenclature of Linear Polyketide Series for details.
The following additional columns are shown if a **linear** or **branched** peptide was searched:

- **N-terminal Modification** - N-terminal modification of the peptide
- **C-terminal Modification** - C-terminal modification of the peptide

The following additional column is shown if a **branched** or **branch-cyclic** peptide was searched:

- **Branch Modification** - a terminal modification of a branch

The following additional column is shown if a **cyclic** peptide was searched:

- **Scrambled Peaks** - a number of matched scrambled peaks between an experimental spectrum and a theoretical spectrum of a peptide sequence candidate

The following additional columns are shown if a **linear polyketide** was searched:

- **Left Terminal Modification** - a terminal modification (N- or C-terminal)
- **Right Terminal Modification** - a terminal modification (N- or C-terminal)

The following columns are shown if MS, LC-MS, and MSI data is processed (see Mode):

- *
- **Spectrum ID** - a number an experimental spectrum when a series of peaklists was processed (e.g., LC-MS or MSI data)
- **Title** - a title of a spectrum
- **Matched Peaks**
- **Ratio of Matched Peaks [%]**
- **Sum of Relative Intensities**
- **Weighted Ratio of Matched Peaks**

The following column is shown if the LC-MS data is processed:

- **Time** - retention time (time units are dataset dependent)

The following columns are shown if the imaging mass spectra are processed:

- **Coordinate X** - x coordinate of a single mass spectrum
- **Coordinate Y** - y coordinate of a single mass spectrum

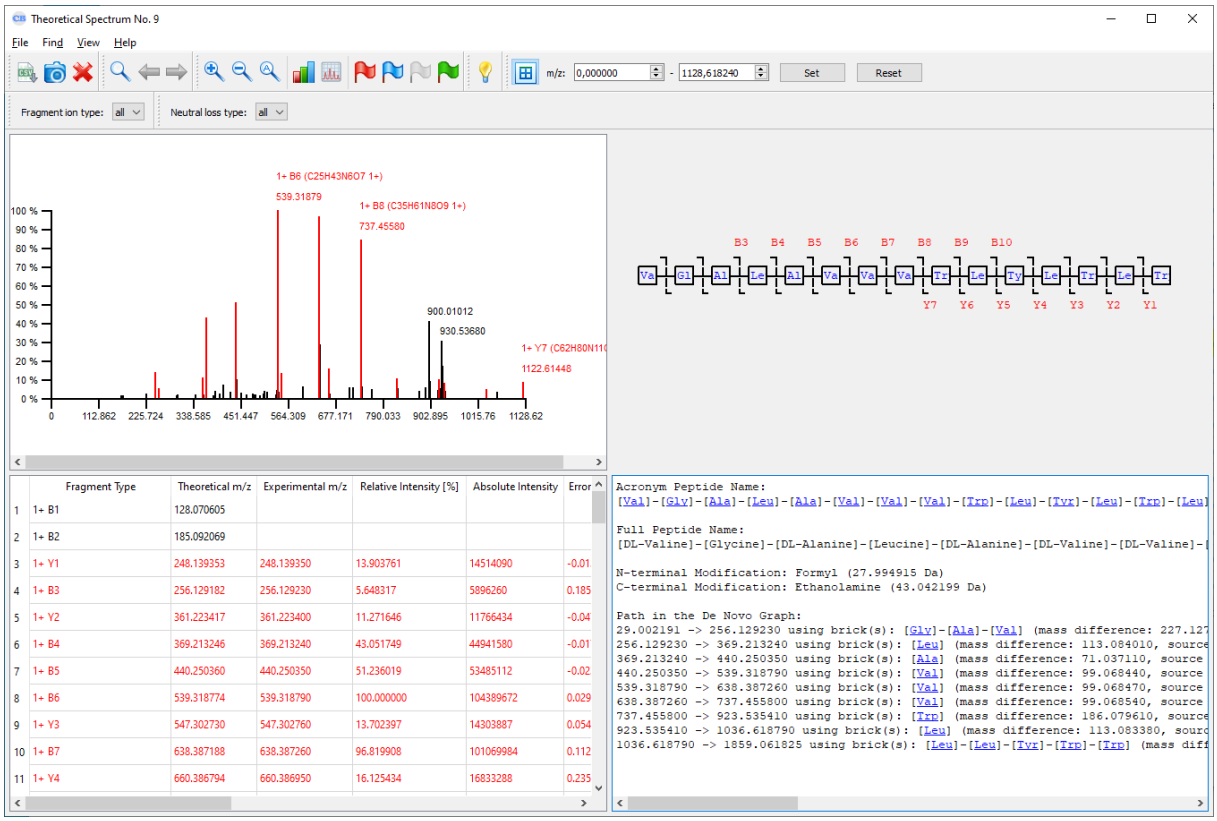
Details about a peptide sequence candidate are shown after a **double click (or Enter key press) on a row** in the output report. See the details on extra pages.

Linear Peptide Detail Window
Cyclic Peptide Detail Window
Branched Peptide Detail Window
Branch-cyclic Peptide Detail Window
Linear Polyketide Detail Window
Cyclic Polyketide Detail Window
Dereplication in MS Data
Dereplication in LC-MS Data
Dereplication in MSI Data

2.4 Linear Peptides

2.4.1 Linear Peptide Detail Window

The window is opened by a double-click on a row in the output report if a linear peptide was identified. The Enter key can also be used if the row is selected. The title of the window contains a "Result ID" of a peptide sequence candidate.



	Fragment Type	Theoretical m/z	Experimental m/z	Relative Intensity [%]	Absolute Intensity	Error
1	1+ B1	128.070605				
2	1+ B2	185.092069				
3	1+ Y1	248.139353	248.139350	13.903761	14514090	-0.01
4	1+ B3	256.129182	256.129230	5.648317	5896260	0.185
5	1+ Y2	361.223417	361.223400	11.271646	11766434	-0.04
6	1+ B4	369.213246	369.213240	43.051749	44941580	-0.01
7	1+ B5	440.250360	440.250350	51.236019	53485112	-0.02
8	1+ B6	539.318774	539.318790	100.000000	104389672	0.029
9	1+ Y3	547.302730	547.302760	13.702397	14303887	0.054
10	1+ B7	638.387188	638.387260	96.819908	101069984	0.112
11	1+ Y4	660.386794	660.386950	16.125434	16833288	0.235

Figure 2.8 De novo sequencing of gramicidin C.

A graphical visualization of the experimental spectrum is shown on the left at the top. Experimental peaks matched with theoretical peaks are red. A graphical visualization of a peptide sequence is shown on the right at the top.
A list of theoretical peaks is shown in the table on the left at the bottom. Following details are available:

- Fragment Type - a name of a theoretical peak including information about its charge
- Theoretical m/z - a theoretical mass-to-charge ratio of the peak
- Theoretical intensity [%] - theoretical intensity of the peak (if Generate Full Isotope Patterns is enabled)

- Experimental m/z - a mass-to-charge ratio of a matched peak
- Relative Intensity [%] - a relative intensity of a matched peak
- Absolute Intensity - an absolute intensity of a matched peak
- Error [ppm] - a ppm error
- Summary Formula - a summary formula of a fragment ion
- Sequence - a sequence of building blocks corresponding to the peak

The theoretical peaks assigned to experimental peaks are red. A list of unassigned experimental peaks is at the bottom of the table. The only items reported for unassigned experimental peaks are "Experimental m/z", "Relative Intensity [%]", and "Absolute intensity".
Following additional details are shown in a text window on the right at the bottom:

- Acronym Peptide Name - a name of the peptide sequence candidate composed from acronyms of building blocks; you can click on an acronym to see the details in ChemSpider, PubChem, etc.
- Full Peptide Name - a name of the peptide sequence candidate composed from full names of building blocks.
- Path in the De Novo Graph - you can see details about a path in the de novo graph from which the peptide sequence candidate was determined.
- Series of matched peaks - the series of matched peaks are shown for each fragment ion type defined in Ion Types in Theoretical Spectra. For each series, a vector of integers is shown, e.g., "`B 0 0 1 1 1 1 1 1 1 1 0 0 0 0`" means that a consecutive series of ions b_3-b_{10} was matched. 0 means that a theoretical fragment ion was not matched. A value greater than '1' can be shown when multiple charge variants of the same theoretical peak were matched.

2.4.2 Toolbar for Linear Peptides

Export Table to CSV (CTRL + E) 
Export the table of peaks into a csv (comma separated values) file.

Export Image (CTRL + G) 
Export the spectrum/peptide image into a PDF, PS (Windows only), PNG or SVG file. PS files in Linux can be generated outside of the application, e.g., using the pdf2ps tool.

Find Text (CTRL + F) 
A find text dialog is opened.

Find Previous (CTRL + 1) 
Move the cursor to the previous search result when a text was found.

Find Next (CTRL + 2) 
Move the cursor to the next search result when a text was found.

Zoom In (CTRL +) 
The window with the graphical visualization of an experimental mass spectrum on the left at the top is zoomed in. A zoom factor in the range $<1, 32>$ is increased by 0.2 when the button is pressed.

Zoom Out (CTRL -) 
The window with the graphical visualization of an experimental mass spectrum is zoomed out. The zoom factor is decreased by 0.2 when the button is pressed.

Zoom Reset (CTRL + R) 
The zoom factor is set up to 1 when the button is pressed.

Absolute Intensity (CTRL + I) 
View absolute/relative intensities of peaks.

Profile Spectrum (CTRL + P) 
View profile spectrum (if applicable). See the list of supported file formats for visualization of profile data. Note that the checkbox next to the file name in the settings dialog must be checked.

Hide Matched Peaks (CTRL + M) 
All experimental peaks which are red (i.e., they have been assigned to theoretical peaks) are hidden in the graphical visualization of an experimental mass spectrum on the left at the top if the button is checked. If the button is unchecked, the peaks are visible. The red peaks in the peaks table are also hidden if the button is checked.

Hide Unmatched Peaks (CTRL + U) 
All experimental peaks which were not assigned to theoretical peaks are hidden if the button is checked. If the button is unchecked, the peaks are visible. The unmatched theoretical and experimental peaks in the peaks table are also hidden if the button is checked.

Hide Peak Labels (CTRL + L) 
The labels of peaks are hidden if the button is checked.

Mouse m/z Selection Tool (CTRL + T) 
If the button is enabled, the mouse can be used to select a range of m/z ratios to be shown.

- Left mouse button - press and hold the button to draw a rectangle determining the minimum and maximum values of m/z ratios; release the button to apply the changes.
- Right mouse button - press the button during the drawing of a rectangle to cancel the drawing.
- Middle mouse button - reset the range of m/z ratios.
- Mouse wheel up - zoom the interval in.
- Mouse wheel down - zoom the interval out.

If the button is disabled, the mouse can be used to shift the interval of m/z ratios to the left or right. Press and hold the left mouse button to shift the spectrum.
m/z

- First field - a minimum value of m/z ratio to be shown.
- Second field - a maximum value of m/z ratio to be shown.
- "Set" button (or Enter key) - apply the changes.
- "Reset" button - reset the range of m/z ratios.

Fragment ion type
A dropdown menu where you can choose which fragment ion type will be reported in the spectrum, table of peaks and graphical visualization of the peptide on the right.
Neutral loss type
A dropdown menu where you can choose which neutral loss type will be reported in the spectrum, table of peaks and graphical visualization of the peptide on the right.

2.4.3 Linear Sequence Detection

A path in a de novo graph corresponding to a linear peptide sequence is detected as shown in the following scheme, i.e, from H^+ to precursor ion for b-series or from H_3O^+ to precursor for y-series. An edge corresponds to a residue mass of a building block or to a sum of residue masses of a combination of building blocks. If an edge corresponds to multiple building blocks, peptide sequence candidates with all permutations of building blocks are generated from a path in a de novo graph. For example, if the edge corresponds to a combination of blocks leucine, proline and

valine, the order of blocks can be LPV, LVP, PVL, PLV, VLP, and VPL; and thus 6 peptide sequence candidates are generated.

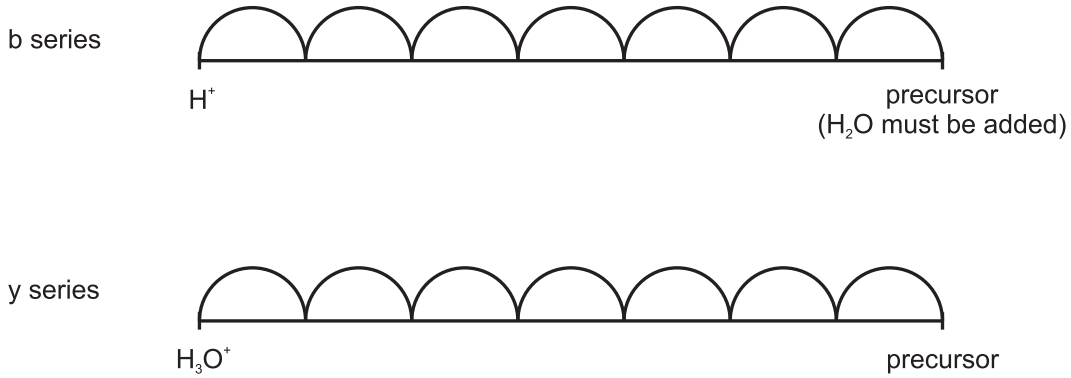


Figure 2.9 Detection of a path corresponding to a linear peptide in a de novo graph.

2.5 Cyclic Peptides

2.5.1 Cyclic Peptide Detail Window

The window is opened by a double-click on a row in the output report if a cyclic peptide was identified. The Enter key can also be used if the row is selected.

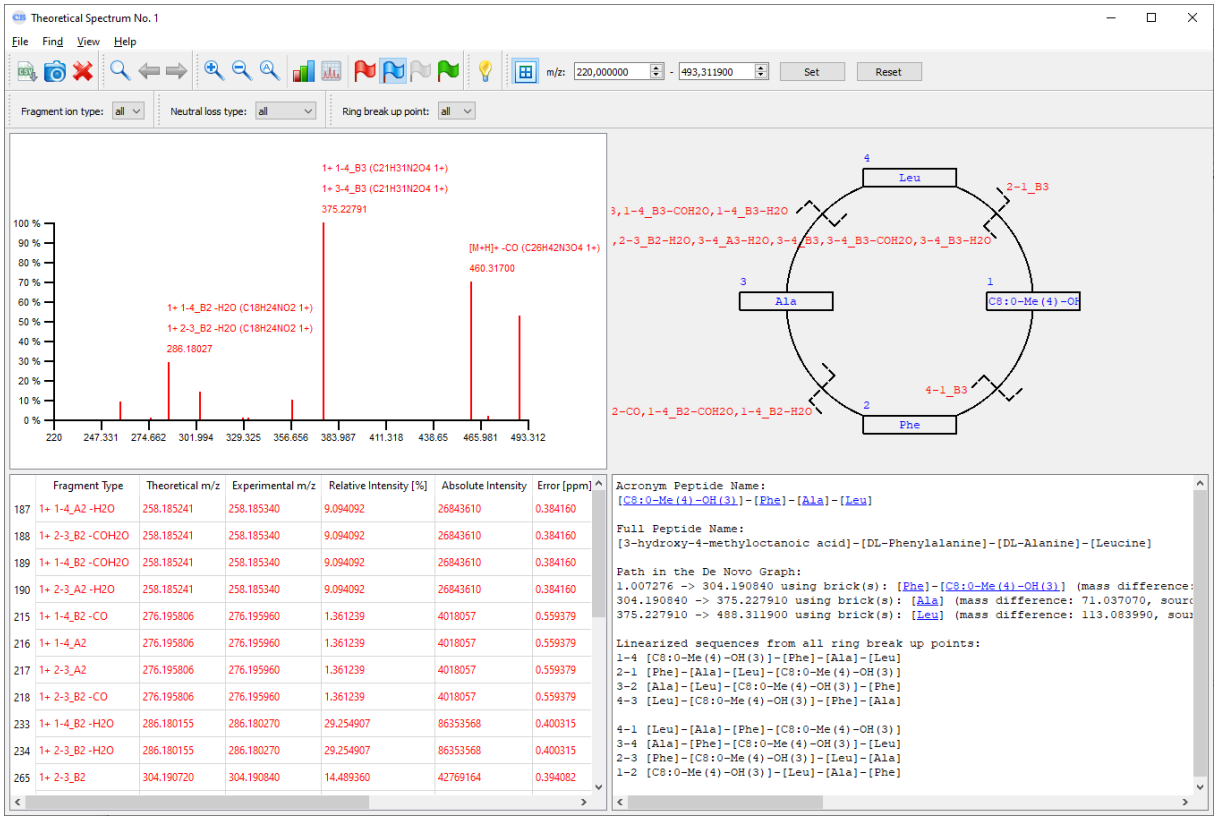


Figure 2.10 De novo sequencing of beauverolide I.

See also Linear Peptide Detail Window. The differences for cyclic peptides are following:

- The text window on the right at the bottom contains a list of all linearized sequences derived from all possible ring break up points of a cyclic sequence.
- A name of a fragment ion is prefixed with an identifier of a linearized peptide sequence from which the fragment ion was generated (e.g., 3-4_A1 means that the fragment ion a_1 was generated from the linearized sequence 3-4).

2.5.2 Toolbar for Cyclic Peptides

See also Toolbar for Linear Peptides. The differences for cyclic peptides are following:

Hide Scrambled Peaks (CTRL + S) 

The function is available if a cyclic peptide was searched and if the checkbox Enable Scrambling was checked. The experimental peaks which were assigned to scrambled theoretical peaks of a cyclic peptide are black if the button is checked. If the button is unchecked, the peaks are red. The scrambled peaks in the peaks table are also hidden if the button is checked.

Ring break up point

A dropdown menu where you can choose which peaks are red in the spectrum on the left and which fragment ions are shown in the visualization of a cyclic peptide on the right. You can choose that only theoretical peaks from one linearized sequence (e.g., 3-4) are red if they are matched with experimental peaks. In this case, the graphical visualization on the right shows only matched fragment ions from the linearized sequence 3-4 and the break up point 3-4 is red. The default value "all" means that all matched theoretical peaks on the left are red and that all matched fragment ions are shown on the right.

2.5.3 Cyclic Sequence Detection

A path in the de novo graph corresponding to a b-ion series of a cyclic peptide is detected as shown in the following scheme. An edge corresponds to a residue mass of a building block or to a sum of residue masses of a combination of building blocks. If an edge corresponds to multiple building blocks, peptide sequence candidates with all permutations of building blocks are generated from a path in a de novo graph. For example, when the edge corresponds to a combination of blocks leucine, proline and valine, the order of blocks can be LPV, LVP, PVL, PLV, VLP, and VPL; and thus 6 peptide sequence candidates are generated.

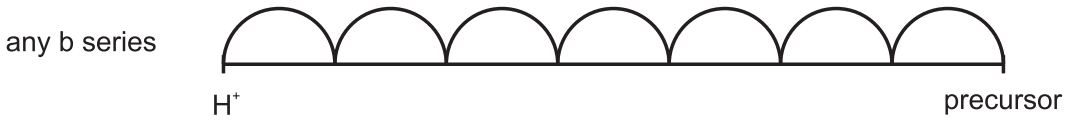


Figure 2.11 Detection of a path corresponding to a cyclic peptide in a de novo graph.

2.5.4 Nomenclature of Cyclic Series

The names of linearized peptide sequences are derived as shown in the following figure on a showcase example of roseotoxin A. Since the peptide is composed from six building blocks, twelve series of b-ions are generated. The numbers in superscripts of the names of b-ions indicate ring opening sites (e.g., $^{1\text{-}6}b_2$ means the ion b_2 arise after ring opening between leucic acid and betaalanine). The first number in the superscript corresponds to the N-terminal building block (leucic acid) and the second number to the C-terminal block (betaalanine). In case of reverted linearized sequences, the first number denotes a C-terminal block and the second number an N-terminal block.

- The ion $^{i\text{-}j}b$ corresponding to a linearized sequence composed from k building blocks has
  - $j = i - 1$ if $i > 1$ and $j < k$;
  - $i = 1$ and $j = k$, otherwise.
- The ion $^{i\text{-}j}b$ corresponding to a reverted linearized sequence composed from k building blocks has
  - $i = j - 1$ if $i < k$ and $j > 1$;
  - $i = k$ and $j = 1$, otherwise.

Note that each fragment ion occurs twice in the scheme (e.g., the fragment ion $^{1\text{-}6}b_2$ is identical with $^{2\text{-}3}b_2$).

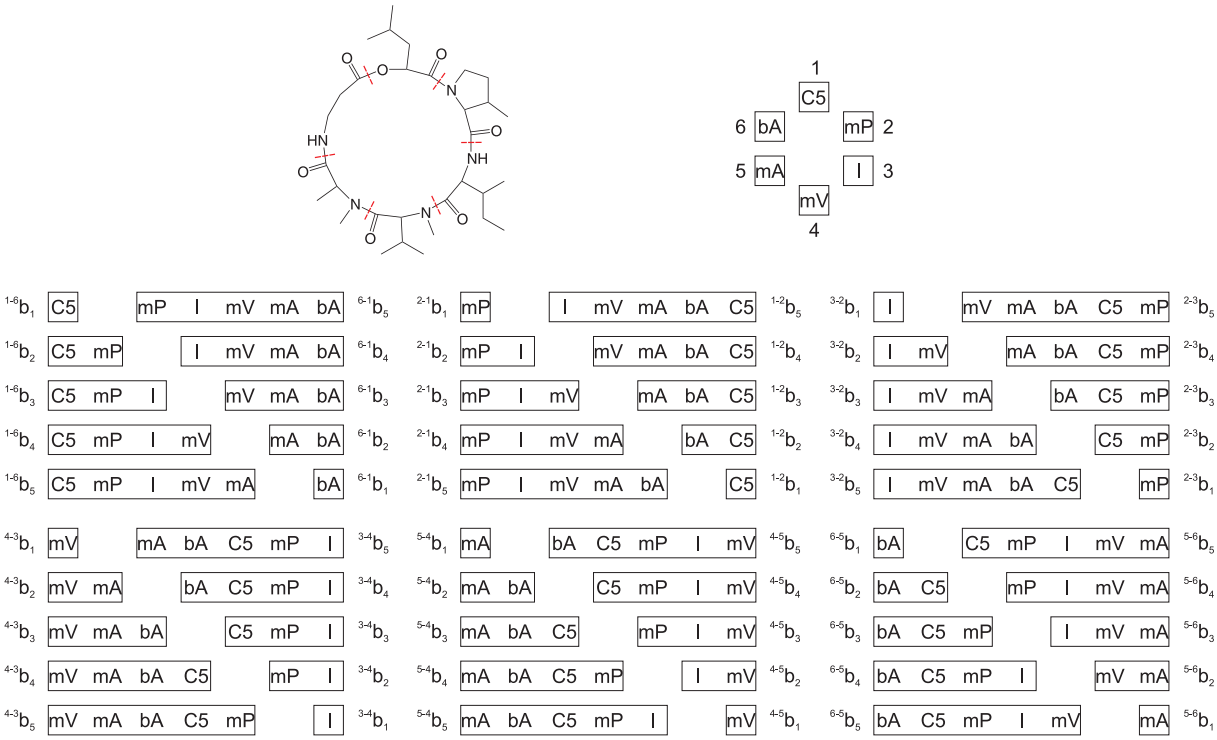


Figure 2.12 Principle of numbering of fragment ions series of roseotoxin A (C5 = leucic acid; mP = 3-methylproline; I = isoleucine; mV = methylvaline; mA = methylalanine; bA = betaalanine).

2.6 Branched Peptides

2.6.1 Branched Peptide Detail Window

The window is opened by a double-click on a row in the output report if a branched peptide was identified. The Enter key can also be used if the row is selected.

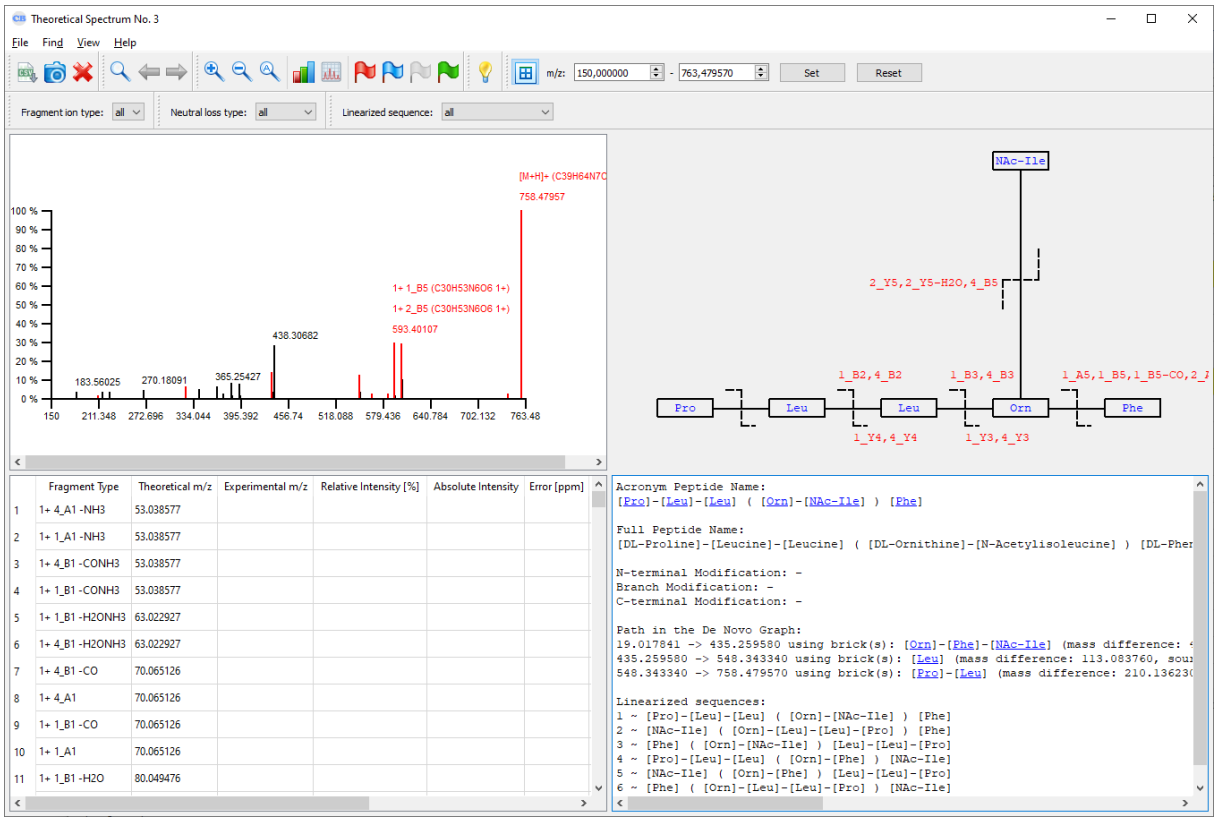


Figure 2.13 De novo sequencing of linearized pseudacyclin A.

See also Linear Peptide Detail Window. The differences for branched peptides are following:

- The text window on the right at the bottom contains a list of all six linearized sequences derived from a singly branched peptide sequence.
- A name of a fragment ion is prefixed with a number of a linearized sequence from which the fragment ion was generated (e.g., 2_B1 means that the fragment ion b_1 was generated from the linearized sequence no. 2).

Numbers of linearized sequences of a branched peptide sequence are defined as follows (with respect to the visualization of a branched peptide):

- sequence no. 1 - starts on the left, ends on the right;
- sequence no. 2 - starts at the top, ends on the right;
- sequence no. 3 - starts on the right, ends on the left;
- sequence no. 4 - starts on the left, ends at the top;
- sequence no. 5 - starts at the top, ends on the left;
- sequence no. 6 - starts on the right, ends at the top.

The start point of a linearized sequence corresponds to a theoretical N-terminus. The end point corresponds to a theoretical C-terminus.

2.6.2 Toolbar for Branched Peptides

See also Toolbar for Linear Peptides. The differences for branched peptides are following:

Linearized sequence

A dropdown menu is shown where you can choose which peaks are red in the spectrum of the left and which fragment ions are shown in the visualization of a branched peptide on the right. You can choose that only theoretical peaks from one linearized sequence (e.g., 2) are red if they were matched with experimental peaks. In this case, the visualization of the peptide on the right shows only matched fragment ions from the linearized sequence 2. The default value "all" means that all matched theoretical peaks are red in the spectrum on the left and that all matched fragment ions are shown in the visualization of the peptide on the right.

2.6.3 Branched Sequence Detection

Series of fragment ions of a branched peptide are shown in the following picture.

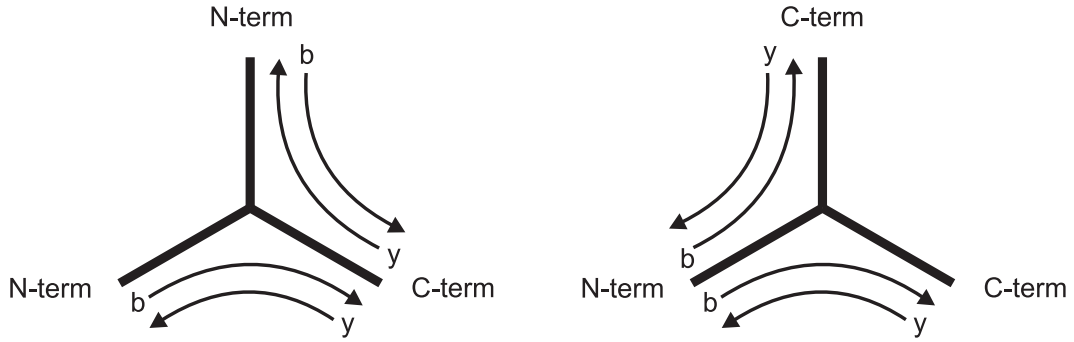


Figure 2.14 Theoretical fragmentation of a branched peptide.

A path in the de novo graph corresponding to a branched peptide is detected as is shown in the following scheme (an edge corresponds to a residue mass of a building block or a sum of residue masses of a combination of building blocks). An edge corresponding to the branch must include at least two building blocks (one block cannot form a branch).

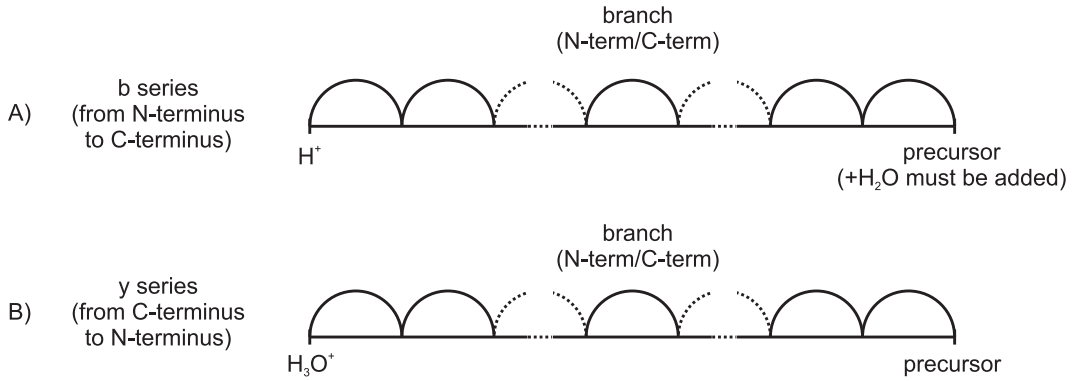


Figure 2.15 Detection of a path corresponding to a branched peptide in a de novo graph.

Since peaks determining the begin and the end of a branch may be missing in an experimental spectrum, the algorithm generates a set of peptide sequence candidates from a path in the graph instead of one sequence candidate. See the following scheme.

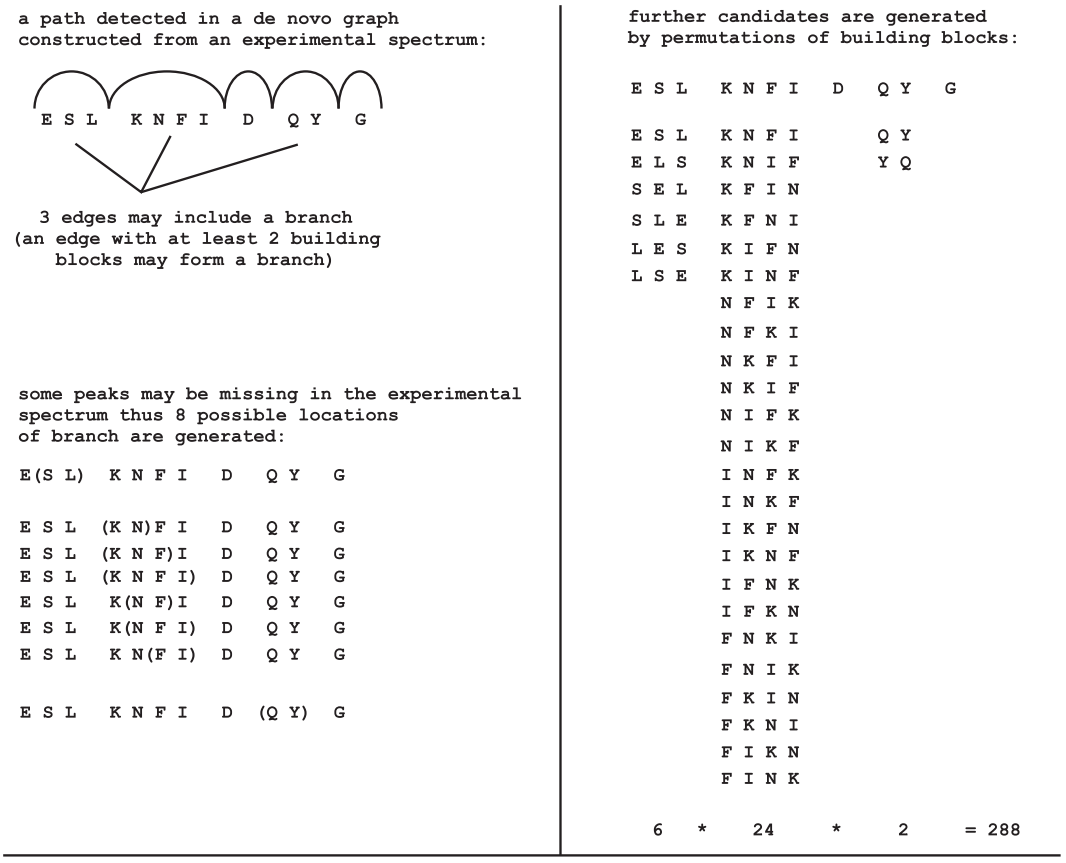
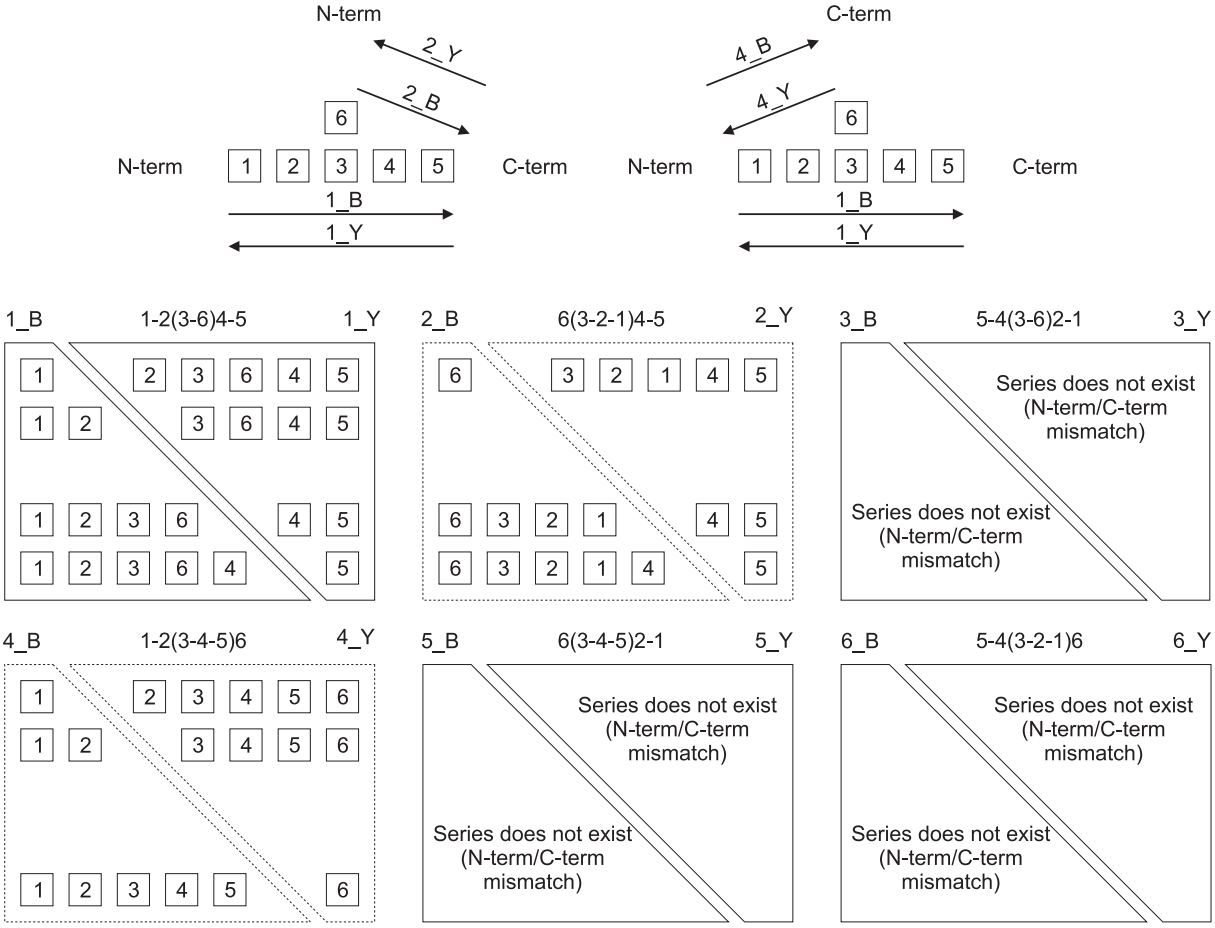


Figure 2.16 Peptide sequence candidates generated from a path in a de novo graph (branched peptides).

2.6.4 Nomenclature of Branched Series

In the following picture, you can see a general scheme of a fragmentation of a singly branched peptide. A sequence of a branched peptide can be written by six different ways (called linearized sequences of a branched peptide sequence). Two series of ions can be generated from each linearized sequence (N-terminal and C-terminal), thus we get series of fragment ions ^{1}B to ^{6}B and ^{1}Y to ^{6}Y. However, the series ^{3}B, ^{5}B and ^{6}B are unsense because N-terminal fragments cannot be generated from a C-terminal side. Similarly, the series ^{3}Y, ^{5}Y and ^{6}Y are unsense because C-terminal fragments cannot be generated from an N-terminal side. The series ^{2}B and ^{2}Y exist only when the branch is N-terminal. The series ^{4}B and ^{4}Y exist only when the branch is C-terminal. Thus for a branched peptide, only four series of fragment ions can be observed in total - two series of b-ions (N-terminal ions) and two series of y-ions (C-terminal ions).



Note:
The series 2_B and 2_Y exist only when the branch is N-terminal.
The series 4_B and 4_Y exist only when the branch is C-terminal.

Figure 2.17 General scheme of a fragmentation of a branched peptide.

2.6.5 Fragmentation Example

A scheme of theoretical fragments generated from a linearized pseudacyclin A is shown in the following picture. We assume that a ion of a branched peptide is fragmented only once and that the branch is not fragmented, thus some b-ions and y-ions do not exist in the theoretical spectrum (struck out in the following picture). If a branch was split, the ion would be fragmented for second time.

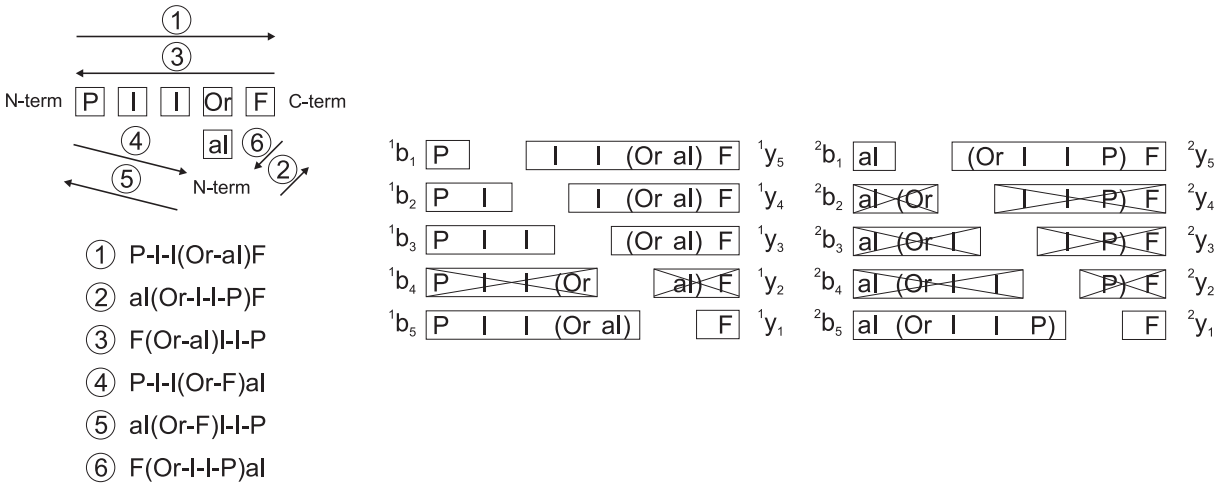


Figure 2.18 Theoretical fragments of linearized pseudacyclin A and its six linearized sequences (P = proline; I = isoleucine, Or = ornithine; al = acetylisoleucine; F = phenylalanine).

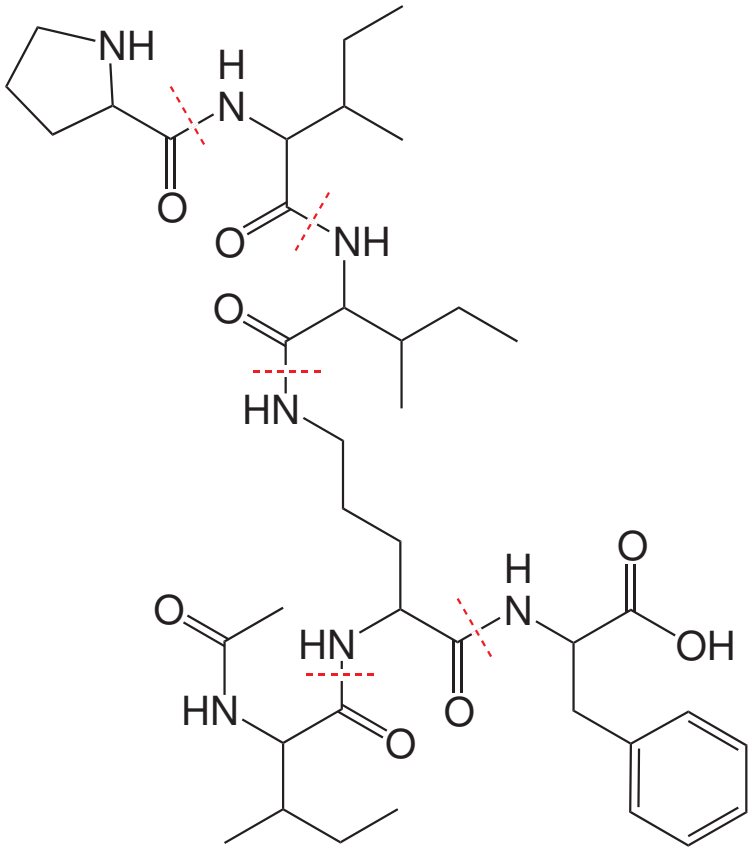


Figure 2.19 Linearized pseudacyclin A.

2.6.6 Branched Siderophores

CycloBranch supports the identification of desferri- and ferri-forms of branched siderophores which are nonribosomal peptides. For details, see Linear Nonribosomal Peptide Siderophores.

2.7 Branch-cyclic Peptides

2.7.1 Branch-cyclic Peptide Detail Window

The window is opened by a double-click on a row in the output report if a branch-cyclic peptide was identified. The Enter key can also be used if the row is selected.

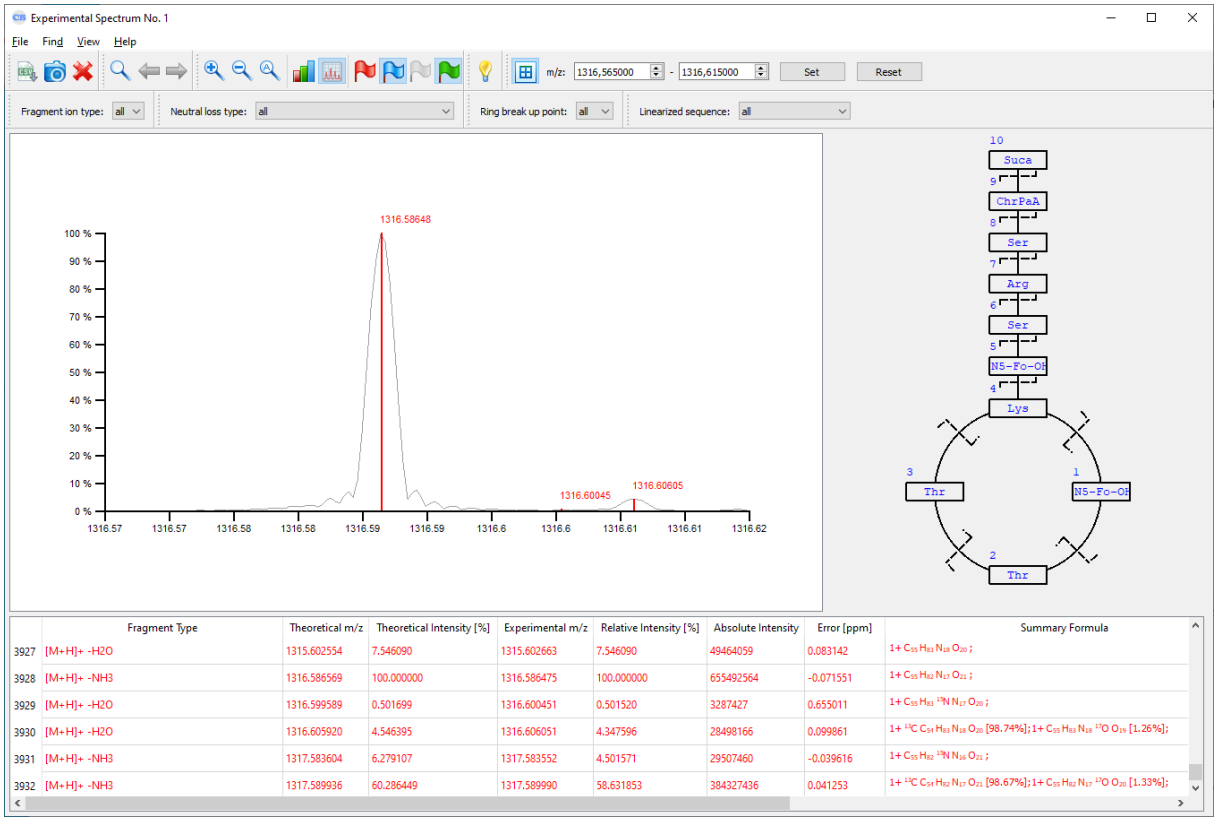


Figure 2.20 Direct comparison of experimental spectrum of pyoverdin E with the theoretical spectrum.

See details in Linear Peptide Detail Window, Cyclic Peptide Detail Window and Branched Peptide Detail Window. The differences for branch-cyclic peptides are following:

- The text window on the right at the bottom (data not shown in the image above) contains a list of all linearized sequences derived from all possible ring opening sites of a branch-cyclic peptide. If the ring is opened, a fragment ion corresponds to a branched peptide. Thus for each linearized sequence (derived from the ring opening), another list of linearized sequences is generated (derived from a branched fragment). The nomenclature of linearized sequences of branch-cyclic peptides simply combines nomenclatures of linearized sequences derived from cyclic and branched peptide sequences. For example, 1-4_5 is a linearized sequence no. 5 of a branched fragment ion arising after a ring opening at the position 1-4.

2.7.2 Toolbar for Branch-cyclic Peptides

See Toolbar for Linear Peptides, Toolbar for Cyclic Peptides and Toolbar for Branched Peptides.

2.7.3 Branch-cyclic Sequence Detection

Series of fragment ions of a branch-cyclic peptide are shown in the following picture.

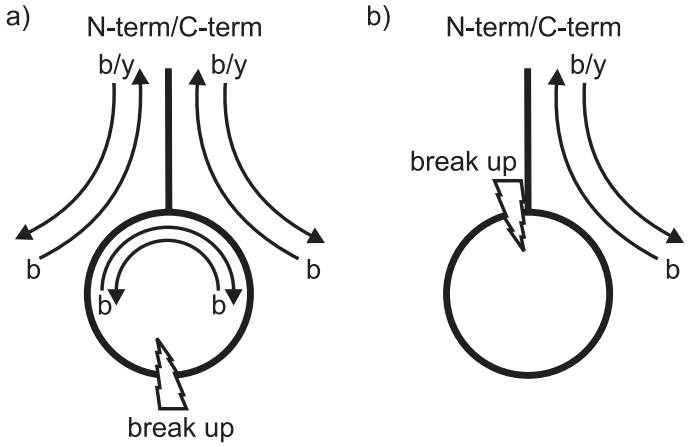


Figure 2.21 Theoretical fragment ion series generated due to ring opening of a branch-cyclic NRP - (a) a branched fragment, (b) a linear fragment.

A path in the de novo graph corresponding to a branch-cyclic peptide is detected as shown in the following scheme (an edge corresponds to a residue mass of a building block or a sum of residue masses of a combination of building blocks). The options E) and I) are currently not implemented.

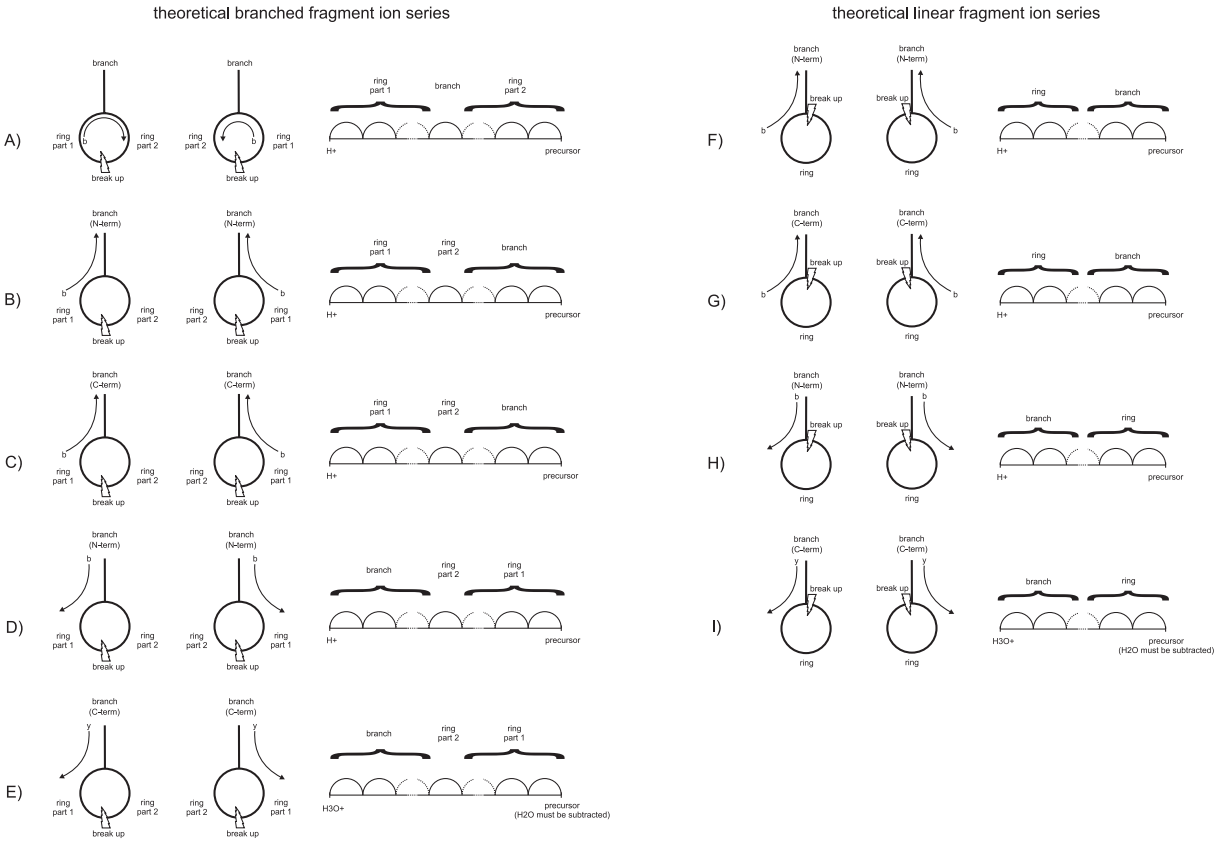


Figure 2.22 Detection of a path corresponding to a branch-cyclic peptide in a de novo graph.

Since peaks determining the begin and the end of a branch may be missing in an experimental spectrum, the algorithm generates a set of peptide sequence candidates from a path in the graph instead of one sequence candidate. See the following scheme and Branched Sequence Detection for details.

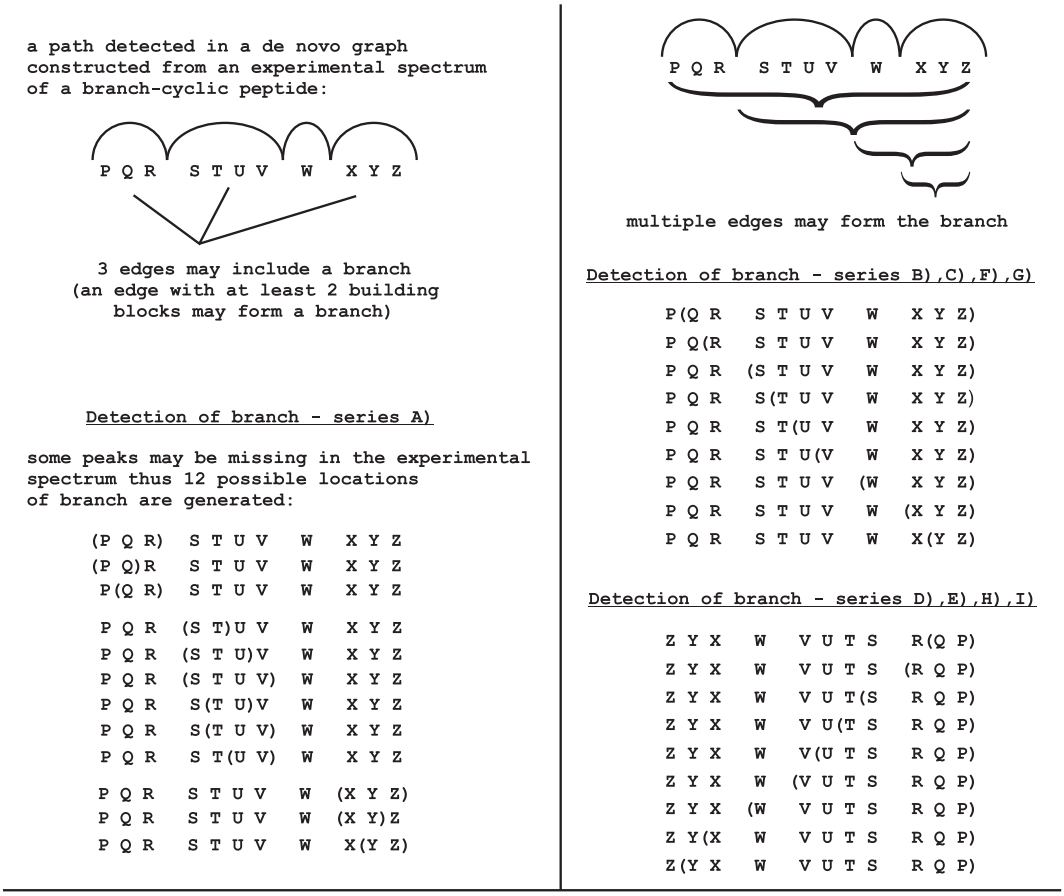


Figure 2.23 Peptide sequence candidates generated from a path in a de novo graph (branch-cyclic peptides).

2.7.4 Nomenclature of Branch-cyclic Series

Since branch-cyclic peptides combine properties of cyclic and branched peptides, the nomenclature of branch-cyclic peptides combines nomenclatures of both types of peptides.

2.7.4.1 Linearized Sequences (Derived from a Ring Opening)

The sequences are derived the same way like linearized sequences of a cyclic peptide with a difference that all building blocks between "(" and ")" are considered as one building block. For example, linearized sequences of pseudacyclin A are generated as follows:
Linearized sequences:
1-5 ∼ [Phe]-[Pro]-[Ile]-[Ile] ([Orn]-[N-Ac-Ile])
2-1 ∼ [Pro]-[Ile]-[Ile] ([Orn]-[N-Ac-Ile]) [Phe]
3-2 ∼ [Ile]-[Ile] ([Orn]-[N-Ac-Ile]) [Phe]-[Pro]
4-3 ∼ [Ile] ([Orn]-[N-Ac-Ile]) [Phe]-[Pro]-[Ile]
5-4 ∼ ([Orn]-[N-Ac-Ile]) [Phe]-[Pro]-[Ile]-[Ile]
Reverted linearized sequences:
5-1 ∼ ([Orn]-[N-Ac-Ile]) [Ile]-[Ile]-[Pro]-[Phe]
4-5 ∼ [Ile]-[Ile]-[Pro]-[Phe] ([Orn]-[N-Ac-Ile])
3-4 ∼ [Ile]-[Pro]-[Phe] ([Orn]-[N-Ac-Ile]) [Ile]
2-3 ∼ [Pro]-[Phe] ([Orn]-[N-Ac-Ile]) [Ile]-[Ile]
1-2 ∼ [Phe] ([Orn]-[N-Ac-Ile]) [Ile]-[Ile]-[Pro]

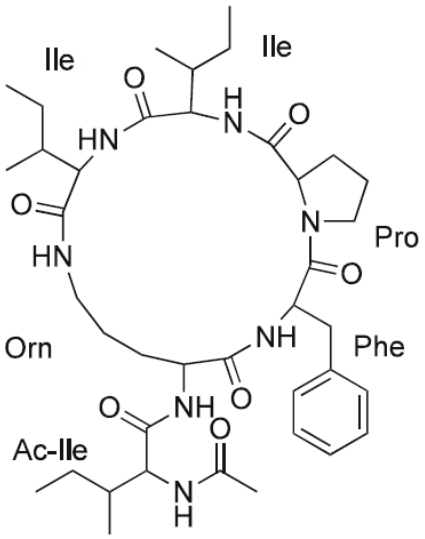
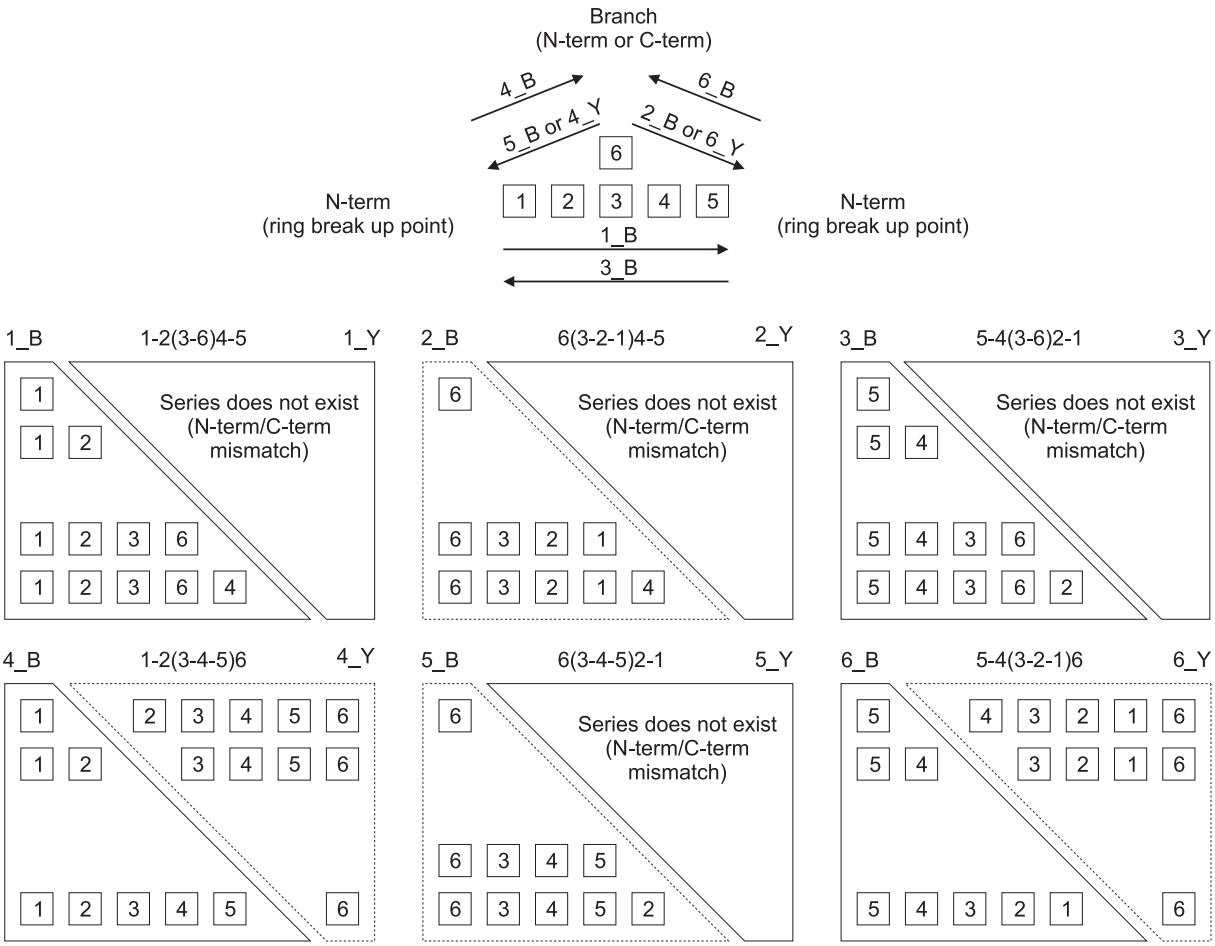


Figure 2.24 Pseudacyclin A.

Each branch-cyclic peptide may form up to *k - 2* branched fragments (or *2∗(k - 2)* when reverted fragments are considered) where *k* is the number of monomers forming the ring. Each branch-cyclic peptide may also produce up to 2 linear fragments (or 4 when reverted fragments are considered). These fragments are indicated by the brackets "(...)" on the left or on the right side of a linearized sequence. Linear fragments of pseudacyclin A are following:
Sequences:
1-5 ∼ [Phe]-[Pro]-[Ile]-[Ile] ([Orn]-[N-Ac-Ile]) corresponds to a linearized sequence [Phe]-[Pro]-[Ile]-[Ile]-[Orn]-[N-↩ Ac-Ile]
5-4 ∼ ([Orn]-[N-Ac-Ile]) [Phe]-[Pro]-[Ile]-[Ile] corresponds to a linearized sequence [N-Ac-Ile]-[Orn]-[Phe]-[Pro]-[Ile]-[Ile]
Reverted sequences:
5-1 ∼ ([Orn]-[N-Ac-Ile]) [Ile]-[Ile]-[Pro]-[Phe] corresponds to a linearized sequence [N-Ac-Ile]-[Orn]-[Ile]-[Ile]-[Pro]-[Phe]
4-5 ∼ [Ile]-[Ile]-[Pro]-[Phe] ([Orn]-[N-Ac-Ile]) corresponds to a linearized sequence [Ile]-[Ile]-[Pro]-[Phe]-[Orn]-[N-↩ Ac-Ile]

2.7.4.2 Linearized Sequences (Derived from a Branched Fragment)

Linearized sequences of a branched fragment are derived similarly to linearized sequences of a branched peptide (see a general fragmentation scheme below). Because of a cyclization in a branch-cyclic peptide, the C-terminal series ^{1}Y and ^{2}Y are replaced with the N-terminal series ^{3}B and ^{6}B.



Note:
The series 2_B and 5_B exist only when the branch is N-terminal.
The series 4_Y and 6_Y exist only when the branch is C-terminal.

Figure 2.25 General scheme of linearized sequences and fragmentation of a branched fragment 1-2(3-6)4-5.

2.7.4.3 Example of Linearized Sequences of a Branched Fragment

Example of linearized sequences of a branched fragment `[Pro]-[Ile]-[Ile]([Orn]-[N-Ac-↩ Ile])[Phe]` derived from a branch-cyclic peptide pseudacyclin A:

2-1_1 ∼ [Pro]-[Ile]-[Ile] ([Orn]-[N-Ac-Ile]) [Phe]
2-1_2 ∼ [N-Ac-Ile] ([Orn]-[Ile]-[Ile]-[Pro]) [Phe]
2-1_3 ∼ [Phe] ([Orn]-[N-Ac-Ile]) [Ile]-[Ile]-[Pro]
2-1_4 ∼ [Pro]-[Ile]-[Ile] ([Orn]-[Phe]) [N-Ac-Ile]
2-1_5 ∼ [N-Ac-Ile] ([Orn]-[Phe]) [Ile]-[Ile]-[Pro]
2-1_6 ∼ [Phe] ([Orn]-[Ile]-[Ile]-[Pro]) [N-Ac-Ile]

The same example is shown in the picture below where the fragment ion series are illustrated for each linearized sequence. We assume that a branched fragment is further fragmented only once, thus some b-ions do not exist in the theoretical spectrum (struck out in the following picture).

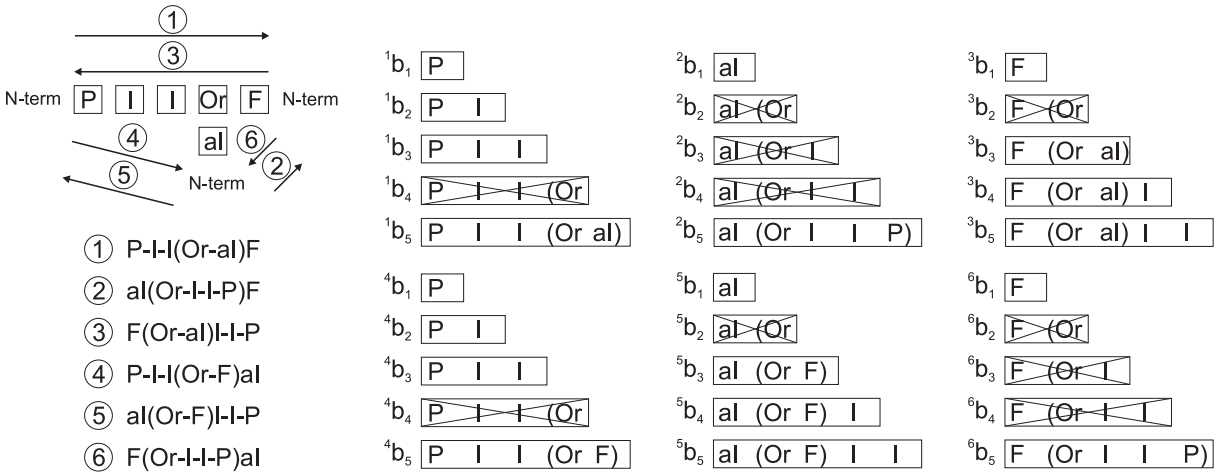


Figure 2.26 Linearized sequences of a branched fragment of pseudacyclin A (P = proline; I = isoleucine, Or = ornithine; al = acetylisoleucine; F = phenylalanine).

2.7.4.4 Example of Linearized Sequences of a Linear Fragment

A linear fragment (e.g., `[Phe]-[Pro]-[Ile]-[Ile]([Orn]-[N-Ac-Ile])`) may also be obtained instead of a branched one after a ring opening. In this case, the notation of linearized sequences can also be understood as follows:

1-5_1 ∼ [Phe]-[Pro]-[Ile]-[Ile] ([Orn]-[N-Ac-Ile]) corresponds to [Phe]-[Pro]-[Ile]-[Ile]-[Orn]-[N-Ac-Ile]
1-5_2 ∼ [N-Ac-Ile] ([Orn]-[Ile]-[Ile]-[Pro]-[Phe]) corresponds to [N-Ac-Ile]-[Orn]-[Ile]-[Ile]-[Pro]-[Phe]
1-5_3 ∼ ([Orn]-[N-Ac-Ile]) [Ile]-[Ile]-[Pro]-[Phe] corresponds to [N-Ac-Ile]-[Orn]-[Ile]-[Ile]-[Pro]-[Phe]
1-5_4 ∼ [Phe]-[Pro]-[Ile]-[Ile] ([Orn]) [N-Ac-Ile] corresponds to [Phe]-[Pro]-[Ile]-[Ile]-[Orn]-[N-Ac-Ile]
1-5_5 ∼ [N-Ac-Ile] ([Orn]) [Ile]-[Ile]-[Pro]-[Phe] corresponds to [N-Ac-Ile]-[Orn]-[Ile]-[Ile]-[Pro]-[Phe]
1-5_6 ∼ ([Orn]-[Ile]-[Ile]-[Pro]-[Phe]) [N-Ac-Ile] corresponds to [Phe]-[Pro]-[Ile]-[Ile]-[Orn]-[N-Ac-Ile]

2.7.5 Branch-cyclic Siderophores

CycloBranch supports the identification of desferri- and ferri- forms of branch-cyclic siderophores which are nonribosomal peptides. For details, see Linear Nonribosomal Peptide Siderophores.

2.8 Linear Siderophores

2.8.1 Linear Nonribosomal Peptide Siderophores

There are no special requirements for identification of desferri-forms of linear siderophores which are nonribosomal peptides (see Linear Peptides). However, the Precursor Ion Adduct must be set up for identification of ferri-forms of siderophores. The peaks of isotopes having natural abundances greater than 3% are generated automatically in theoretical peaklists next to the peaks of most abundant isotopes. If a metal is included in the precursor ion adduct field, the generated theoretical fragments are suffixed with its name and corresponding theoretical isotope fragments are generated (e.g., the fragments B4_Fe and B4_54Fe are generated instead of B4 when precursor ion adduct "FeH-2" is used).

2.8.2 Linear Polyketide Detail Window

The identification of linear polyketides requires an additional annotation of building blocks. See Format of Ketide Building Blocks. The window is opened by a double-click on a row in the output report if a linear polyketide was identified. The Enter key can also be used if the row is selected. See also Linear Peptide Detail Window and Toolbar for Linear Peptides.

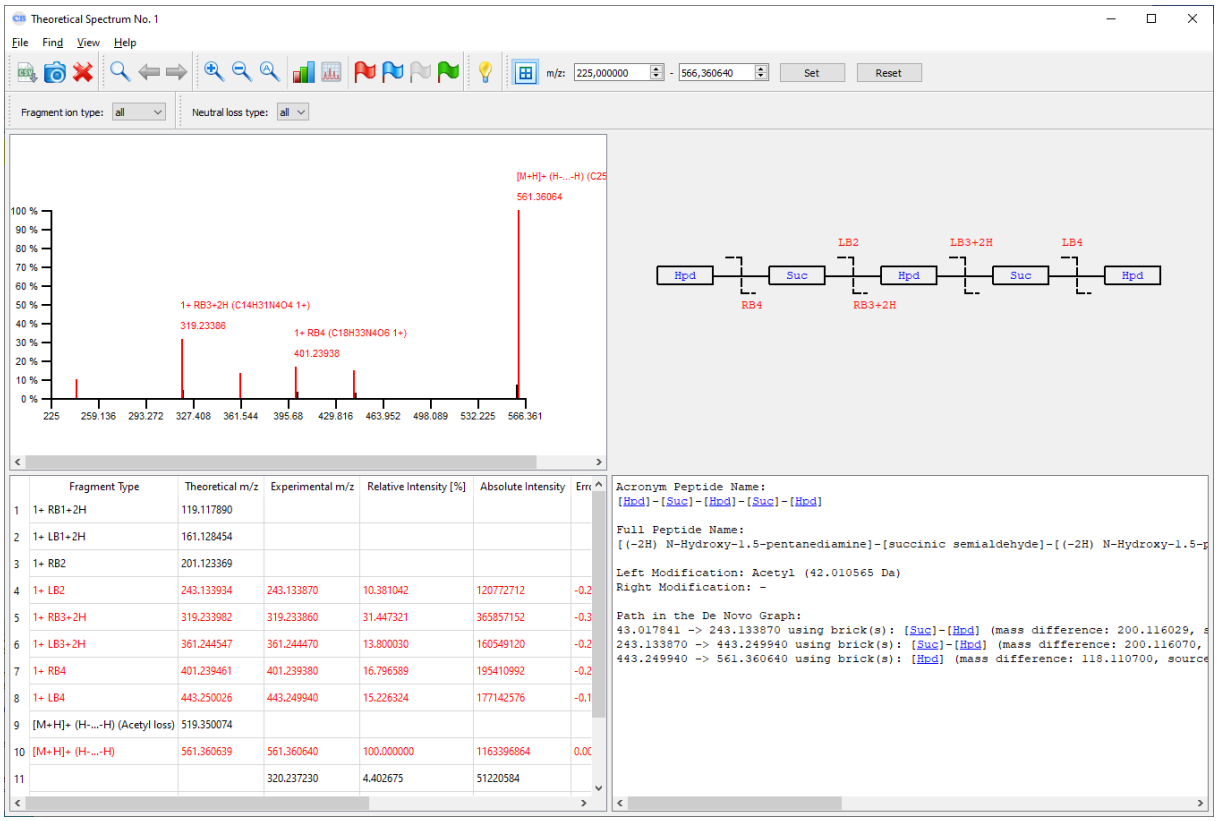


Figure 2.27 De novo sequencing of desferri-ferrioxamine B.

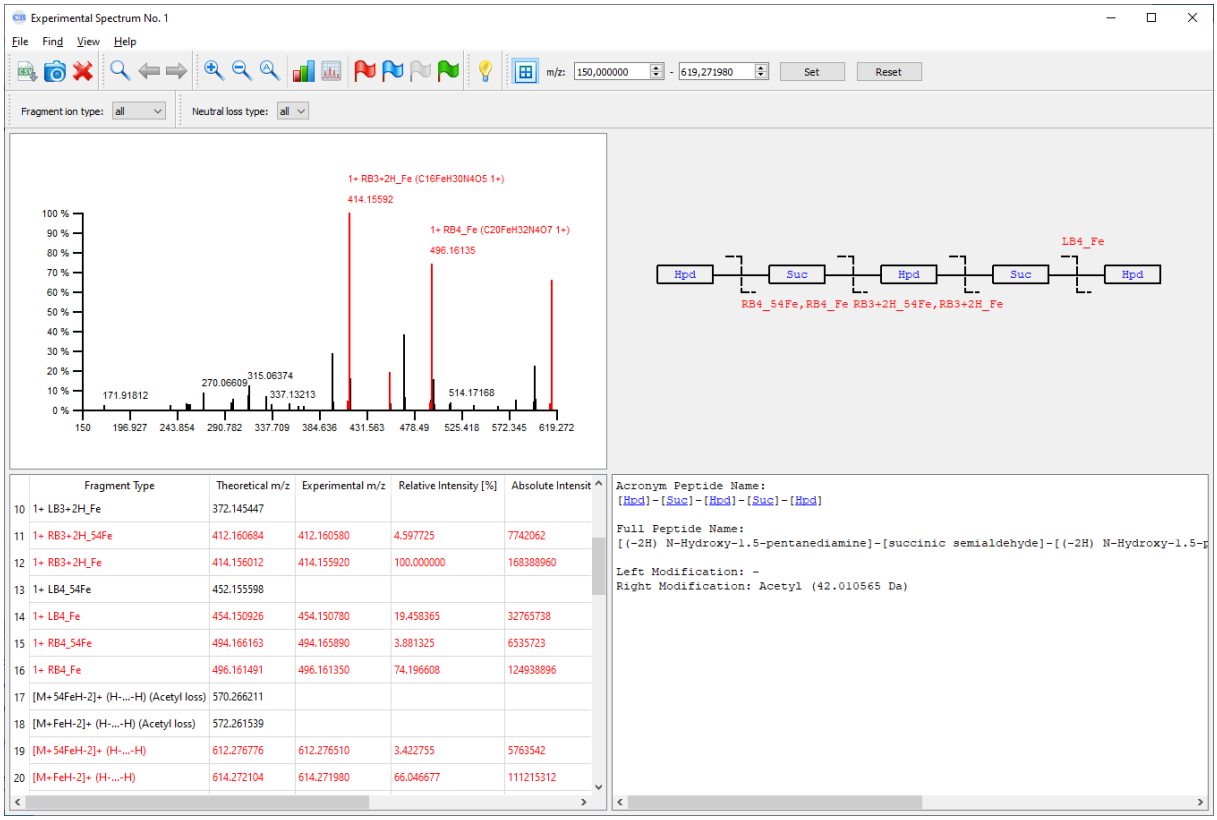


Figure 2.28 Sequencing of ferri-ferrioxamine B.

2.8.3 Nomenclature of Linear Polyketide Series

A polyketide is composed of a sequence of ketide building blocks which eliminate hydrogen atoms from both termini or eliminate a molecule of water. A linear polyketide can have N-terminus and C-terminus; but also two N-termini or two C-termini. CycloBranch generates 8 basic series of fragment ions for linear polyketides (L and R stand for left and right, respectively):

- ^{L}b (left b-ion)
- ^{L}b + 2H (left b-ion + 2H)
- ^{R}b (right b-ion)
- ^{R}b + 2H (right b-ion + 2H)
- ^{L}y (left y-ion)
- ^{L}y - 2H (left y-ion - 2H)
- ^{R}y (right y-ion)
- ^{R}y - 2H (right y-ion - 2H)

If a metal is included in the precursor ion adduct field, the generated theoretical fragments are suffixed with its name and corresponding theoretical isotope fragments are generated (e.g., the fragments LB3+2H_Fe and LB3+2H_54Fe are generated instead of LB3+2H when precursor ion adduct "FeH-2" is used). See also Precursor Ion Adduct. A mass spectrometry nomenclature for linear polyketide siderophores is shown in the following figure.

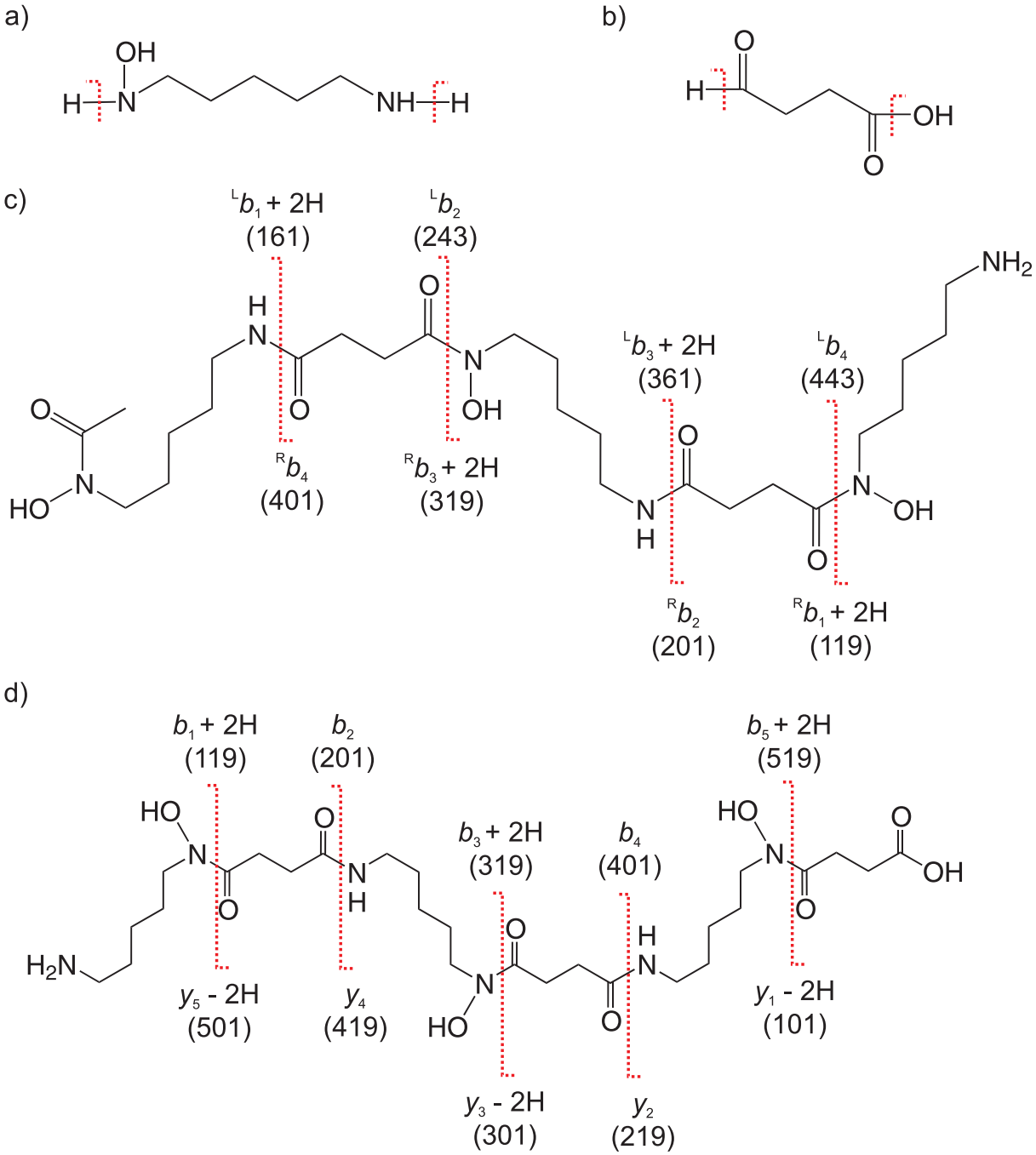


Figure 2.29 Mass spectrometry nomenclature for linear polyketide siderophores; ketide building blocks - Hpd (N-Hydroxy-1,5-pentanediamine; a) and Suc (Succinic semialdehyde; b); fragment ion series of desferrioxamine B (c), and desferrioxamine G1 (d).

2.8.4 Linear Polyketide Sequence Detection

The detection of linear polyketide sequence candidates is similar to the detection of linear peptide sequence candidates. See Linear Sequence Detection.

A difference is that four starting points are applicable to linear polyketides: H^+, H^+ + 2H, H_3O^+ and H_3O^+ - 2H. If a ferri-form of a siderophore is being detected, masses of all starting points must be increased by mass of Fe - 3H. The process is analogous for other metals.

2.9 Cyclic Siderophores

2.9.1 Cyclic Nonribosomal Peptide Siderophores

There are no special requirements for identification of desferri-forms of cyclic siderophores which are nonribosomal peptides (see Cyclic Peptides). The precursor ion adduct must be set up for identification of their ferri-forms (see Linear Nonribosomal Peptide Siderophores).

2.9.2 Cyclic Polyketide Detail Window

The identification of cyclic polyketides requires an additional annotation of building blocks. See Format of Ketide Building Blocks. The window is opened by a double-click on a row in the output report if a cyclic polyketide was identified. The Enter key can also be used if the row is selected. See also Cyclic Peptide Detail Window and Toolbar for Cyclic Peptides.

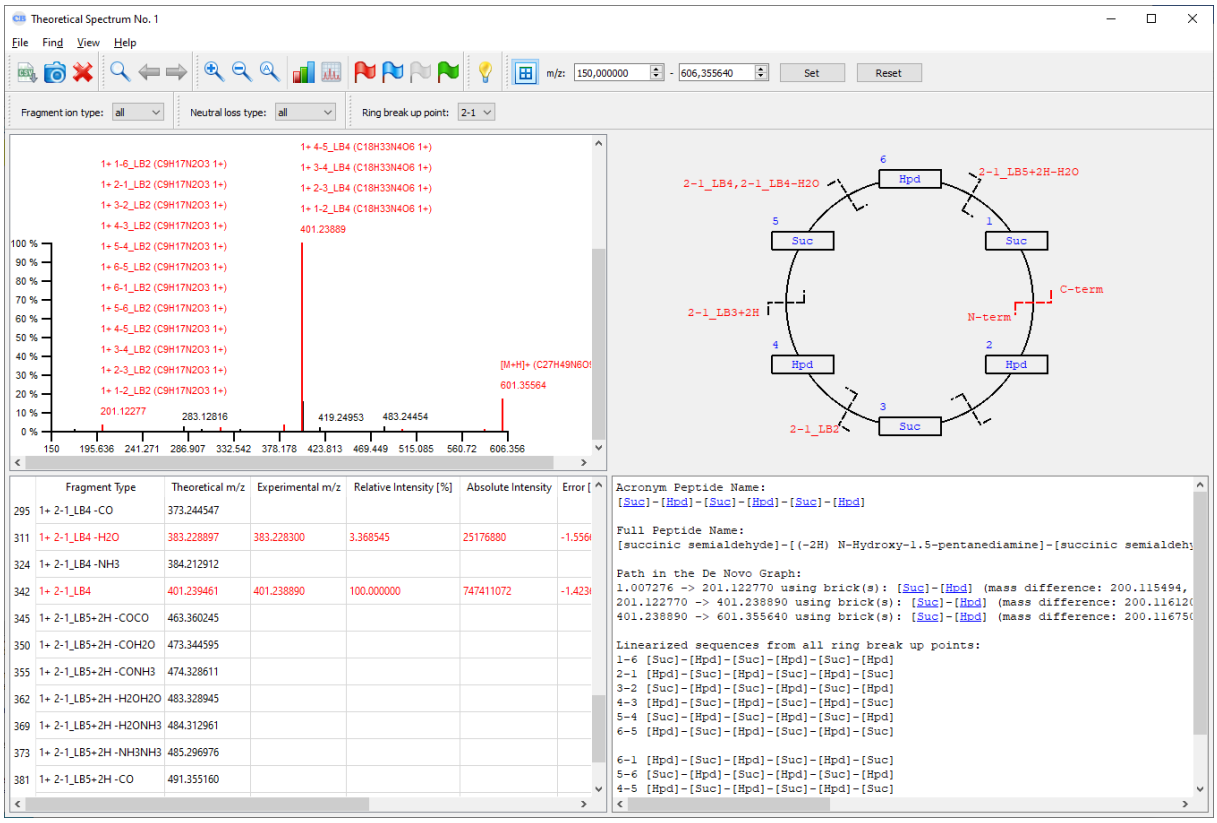


Figure 2.30 De novo sequencing of desferri-ferrioxamine E.

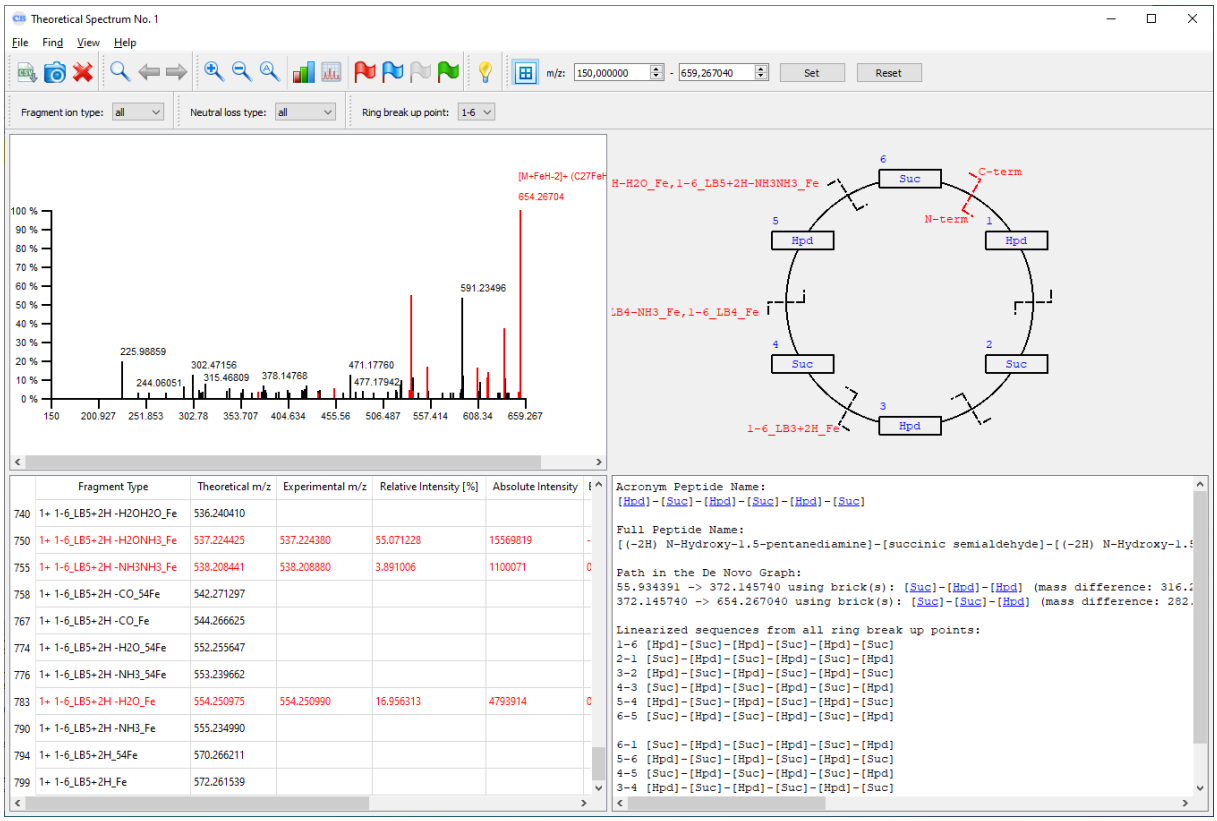


Figure 2.31 De novo sequencing of ferri-ferrioxamine E.

2.9.3 Nomenclature of Cyclic Polyketide Series

The nomenclature of cyclic polyketides is based on the nomenclatures of linear polyketides and cyclic peptides. See Nomenclature of Linear Polyketide Series and Nomenclature of Cyclic Series. CycloBranch generates 2 basic series of fragment ions for cyclic polyketides (L and R stand for left and right, respectively):

- ^{L}b (left b-ion)
- ^{L}b + 2H (left b-ion + 2H)

^{R}b and ^{R}b + 2H series series are not generated because the fragment ions are redundant with the series ^{L}b and ^{L}b + 2H. The fragment ion series ^{L}y, ^{L}y - 2H, ^{R}y and ^{R}y - 2H (utilized in Nomenclature of Linear Polyketide Series) are not applicable to cyclic polyketide siderophores due to head-to-tail cyclization.
Like in the case of cyclic peptides, CycloBranch generates theoretical series of ions $^{i-j,L}$b and $^{i-j,L}$b + 2H where i-j are numbers of building blocks between which a ring having n building blocks is opened (i,j <= n; i=1, j=n; or i=j+1, otherwise). CycloBranch also generates reversed series of ions $^{j-i,L}$b and $^{j-i,L}$b + 2H.
When a metal is included in the precursor ion adduct field, the generated theoretical fragments are suffixed with its name and corresponding theoretical isotope fragments are generated (e.g., the fragments LB3+2H_↩ Fe and LB3+2H_54Fe are generated instead of LB3+2H when precursor ion adduct "FeH-2" is used). See also Precursor Ion Adduct. A mass spectrometry nomenclature for cyclic polyketide siderophore desferrioxamine E is shown in the following figure.

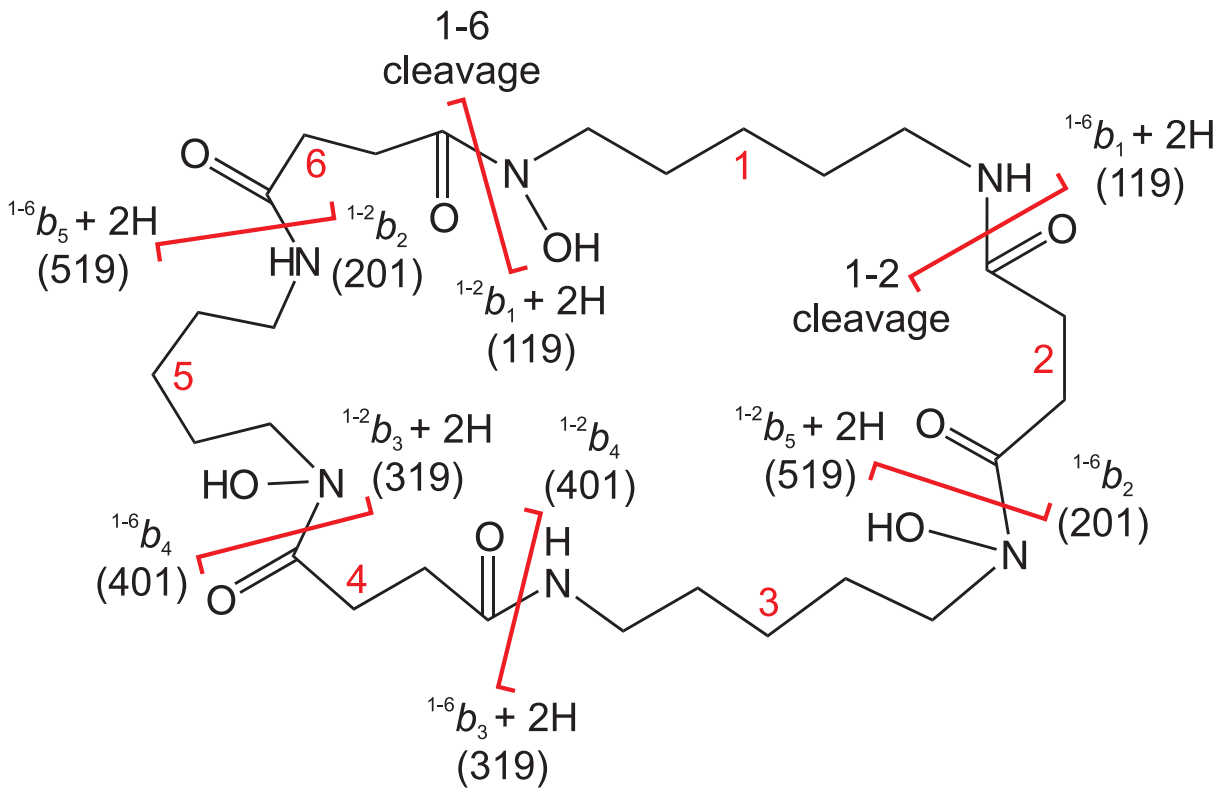


Figure 2.32 Mass spectrometry nomenclature for cyclic polyketide siderophore desferrioxamine E.

2.9.4 Cyclic Polyketide Sequence Detection

The detection of cyclic polyketide sequence candidates is based on Linear Polyketide Sequence Detection and Cyclic Sequence Detection. The starting points applicable to cyclic polyketides are H^+ and H^+ + 2H.

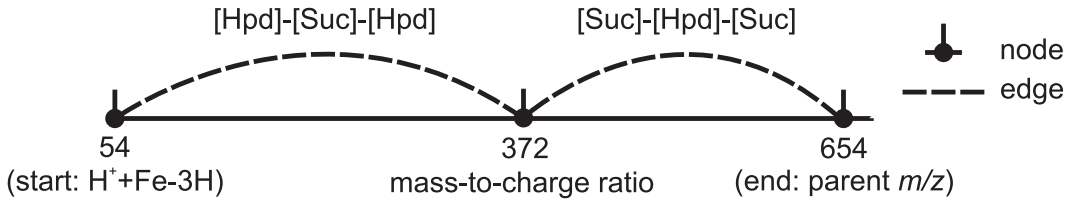


Figure 2.33 De novo graph created from the experimental spectrum of ferrioxamine E (Hpd = N-Hydroxy-1,5-pentanediamine; Suc = Succinic semialdehyde).

2.10 Dereplication

2.10.1 Dereplication in MS Data

CycloBranch supports the dereplication of compounds in single mass spectra (MALDI-MS). The mode "Compare Peaklist(s) with Database - MS, LC-MS, MSI" must be selected (see Mode). The detectable ion types are listed in Ion Types in Theoretical Spectra. An output report example is shown in the following picture.

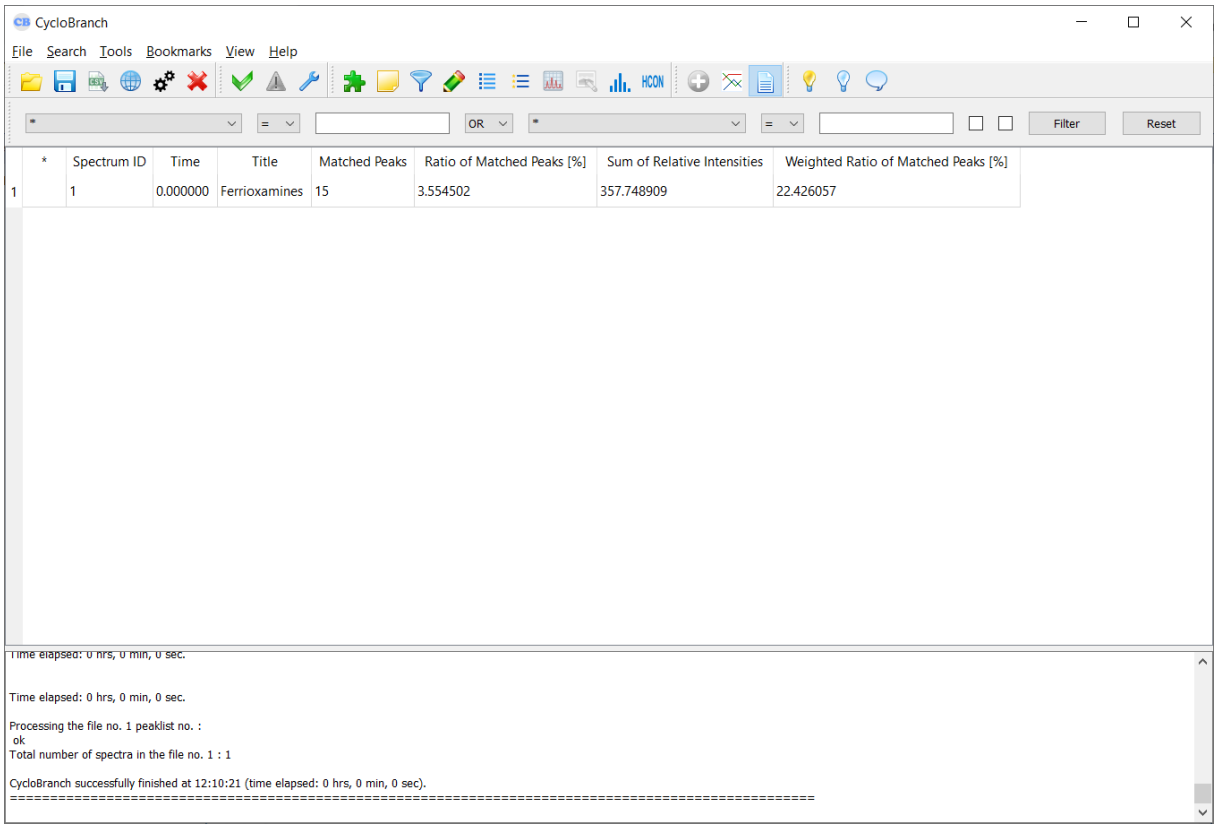


Figure 2.34 Single mass spectrum was searched against an in-house database of siderophores - output report.

The detail window is opened by a double-click on a row in the output report if a linear peptide was identified. The Enter key can also be used if the row is selected. See also Linear Peptide Detail Window and Toolbar for Linear Peptides. The differences for MS data are following:

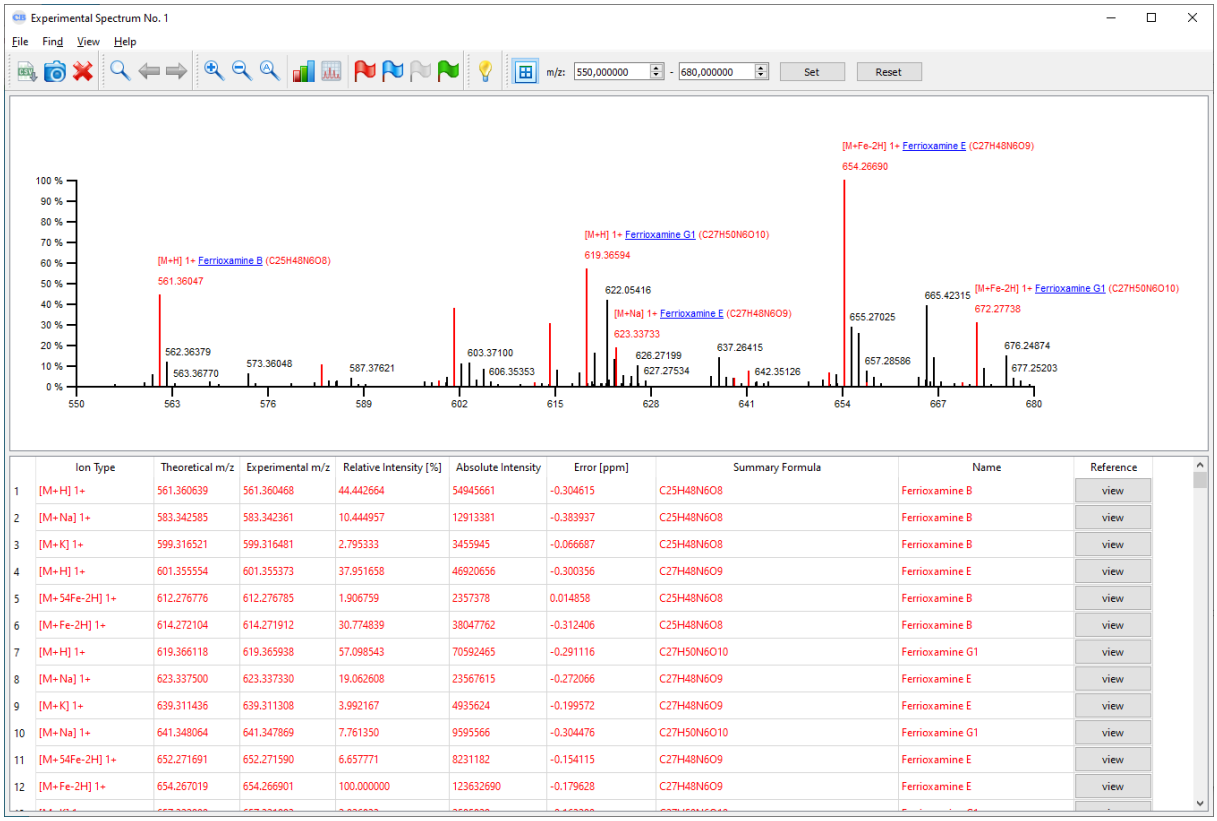
- Only the windows on the left side are available.
- The red peaks in the graphical visualization of a mass spectrum were assigned to compounds in a database.

The following details are listed in the table of peaks:

- Ion Type - a name of a theoretical peak including information about its charge (if Generate Full Isotope Patterns is disabled)
- Pattern Type - a name of a theoretical isotope pattern including information about its charge (if Generate Full Isotope Patterns is enabled)
- Theoretical m/z - a theoretical mass-to-charge ratio of the peak
- Theoretical intensity [%] - theoretical intensity of the peak (if Generate Full Isotope Patterns is enabled)
- Experimental m/z - a mass-to-charge ratio of a matched peak
- Relative Intensity [%] - a relative intensity of a matched peak
- Absolute Intensity - an absolute intensity of a matched peak
- Error [ppm] - a ppm error
- Score - the angle distance calculated between theoretical and experimental isotopic patterns (if Generate Full Isotope Patterns and Calculate FDRs are enabled)
- FDR - false discovery rate of a matched isotopic pattern (if Generate Full Isotope Patterns and Calculate FDRs are enabled)

- Summary Formula - a summary formula of a neutral molecule
- Name - a name of a compound
- Reference - a link into a database if available (ChemSpider, PubChem or Norine)

Note that theoretical peaks which were not assigned to experimental peaks are not reported in the table.

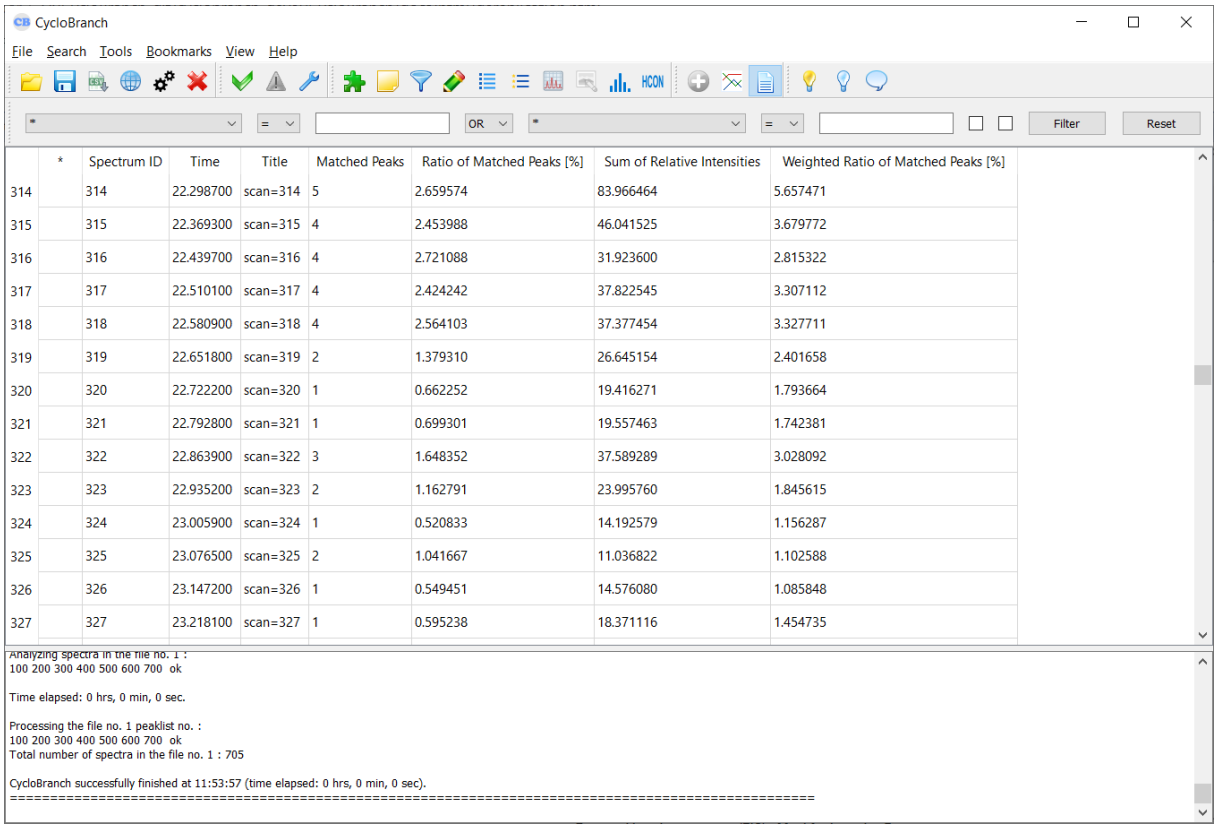


	Ion Type	Theoretical m/z	Experimental m/z	Relative Intensity [%]	Absolute Intensity	Error [ppm]	Summary Formula	Name	Reference
1	[M+H] 1+	561.360639	561.360468	44.442664	54945661	-0.304615	C25H48N6O8	Ferrioxamine B	view
2	[M+Na] 1+	583.342585	583.342361	10.444957	12913381	-0.383937	C25H48N6O8	Ferrioxamine B	view
3	[M+K] 1+	599.316521	599.316481	2.795333	3455945	-0.066687	C25H48N6O8	Ferrioxamine B	view
4	[M+H] 1+	601.355554	601.355373	37.951658	46920656	-0.300356	C27H48N6O9	Ferrioxamine E	view
5	[M+54Fe-2H] 1+	612.276776	612.276785	1.906759	2357378	0.014858	C25H48N6O8	Ferrioxamine B	view
6	[M+Fe-2H] 1+	614.272104	614.271912	30.774839	38047762	-0.312406	C25H48N6O8	Ferrioxamine B	view
7	[M+H] 1+	619.366118	619.365938	57.098543	70592465	-0.291116	C27H50N6O10	Ferrioxamine G1	view
8	[M+Na] 1+	623.337500	623.337330	19.062608	23567615	-0.272066	C27H48N6O9	Ferrioxamine E	view
9	[M+K] 1+	639.311436	639.311308	3.992167	4935624	-0.199572	C27H48N6O9	Ferrioxamine E	view
10	[M+Na] 1+	641.348064	641.347869	7.761350	9595566	-0.304476	C27H50N6O10	Ferrioxamine G1	view
11	[M+54Fe-2H] 1+	652.271691	652.271590	6.657771	8231182	-0.154115	C27H48N6O9	Ferrioxamine E	view
12	[M+Fe-2H] 1+	654.267019	654.266901	100.000000	123632690	-0.179628	C27H48N6O9	Ferrioxamine E	view

Figure 2.35 Single mass spectrum was searched against the in-house database of siderophores - detail window.

2.10.2 Dereplication in LC-MS data

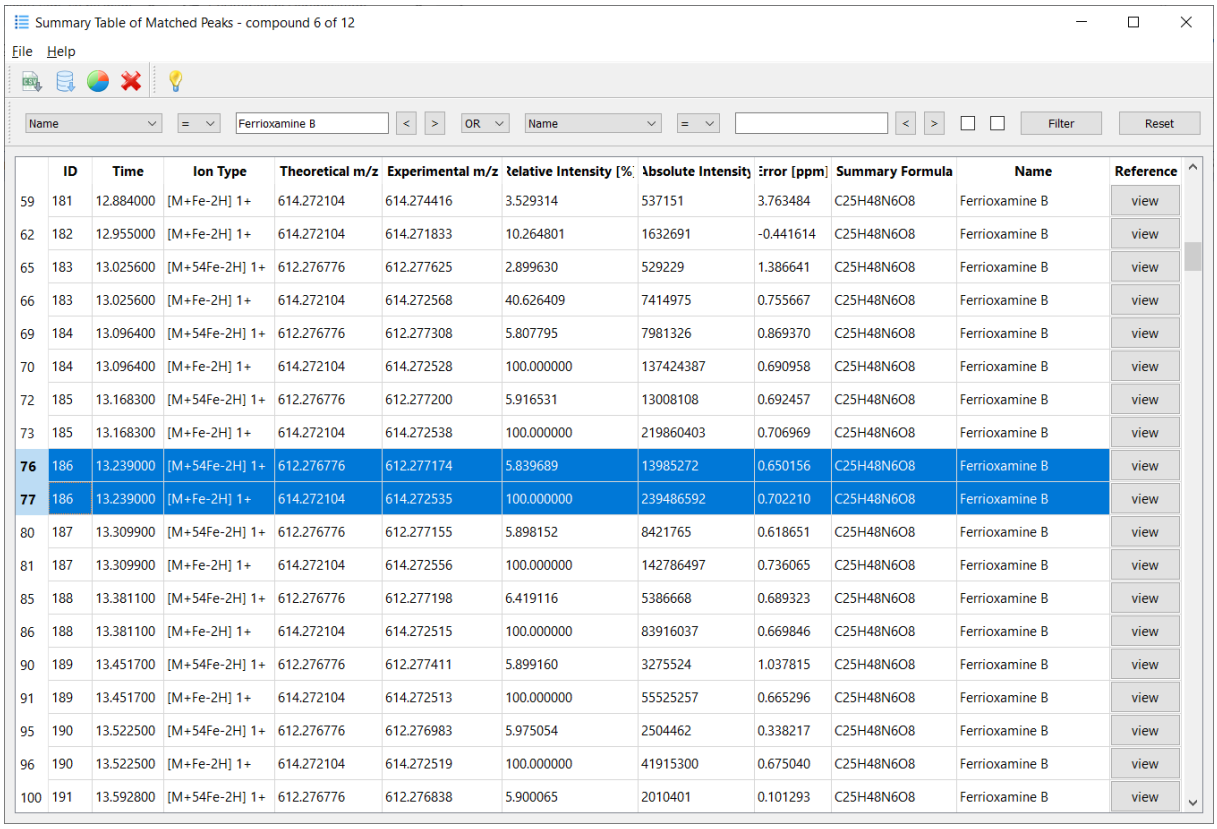
CycloBranch supports also the dereplication of compounds in LC-MS data. The mode "Compare Peaklist(s) with Database - MS, LC-MS, MSI" must be selected (see Mode). The data files containing spectra from multiple LC-MS scans are processed automatically. In this case, each row in the output report corresponds to one scan.



	*	Spectrum ID	Time	Title	Matched Peaks	Ratio of Matched Peaks [%]	Sum of Relative Intensities	Weighted Ratio of Matched Peaks [%]
314		314	22.298700	scan=314	5	2.659574	83.966464	5.657471
315		315	22.369300	scan=315	4	2.453988	46.041525	3.679772
316		316	22.439700	scan=316	4	2.721088	31.923600	2.815322
317		317	22.510100	scan=317	4	2.424242	37.822545	3.307112
318		318	22.580900	scan=318	4	2.564103	37.377454	3.327711
319		319	22.651800	scan=319	2	1.379310	26.645154	2.401658
320		320	22.722200	scan=320	1	0.662252	19.416271	1.793664
321		321	22.792800	scan=321	1	0.699301	19.557463	1.742381
322		322	22.863900	scan=322	3	1.648352	37.589289	3.028092
323		323	22.935200	scan=323	2	1.162791	23.995760	1.845615
324		324	23.005900	scan=324	1	0.520833	14.192579	1.156287
325		325	23.076500	scan=325	2	1.041667	11.036822	1.102588
326		326	23.147200	scan=326	1	0.549451	14.576080	1.085848
327		327	23.218100	scan=327	1	0.595238	18.371116	1.454735

Figure 2.36 Batch processed LC-MS data - output report.

The compounds detected in the scans are summarized in the Summary Table of Matched Peaks. The "ID" in the summary table of matched peaks refers to the "Spectrum ID" in the output report, and it is the scan number. See the Summary Table of Matched Peaks in the following image and the corresponding chromatogram (see also Chromatogram).



	ID	Time	Ion Type	Theoretical m/z	Experimental m/z	Relative Intensity [%	Absolute Intensity	Error [ppm]	Summary Formula	Name	Reference
59	181	12.884000	[M+Fe-2H] 1+	614.272104	614.274416	3.529314	537151	3.763484	C25H48N6O8	Ferrioxamine B	view
62	182	12.955000	[M+Fe-2H] 1+	614.272104	614.271833	10.264801	1632691	-0.441614	C25H48N6O8	Ferrioxamine B	view
65	183	13.025600	[M+54Fe-2H] 1+	612.276776	612.277625	2.899630	529229	1.386641	C25H48N6O8	Ferrioxamine B	view
66	183	13.025600	[M+Fe-2H] 1+	614.272104	614.272568	40.626409	7414975	0.755667	C25H48N6O8	Ferrioxamine B	view
69	184	13.096400	[M+54Fe-2H] 1+	612.276776	612.277308	5.807795	7981326	0.869370	C25H48N6O8	Ferrioxamine B	view
70	184	13.096400	[M+Fe-2H] 1+	614.272104	614.272528	100.000000	137424387	0.690958	C25H48N6O8	Ferrioxamine B	view
72	185	13.168300	[M+54Fe-2H] 1+	612.276776	612.277200	5.916531	13008108	0.692457	C25H48N6O8	Ferrioxamine B	view
73	185	13.168300	[M+Fe-2H] 1+	614.272104	614.272538	100.000000	219860403	0.706969	C25H48N6O8	Ferrioxamine B	view
76	186	13.239000	[M+54Fe-2H] 1+	612.276776	612.277174	5.839689	13985272	0.650156	C25H48N6O8	Ferrioxamine B	view
77	186	13.239000	[M+Fe-2H] 1+	614.272104	614.272535	100.000000	239486592	0.702210	C25H48N6O8	Ferrioxamine B	view
80	187	13.309900	[M+54Fe-2H] 1+	612.276776	612.277155	5.898152	8421765	0.618651	C25H48N6O8	Ferrioxamine B	view
81	187	13.309900	[M+Fe-2H] 1+	614.272104	614.272556	100.000000	142786497	0.736065	C25H48N6O8	Ferrioxamine B	view
85	188	13.381100	[M+54Fe-2H] 1+	612.276776	612.277198	6.419116	5386668	0.689323	C25H48N6O8	Ferrioxamine B	view
86	188	13.381100	[M+Fe-2H] 1+	614.272104	614.272515	100.000000	83916037	0.669846	C25H48N6O8	Ferrioxamine B	view
90	189	13.451700	[M+54Fe-2H] 1+	612.276776	612.277411	5.899160	3275524	1.037815	C25H48N6O8	Ferrioxamine B	view
91	189	13.451700	[M+Fe-2H] 1+	614.272104	614.272513	100.000000	55525257	0.665296	C25H48N6O8	Ferrioxamine B	view
95	190	13.522500	[M+54Fe-2H] 1+	612.276776	612.276983	5.975054	2504462	0.338217	C25H48N6O8	Ferrioxamine B	view
96	190	13.522500	[M+Fe-2H] 1+	614.272104	614.272519	100.000000	41915300	0.675040	C25H48N6O8	Ferrioxamine B	view
100	191	13.592800	[M+54Fe-2H] 1+	612.276776	612.276838	5.900065	2010401	0.101293	C25H48N6O8	Ferrioxamine B	view

Figure 2.37 List of peaks assigned to ferri-ferrioxamine B in the batch-processed LC-MS run.

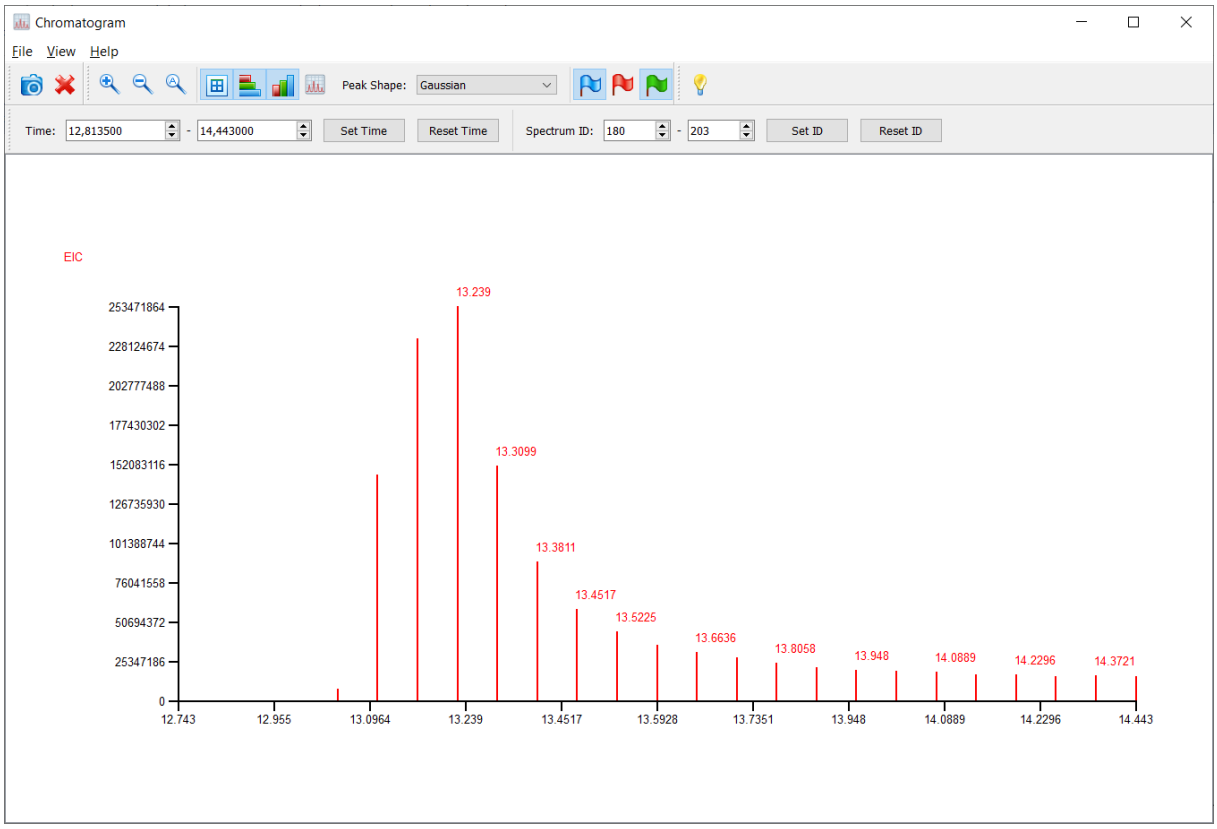
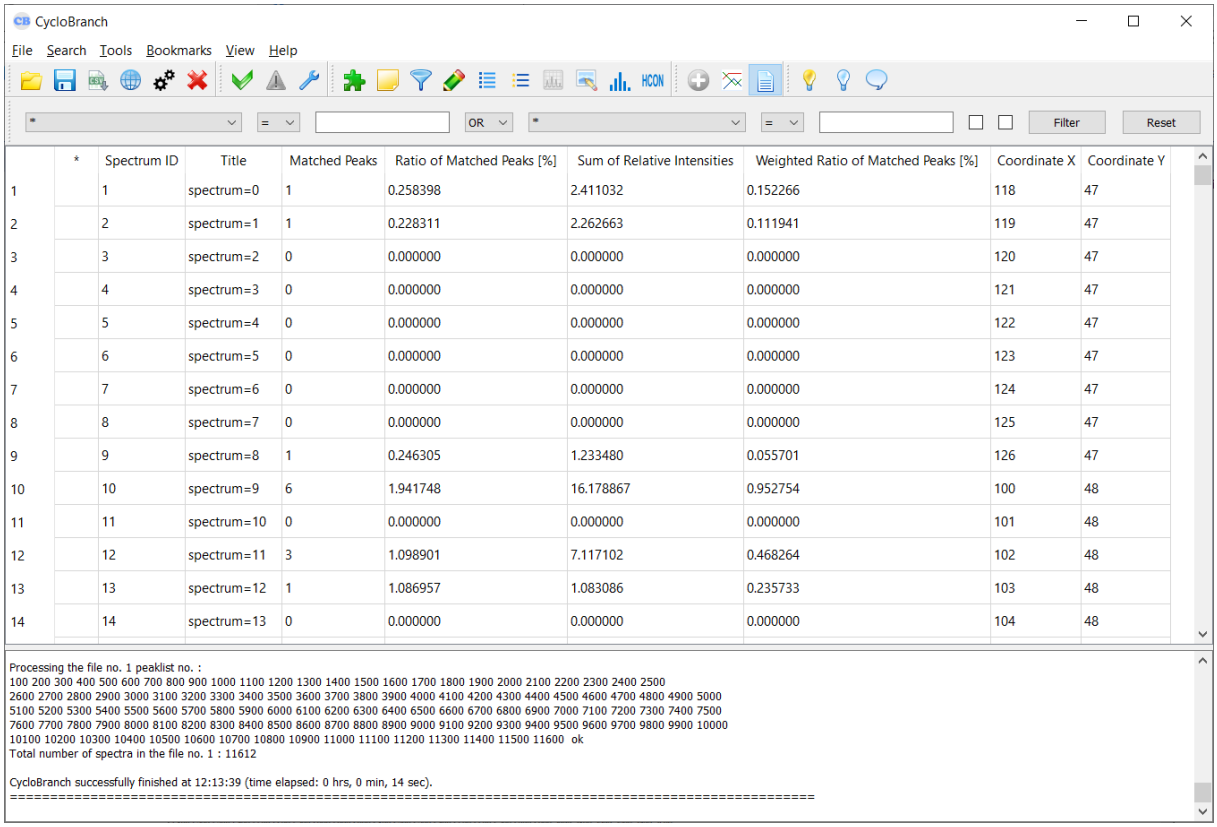


Figure 2.38 Extracted ion chromatogram (EIC) of ferri-ferrioxamine B.

2.10.3 Dereplication in MSI Data

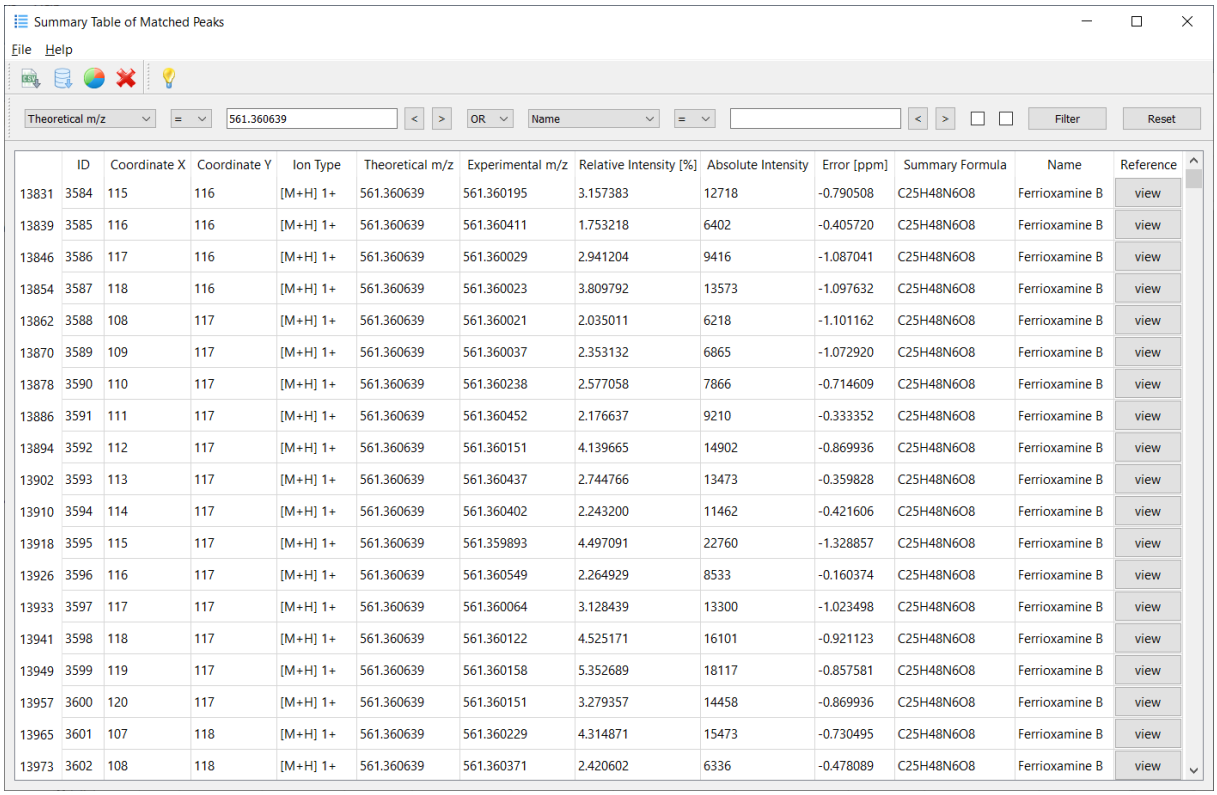
CycloBranch supports the dereplication of compounds in imaging mass spectra (MSI). The mode "Compare Peaklist(s) with Database - MS, LC-MS, MSI" must be selected (see Mode). An imzML file must be used as the input file in the settings dialog. A row in the output report corresponds to single mass spectrum with the coordinates [X,Y].



	*	Spectrum ID	Title	Matched Peaks	Ratio of Matched Peaks [%]	Sum of Relative Intensities	Weighted Ratio of Matched Peaks [%]	Coordinate X	Coordinate Y
1		1	spectrum=0	1	0.258398	2.411032	0.152266	118	47
2		2	spectrum=1	1	0.228311	2.262663	0.111941	119	47
3		3	spectrum=2	0	0.000000	0.000000	0.000000	120	47
4		4	spectrum=3	0	0.000000	0.000000	0.000000	121	47
5		5	spectrum=4	0	0.000000	0.000000	0.000000	122	47
6		6	spectrum=5	0	0.000000	0.000000	0.000000	123	47
7		7	spectrum=6	0	0.000000	0.000000	0.000000	124	47
8		8	spectrum=7	0	0.000000	0.000000	0.000000	125	47
9		9	spectrum=8	1	0.246305	1.233480	0.055701	126	47
10		10	spectrum=9	6	1.941748	16.178867	0.952754	100	48
11		11	spectrum=10	0	0.000000	0.000000	0.000000	101	48
12		12	spectrum=11	3	1.098901	7.117102	0.468264	102	48
13		13	spectrum=12	1	1.086957	1.083086	0.235733	103	48
14		14	spectrum=13	0	0.000000	0.000000	0.000000	104	48

Figure 2.39 A mass spectrometry imaging dataset was searched against the in-house database of siderophores - output report.

If MSI data are analyzed the Summary Table of Matched Peaks includes compounds detected in all single mass spectra. The Image Fusion tool visualizes the localization of compounds in an optical image.



	ID	Coordinate X	Coordinate Y	Ion Type	Theoretical m/z	Experimental m/z	Relative Intensity [%]	Absolute Intensity	Error [ppm]	Summary Formula	Name	Reference
13831	3584	115	116	[M+H] 1+	561.360639	561.360195	3.157383	12718	-0.790508	C25H48N6O8	Ferrioxamine B	view
13839	3585	116	116	[M+H] 1+	561.360639	561.360411	1.753218	6402	-0.405720	C25H48N6O8	Ferrioxamine B	view
13846	3586	117	116	[M+H] 1+	561.360639	561.360029	2.941204	9416	-1.087041	C25H48N6O8	Ferrioxamine B	view
13854	3587	118	116	[M+H] 1+	561.360639	561.360023	3.809792	13573	-1.097632	C25H48N6O8	Ferrioxamine B	view
13862	3588	108	117	[M+H] 1+	561.360639	561.360021	2.035011	6218	-1.101162	C25H48N6O8	Ferrioxamine B	view
13870	3589	109	117	[M+H] 1+	561.360639	561.360037	2.353132	6865	-1.072920	C25H48N6O8	Ferrioxamine B	view
13878	3590	110	117	[M+H] 1+	561.360639	561.360238	2.577058	7866	-0.714609	C25H48N6O8	Ferrioxamine B	view
13886	3591	111	117	[M+H] 1+	561.360639	561.360452	2.176637	9210	-0.333352	C25H48N6O8	Ferrioxamine B	view
13894	3592	112	117	[M+H] 1+	561.360639	561.360151	4.139665	14902	-0.869936	C25H48N6O8	Ferrioxamine B	view
13902	3593	113	117	[M+H] 1+	561.360639	561.360437	2.744766	13473	-0.359828	C25H48N6O8	Ferrioxamine B	view
13910	3594	114	117	[M+H] 1+	561.360639	561.360402	2.243200	11462	-0.421606	C25H48N6O8	Ferrioxamine B	view
13918	3595	115	117	[M+H] 1+	561.360639	561.359893	4.497091	22760	-1.328857	C25H48N6O8	Ferrioxamine B	view
13926	3596	116	117	[M+H] 1+	561.360639	561.360549	2.264929	8533	-0.160374	C25H48N6O8	Ferrioxamine B	view
13933	3597	117	117	[M+H] 1+	561.360639	561.360064	3.128439	13300	-1.023498	C25H48N6O8	Ferrioxamine B	view
13941	3598	118	117	[M+H] 1+	561.360639	561.360122	4.525171	16101	-0.921123	C25H48N6O8	Ferrioxamine B	view
13949	3599	119	117	[M+H] 1+	561.360639	561.360158	5.352689	18117	-0.857581	C25H48N6O8	Ferrioxamine B	view
13957	3600	120	117	[M+H] 1+	561.360639	561.360151	3.279357	14458	-0.869936	C25H48N6O8	Ferrioxamine B	view
13965	3601	107	118	[M+H] 1+	561.360639	561.360229	4.314871	15473	-0.730495	C25H48N6O8	Ferrioxamine B	view
13973	3602	108	118	[M+H] 1+	561.360639	561.360371	2.420602	6336	-0.478089	C25H48N6O8	Ferrioxamine B	view

Figure 2.40 Summary table of matched peaks - desferri-ferrioxamine B was filtered by its theoretical m/z ratio.

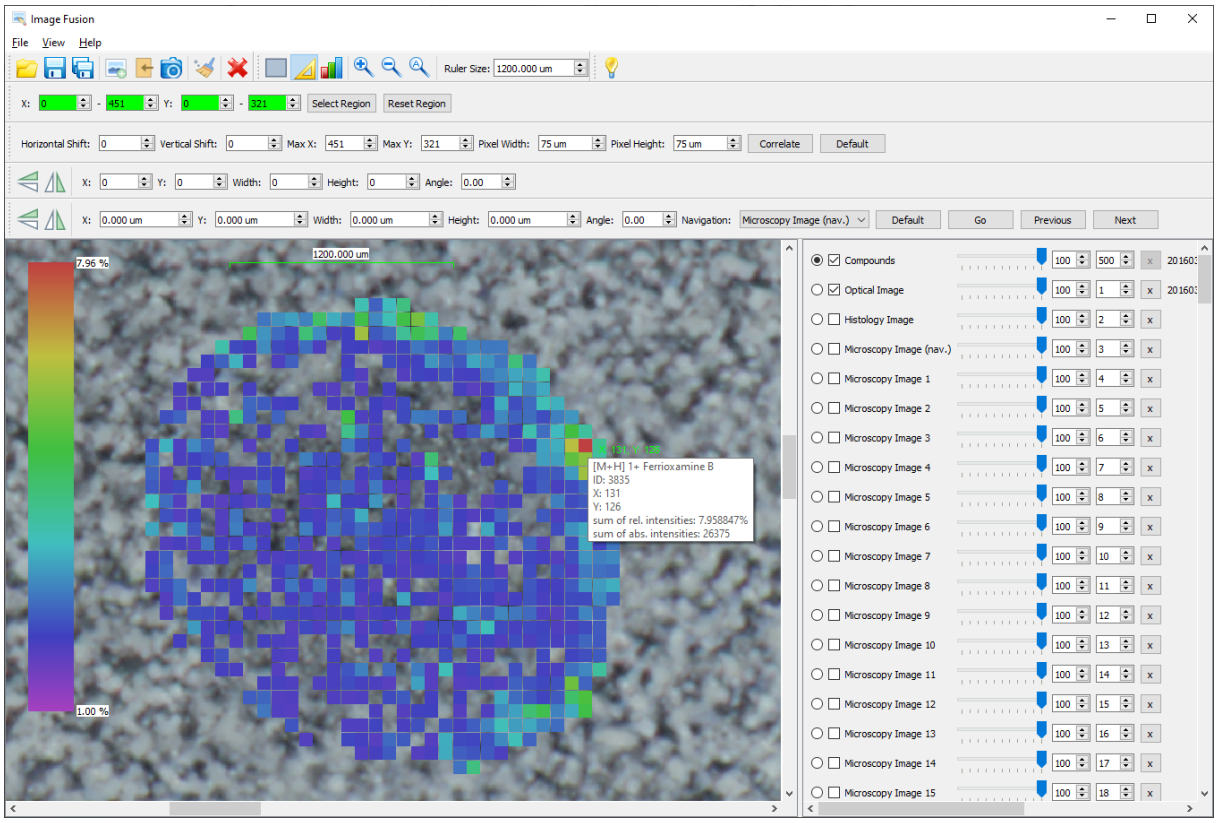


Figure 2.41 Image Fusion - detail of a region containing desferri-ferrioxamine B.

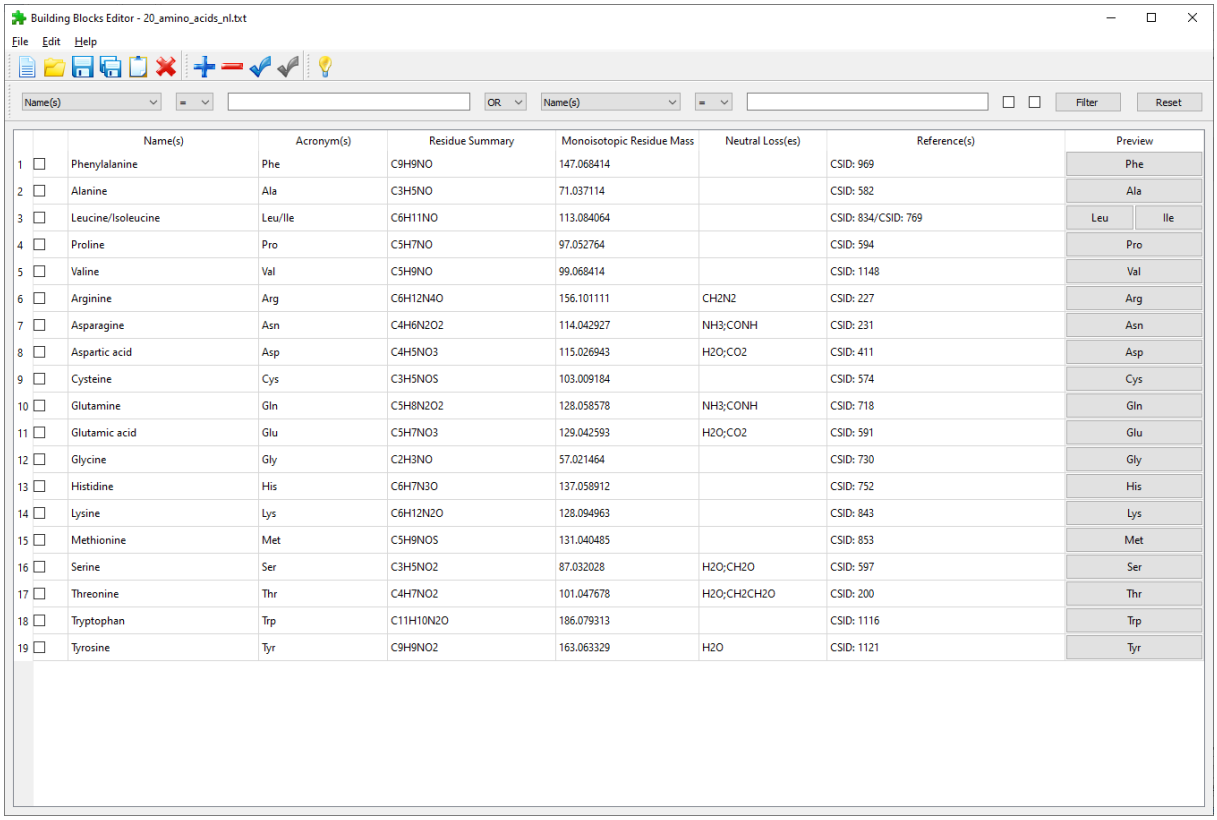
2.10.4 Databases of Compounds

The databases of sequences/compounds can be created/edited using Sequence/Compound Database Editor. The same database can be used in both MS and MS/MS mode. However, the column "Sequence" must be defined in MS/MS mode. See List of Available Sequence/Compound Databases.

2.11 Building Blocks Editor

2.11.1 Building Blocks Editor

The editor is opened using the command "Tools -> Building Blocks Editor" from the Main Window. Use F1 to open this help.
Changes in the editor do not impact the settings of the engine until they are saved !



		Name(s)	Acronym(s)	Residue Summary	Monoisotopic Residue Mass	Neutral Loss(es)	Reference(s)	Preview
1	☐	Phenylalanine	Phe	C9H9NO	147.068414		CSID: 969	Phe
2	☐	Alanine	Ala	C3H5NO	71.037114		CSID: 582	Ala
3	☐	Leucine/Isoleucine	Leu/Ile	C6H11NO	113.084064		CSID: 834/CSID: 769	Leu Ile
4	☐	Proline	Pro	C5H7NO	97.052764		CSID: 594	Pro
5	☐	Valine	Val	C5H9NO	99.068414		CSID: 1148	Val
6	☐	Arginine	Arg	C6H12N4O	156.101111	CH2N2	CSID: 227	Arg
7	☐	Asparagine	Asn	C4H6N2O2	114.042927	NH3;CONH	CSID: 231	Asn
8	☐	Aspartic acid	Asp	C4H5NO3	115.026943	H2O;CO2	CSID: 411	Asp
9	☐	Cysteine	Cys	C3H5NOS	103.009184		CSID: 574	Cys
10	☐	Glutamine	Gln	C5H8N2O2	128.058578	NH3;CONH	CSID: 718	Gln
11	☐	Glutamic acid	Glu	C5H7NO3	129.042593	H2O;CO2	CSID: 591	Glu
12	☐	Glycine	Gly	C2H3NO	57.021464		CSID: 730	Gly
13	☐	Histidine	His	C6H7N3O	137.058912		CSID: 752	His
14	☐	Lysine	Lys	C6H12N2O	128.094963		CSID: 843	Lys
15	☐	Methionine	Met	C5H9NOS	131.040485		CSID: 853	Met
16	☐	Serine	Ser	C3H5NO2	87.032028	H2O;CH2O	CSID: 597	Ser
17	☐	Threonine	Thr	C4H7NO2	101.047678	H2O;CH2CH2O	CSID: 200	Thr
18	☐	Tryptophan	Trp	C11H10N2O	186.079313		CSID: 1116	Trp
19	☐	Tyrosine	Tyr	C9H9NO2	163.063329	H2O	CSID: 1121	Tyr

Figure 2.42 Building Blocks Editor

2.11.1.1 Toolbar

New Database (CTRL + N) 
Create a new database of building blocks.

Open Database (CTRL + O) 
Open a database of building blocks.

Save Database (CTRL + S) 
Save the database of building blocks.

Save Database As... (CTRL + D) 
Save the database of building blocks with another file name.

Import Database (CTRL + I) 
Import a database of building blocks.

Close (Esc) 
Close window.

Insert Row (Insert) 
Insert a new row at the bottom of the table.

Remove Selected Rows (Delete) 
Remove checked rows.

Select All (CTRL + A) 
Select all rows.

Unselect Rows (CTRL + U) 
Unselect all rows.

HTML Documentation (F1) 
Show HTML documentation.

2.11.1.2 Text Filter

See details here.

2.11.1.3 Columns

- Name(s) - Name of a building block (or more blocks corresponding to isomers separated by '/').
- Acronym(s) - Acronym of a building block (or more blocks corresponding to isomers separated by '/').
- Residue Summary - Residue summary formula of a building block.
- Monoisotopic Residue Mass - Monoisotopic mass corresponding to a block. The value is recalculated when the summary formula is changed.
- Neutral Loss(es) - List of molecular formulas of neutral losses separated by semicolon (one list is used for all isomers i.e. the symbol '/' cannot be used in this column).
- Reference(s) - Reference id of a building block (or more blocks corresponding to isomers separated by '/'). See Format of Building Blocks Databases.
- Preview - A link to a reference database (e.g., ChemSpider) generated from the fields acronym and reference. The link is generated automatically and cannot be edited manually.

The numbers of '/' must be equal in a row in the fields Name, Acronym and Reference. The names and acronyms must not contain the characters '[' or ']'.

2.11.2 Format of Building Block Databases

The database of building blocks is a text file (∗.txt) containing a list of building blocks (one building block per line). Each line contains the following items separated by tabulars:

- name of a block (or more blocks corresponding to isomers separated by '/')
- acronym of a block (or more blocks corresponding to isomers separated by '/')
- residue summary formula of a block (important: an exact mass of the block is calculated from this value)

- mass corresponding to the block (the value is ignored; the mass is automatically calculated from the residue summary formula)
- list of neutral losses separated by semicolon (one list is used for all isomers i.e. the symbol '/' cannot be used in this column)
- reference id (or more blocks corresponding to isomers separated by '/')

The numbers of '/' must be equal on a line in the fields name, acronym and reference id. The names and acronyms must not contain the characters '[' or ']'. A line starting with the hash '#' is a comment which is not processed. Reference ids can be written in the following formats (without quotes):

- "CSID: <number>" (ChemSpider ID; example: "CSID: 13539649")
- "CID: <number>" (PubChem ID; example: "CID: 12439044")
- "CHEBI: <number>" (ChEBI ID; example: "CHEBI: 4031")
- "PDB: <pdb_code>" (PDB code; example: "PDB: UAL")
- "<smiles> in CSID: <number>" (smiles in ChemSpider ID; example: "C(C(CN)NC(C)=O)(=O)O in CSID: 8570696")
- "<smiles> in CID: <number>" (smiles in PubChem ID; example: "C1=C(C(=CC2=C1N3C(C(=C2)N)NC(↩ CC3)C(O)=O)O)O in CID: 5289234")
- "<smiles> in: <norine_id>" (smiles in Norine; example: "C1=C(C(=CC2=C1N3C(C(N)C2)NCCC3C(O)=↩ O)O)O in: NOR00903")

2.11.3 Format of Ketide Building Blocks

The identification of linear and cyclic polyketides requires an additional annotation of building blocks. A name of a building block eliminating 2 hydrogens instead of the water molecule must start with the string "(-2H) " without quotes. Corresponding residue summary formulas must be entered in the column "Residue Summary" (i.e., 2 hydrogens or the water molecule must be deducted manually).

2.11.4 List of Available Building Block Databases

The following pre-defined databases of building blocks are available in "BrickDatabases" folder:

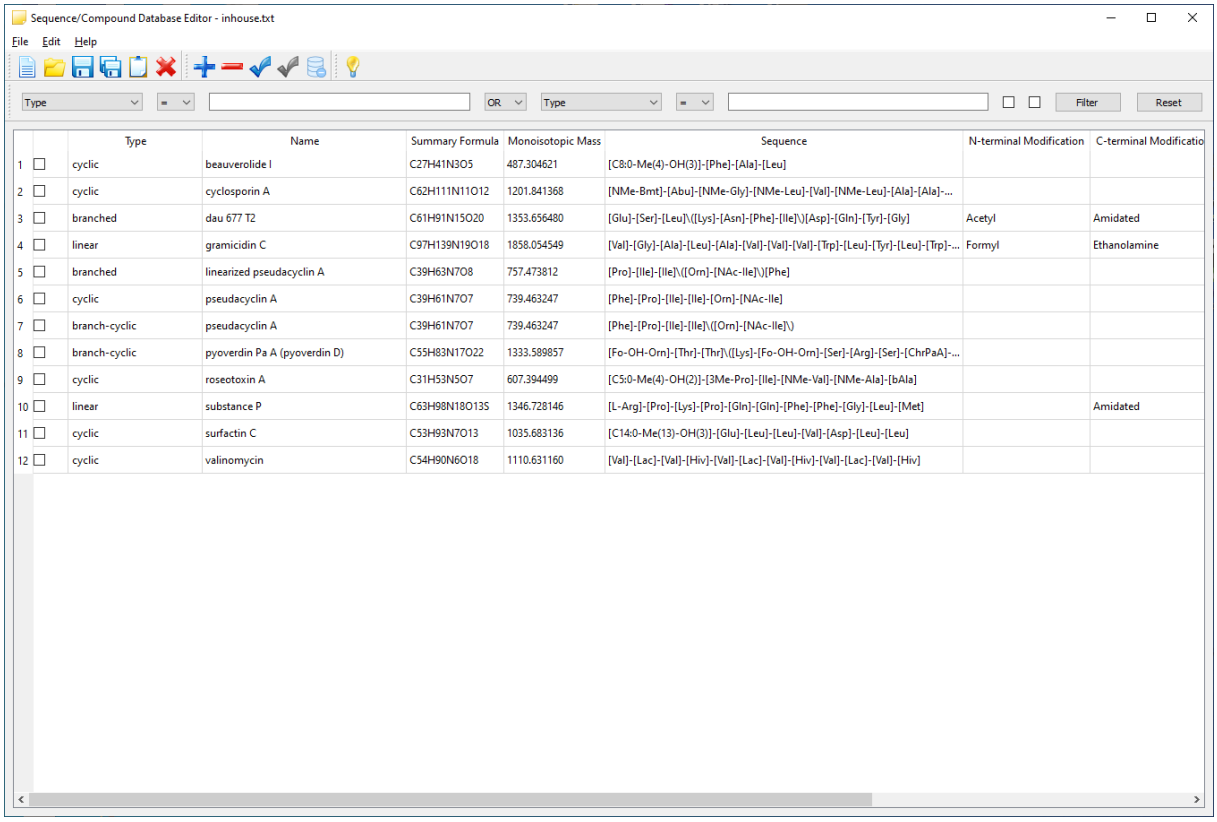
Database	Content	Location
D_{19}	19 proteinogenic amino acids (20 including isobars)	20_amino_acids.txt
D_{19} (nl)	19 proteinogenic amino acids (20 including isobars) with neutral losses	20_amino_acids_nl.txt
D_{33} (v1)	33 NRP building blocks (42 including isobars)	db33_v1.txt
D_{33} (v2)	33 NRP building blocks (132 including isobars)	db33_v2.txt
D_{287} (v1)	287 NRP building blocks (412 including isobars)	db287_v1.txt
D_{287} (v2)	287 NRP building blocks (521 including isobars)	db287_v2.txt
D_{287} (v3)	287 NRP building blocks (523 including isobars)	db287_v3.txt
D_{291}	291 NRP building blocks (525 including isobars)	db291.txt
ketide blocks	19 ketide building blocks (22 including isobars)	inhouse_siderophores_blocks.txt
custom		Download Template for Excel

2.12 Sequence/Compound Database Editor

2.12.1 Sequence/Compound Database Editor

The editor is opened using the command "Tools -> Sequence/Compound Database Editor" from the Main Window. Use F1 to open this help.

Changes in the editor do not impact the settings of the engine until they are saved !



	Type	Name	Summary Formula	Monoisotopic Mass	Sequence	N-terminal Modification	C-terminal Modificatio
1	cyclic	beauverolide I	C27H41N3O5	487.304621	[C8:0-Me(4)-OH(3)]-[Phe]-[Ala]-[Leu]		
2	cyclic	cyclosporin A	C62H111N11O12	1201.841368	[NMe-Bmt]-[Abu]-[NMe-Gly]-[NMe-Leu]-[Val]-[NMe-Leu]-[Ala]-[Ala]-...		
3	branched	dau 677 T2	C61H91N15O20	1353.656480	[Glu]-[Ser]-[Leu]\([Lys]-[Asn]-[Phe]-[Ile]\)[Asp]-[Gln]-[Tyr]-[Gly]	Acetyl	Amidated
4	linear	gramicidin C	C97H139N19O18	1858.054549	[Val]-[Gly]-[Ala]-[Leu]-[Ala]-[Val]-[Val]-[Val]-[Trp]-[Leu]-[Tyr]-[Leu]-[Trp]-...	Formyl	Ethanolamine
5	branched	linearized pseudacyclin A	C39H63N7O8	757.473812	[Pro]-[Ile]-[Ile]\([Orn]-[NAc-Ile]\)[Phe]		
6	cyclic	pseudacyclin A	C39H61N7O7	739.463247	[Phe]-[Pro]-[Ile]-[Ile]-[Orn]-[NAc-Ile]		
7	branch-cyclic	pseudacyclin A	C39H61N7O7	739.463247	[Phe]-[Pro]-[Ile]-[Ile]\([Orn]-[NAc-Ile]\)		
8	branch-cyclic	pyoverdin Pa A (pyoverdin D)	C55H83N17O22	1333.589857	[Fo-OH-Orn]-[Thr]-[Thr]\([Lys]-[Fo-OH-Orn]-[Ser]-[Arg]-[Ser]-[ChrPaA]-...		
9	cyclic	roseotoxin A	C31H53N5O7	607.394499	[C5:0-Me(4)-OH(2)]-[3Me-Pro]-[Ile]-[NMe-Val]-[NMe-Ala]-[bAla]		
10	linear	substance P	C63H98N18O13S	1346.728146	[L-Arg]-[Pro]-[Lys]-[Pro]-[Gln]-[Gln]-[Phe]-[Phe]-[Gly]-[Leu]-[Met]		Amidated
11	cyclic	surfactin C	C53H93N7O13	1035.683136	[C14:0-Me(13)-OH(3)]-[Glu]-[Leu]-[Leu]-[Val]-[Asp]-[Leu]-[Leu]		
12	cyclic	valinomycin	C54H90N6O18	1110.631160	[Val]-[Lac]-[Val]-[Hiv]-[Val]-[Lac]-[Val]-[Hiv]-[Val]-[Lac]-[Val]-[Hiv]		

Figure 2.43 Sequence/Compound Database Editor - sample database of sequences for MS/MS data analysis

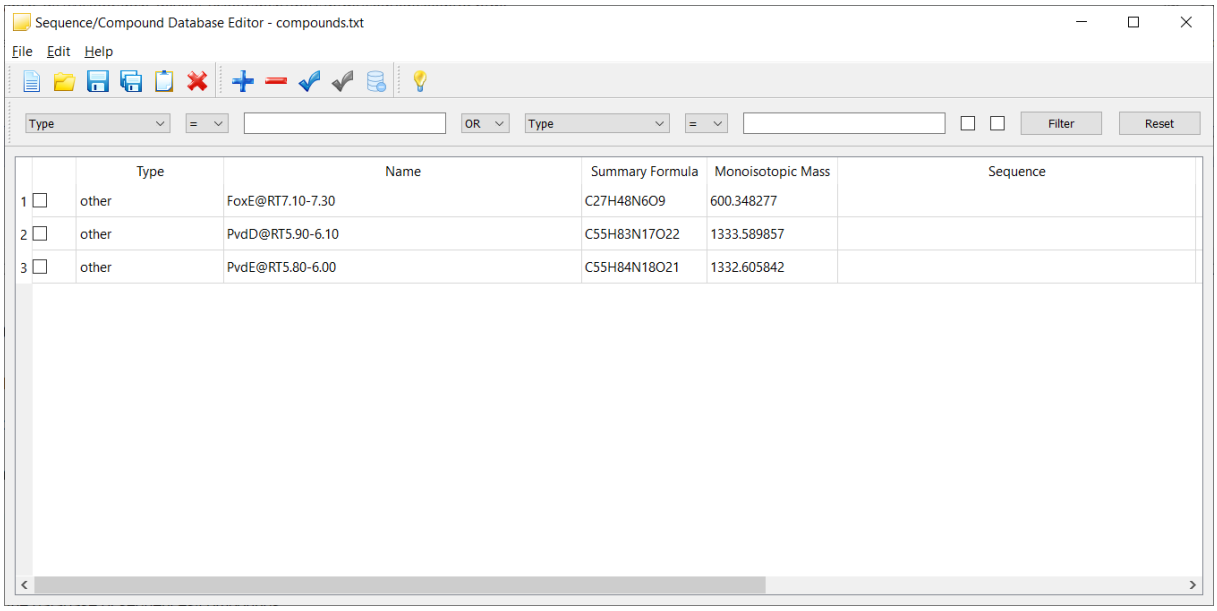


Figure 2.44 Sequence/Compound Database Editor - sample database of compounds for LC-MS data analysis

2.12.1.1 Toolbar

New Database (CTRL + N)
Create a new database of sequences/compounds.

Open Database (CTRL + O)
Open a database of sequences/compounds.

Save Database (CTRL + S)
Save the database of sequences/compounds.

Save Database As... (CTRL + D)
Save the database of sequences/compounds with another file name.

Import Database (CTRL + I)
Import a database of sequences/compounds.

Close (Esc)
Close window.

Insert Row (Insert)
Insert a new row at the bottom of the table.

Remove Selected Rows (Delete)
Remove checked rows.

Select All (CTRL + A)
Select all rows.

Unselect Rows (CTRL + U)
Unselect all rows.

Remove Duplicate Rows (CTRL + L)
Remove duplicate rows (all fields in the rows must be equal; the rows does not have to be sorted).

HTML Documentation (F1) 
Show HTML documentation.

2.12.1.2 Text Filter

See details here.

2.12.1.3 Columns

- Type - The type of a peptide (linear, cyclic, branched, branch-cyclic, linear-polyketide, cyclic-polyketide, and other).
- Name - Name of a peptide or compound (the syntax `FoxE@RT7.10-7.30` means that ferrioxamine E is searched between 7.1 and 7.3 minutes if LC-MS data are processed; time units are dataset dependent).
- Summary Formula - Summary formula of a neutral molecule.
- Monoisotopic Mass - Monoisotopic mass of the peptide. The value is recalculated when the summary formula is changed.
- Sequence - Peptide sequence. If the column is empty, the database can be utilized for identification of compounds in MS spectra. If the column is non-empty, the database can be used in both MS and MS/MS mode.
- N-terminal Modification - A name of an N-terminal modification of a linear or a branched peptide.
- C-terminal Modification - A name of a C-terminal modification of a linear or a branched peptide.
- Branch Modification - A name of an N-terminal or a C-terminal modification of a branch of a branched or a branch-cyclic peptide.
- Reference - Reference id of a peptide. See Format of Sequence Databases.
- View - View the molecule structure in a reference database (ChemSpider, PubChem or Norine).

2.12.2 Format of Sequence Databases

The database of sequences is a text file (*.txt) containing a list of sequences (one sequence per line). Each line contains the following items separated by tabulars:

- type of a sequence (linear, cyclic, branched, branch-cyclic, linear-polyketide or cyclic-polyketide)
- name of a peptide
- summary formula of a neutral molecule
- monoisotopic mass of peptide (the value is ignored; the mass is automatically calculated from the summary formula)
- a peptide sequence
- name of an N-terminal modification
- name of a C-terminal modification
- name of a branch modification
- reference id of a peptide

A line starting with the hash '#' is a comment which is not processed.
The format of a peptide sequence differs according to a peptide type:

- **Linear peptide:** `[aaa/bbb]-[ccc]-[ddd]-[eee]-[fff]`. The names of building blocks must be enclosed in `[` and `]`. The building blocks are separated by `-`. Isomers of a building block are separated by `/`.

- **Cyclic peptide:** the same syntax like in case of a linear peptide.
- **Branched peptide:** `[aaa/bbb]\([ccc]-[ddd]-[eee]\)[fff]`. The syntax is similar to a linear peptide. The string `\(` is used instead of − to determine the start of a branch. The string `\)` is used instead of − to determine the end of a branch. The building block `[ccc]` is the block where the branching occurs. The building block `[eee]` is the terminal block of the branch. See also Branched Peptides.
- **Branch-cyclic peptide:** the same syntax like in case of a branched peptide.
- **Linear polyketide:** the same like in case of a linear peptide (acronyms of building blocks eliminating 2 hydrogens can be included in the sequence).
- **Cyclic polyketide:** the same like in case of a cyclic peptide (acronyms of building blocks eliminating 2 hydrogens can be included in the sequence).

Reference ids can be written in the following formats (without quotes):

- "CSID: <number>" (ChemSpider ID; example: "CSID: 13539649")
- "CID: <number>" (PubChem ID; example: "CID: 12439044")
- "CHEBI: <number>" (ChEBI ID; example: "CHEBI: 27266")
- "DOI: <doi>" (example: "DOI: 10.1016/j.bbapap.2016.12.003")
- "<npatlas_id>" (NP Atlas ID; example: "NPA017720")
- "<coconut_id>" (COCONUT ID; example: "CNP0187739")
- "<norine_id>" (Norine ID; example: "NOR00205")
- "<lipidmaps_id>" (LipidMaps ID; example: "LMFA00000001")

2.12.3 List of Available Sequence/Compound Databases

All sequence databases can be also used as databases of compounds in MS/MSI mode. See Dereplication. The following databases of are available in "SequenceDatabases" folder:

Location	Content	Mode
inhouse.txt	12 linear, cyclic, branched and branch-cyclic NRP sequences from an in-house collection	MS, MSI, MS/MS
norine-07-2013-linear.txt	310 linear NRP sequences exported from Norine (version July, 2013)	MS, MSI, MS/MS
norine-07-2013-cyclic.txt	443 cyclic NRP sequences exported from Norine (version July, 2013)	MS, MSI, MS/MS
norine-07-2013-branched.txt	5 branched NRP sequences exported from Norine (version July, 2013)	MS, MSI, MS/MS
norine-07-2013-branch-cyclic.txt	283 branch-cyclic NRP sequences exported from Norine (version July, 2013)	MS, MSI, MS/MS
norine-07-2013.txt	1041 linear, cyclic, branched and branch-cyclic NRP sequences exported from Norine (version July, 2013)	MS, MSI, MS/MS
siderophores/inhouse_↩ siderophores.txt	18 NRP and polyketide siderophores (compiled in-house)	MS, MSI, MS/MS

Location	Content	Mode
siderophores/709_siderophores↩ _and_secondary_metabolites.txt	709 siderophores and *Aspergillus* secondary metabolites (compiled in-house)	MS, MSI
metabolites/all_metabolites.txt	850 metabolites of *Alternaria*, *Aspergillus*, *Candida albicans*, *Cladosporium*, *Eurotium*, *Fennellia*, *Metarrhizium*, *Paecilomyces*, *Phoma*, and *Trichothecium* genera (compiled in-house)	MS, MSI
metabolites/alternaria.txt	138 metabolites of *Alternaria* (compiled in-house)	MS, MSI
metabolites/aspergillus.txt	306 metabolites of *Aspergillus* (compiled in-house)	MS, MSI
metabolites/candida_albicans.txt	11 metabolites of *Candida albicans* (compiled in-house)	MS, MSI
metabolites/cladosporium.txt	54 metabolites of *Cladosporium* (compiled in-house)	MS, MSI
metabolites/eurotium_and_↩ fennellia.txt	44 metabolites of *Eurotium* and *Fennellia* (compiled in-house)	MS, MSI
metabolites/metarrhizium.txt	79 metabolites of *Metarrhizium* (compiled in-house)	MS, MSI
metabolites/paecilomyces.txt	97 metabolites of *Paecilomyces* (compiled in-house)	MS, MSI
metabolites/phoma.txt	52 metabolites of *Phoma* (compiled in-house)	MS, MSI
metabolites/trichothecium.txt	69 metabolites of *Trichothecium* (compiled in-house)	MS, MSI
lipids/LMSD_∗_All.txt	All lipids (compiled from http↩ ://www.lipidmaps.org/)	MS, MSI
lipids/LMSD_∗_FA.txt	Fatty Acyls (compiled from http://www.lipidmaps.↩ org/)	MS, MSI
lipids/LMSD_∗_GL.txt	Glycerolipids (compiled from http://www.lipidmaps.↩ org/)	MS, MSI
lipids/LMSD_∗_GP.txt	Glycerophospholipids (compiled from http://www.↩ lipidmaps.org/)	MS, MSI
lipids/LMSD_∗_PK.txt	Polyketides (compiled from http://www.lipidmaps.↩ org/)	MS, MSI
lipids/LMSD_∗_PR.txt	Prenol Lipids (compiled from http://www.lipidmaps.↩ org/)	MS, MSI
lipids/LMSD_∗_SL.txt	Saccharolipids (compiled from http://www.lipidmaps.↩ org/)	MS, MSI
lipids/LMSD_∗_SP.txt	Sphingolipids (compiled from http://www.lipidmaps.↩ org/)	MS, MSI
lipids/LMSD_∗_ST.txt	Sterol Lipids (compiled from http://www.lipidmaps.↩ org/)	MS, MSI

Location	Content	Mode
custom		Download Template for Excel

2.13 Modifications Editor

2.13.1 Modifications Editor

The editor is opened using the command "Tools -> Modifications Editor" from the Main Window. Use F1 to open this help.
Changes in the editor do not impact the settings of the engine until they are saved !

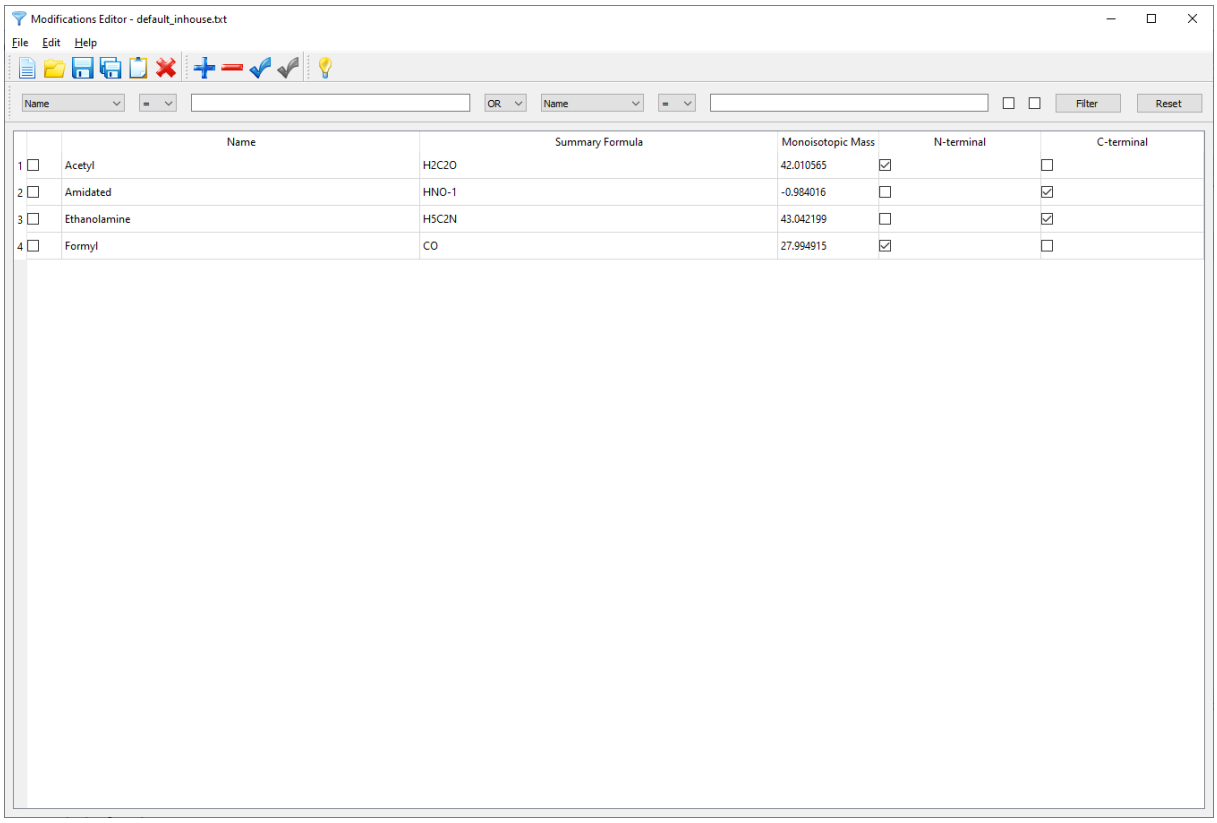


Figure 2.45 Modifications Editor

2.13.1.1 Toolbar

New Database (CTRL + N)
Create a new database of modifications.

Open Database (CTRL + O)
Open a database of modifications.

Save Database (CTRL + S)
Save the database of modifications.

Save Database As... (CTRL + D)
Save the database of modifications with another file name.

Import Database (CTRL + I) 
Import a database of modifications.

Close (Esc) 
Close window.

Insert Row (Insert) 
Insert a new row at the bottom of the table.

Remove Selected Rows (Delete) 
Remove checked rows.

Select All (CTRL + A) 
Select all rows.

Unselect Rows (CTRL + U) 
Unselect all rows.

HTML Documentation (F1) 
Show HTML documentation.

2.13.1.2 Text Filter

See details here.

2.13.1.3 Columns

- Name - Name of an N-terminal or a C-terminal modification.
- Summary Formula - Summary formula of a modification (e.g., H2C2O for acetylation or HNO-1 for amidation). Negative numbers of atoms are supported (e.g., H-2O3N-5 means remove two hydrogens, add three oxygens and remove five nitrogens).
- Monoisotopic Mass - Monoisotopic mass corresponding to a modification. The value is recalculated when the summary formula is changed.
- N-terminal - When checked, the modification is considered to be N-terminal.
- C-terminal - When checked, the modification is considered to be C-terminal.

2.13.2 Format of Modification Databases

The list of modifications is a text file (∗.txt) containing one modification per line. Each line contains the following items separated by tabulars:

- name of a modification
- summary formula of a modification (important: an exact mass of the block is calculated from this value)
- monoisotopic mass (the value is ignored; the mass is automatically calculated from the summary formula)
- the value "1", when the modification is N-terminal; "0" otherwise
- the value "1", when the modification is C-terminal; "0" otherwise

A line starting with the hash '#' is a comment which is not processed.

2.13.3 List of Available Modification Databases

The following pre-defined databases of terminal modifications are available in "Modifications" folder:

Location	Content
acetyl.txt	acetylation
acetyl_amidated.txt	acetylation and amidation
amidated.txt	amidation
default_inhouse.txt	4 modifications (compiled in-house)
ethanolamine.txt	ethanolamine
ethanolamine_formyl.txt	ethanolamine and formyl
norine-terminal-blocks.txt	7 terminal blocks/modifications compiled from Norine (version July, 2013)
putrescine.txt	putrescine
tutorial5.txt	4 modifications used in Tutorial 5
custom	Download Template for Excel

2.14 Draw Peptide Tool

2.14.1 Draw Peptide Tool

The tool is opened using the command "Tools -> Draw Peptide". Use F1 to open this help.

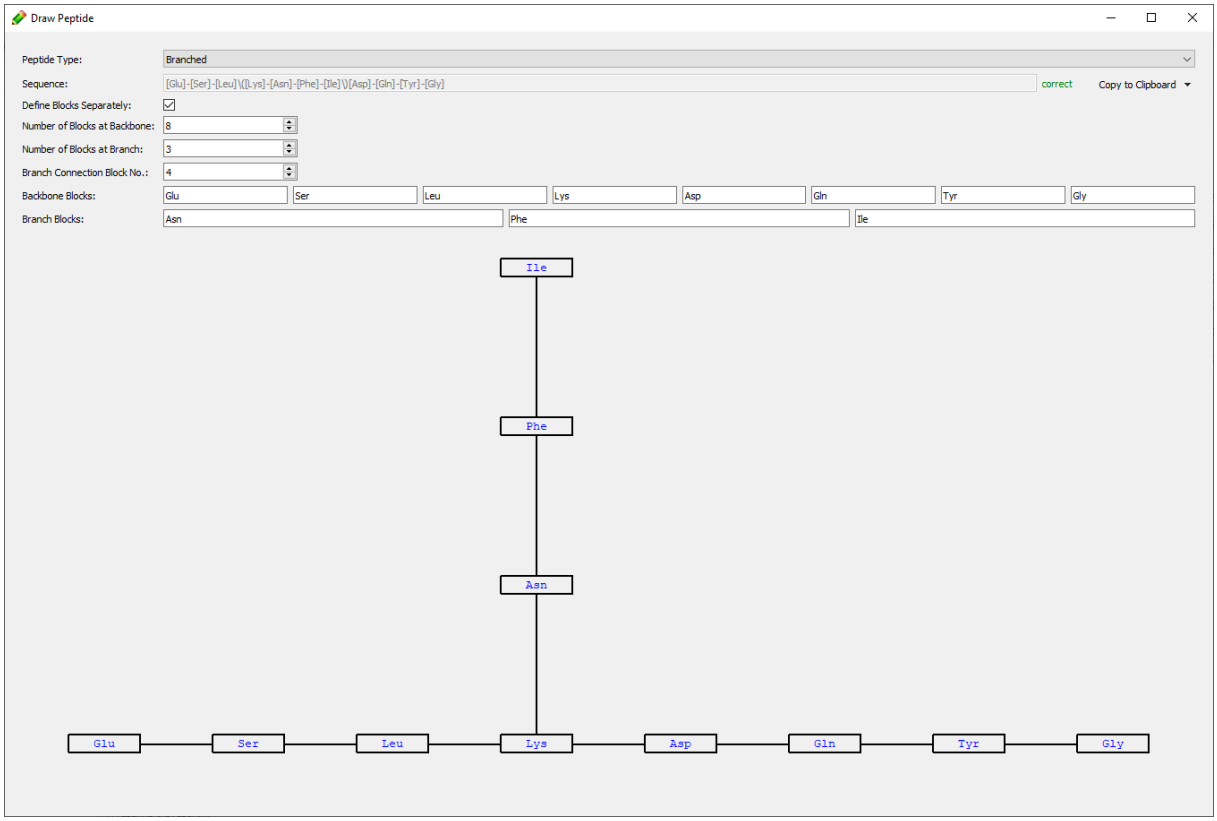


Figure 2.46 Draw Peptide Tool (draw a peptide from an input sequence)

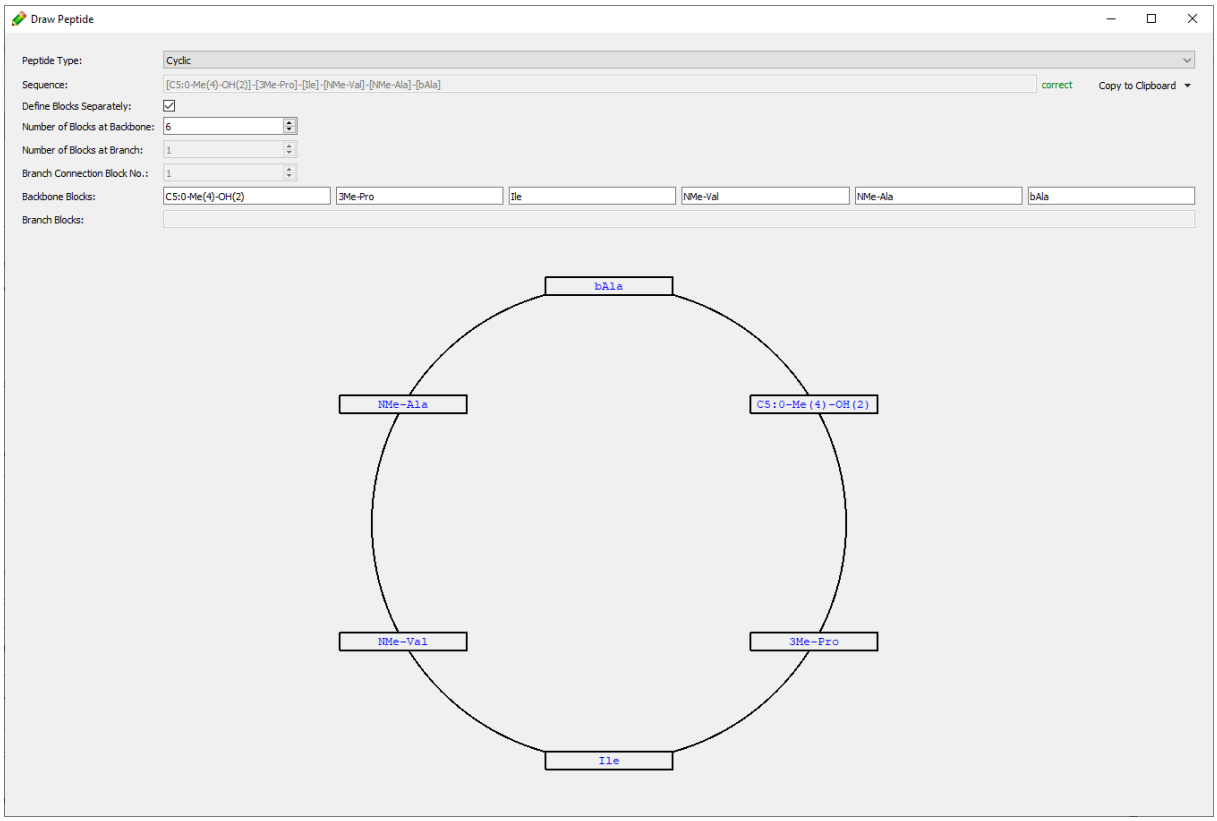


Figure 2.47 Draw Peptide Tool (draw a peptide using a set of building blocks)

2.14.1.1 Controls

Peptide Type
Select the type of peptide (linear, cyclic, branched, branch-cyclic, linear polyketide or cyclic polyketide).
Sequence
A peptide sequence to be drawn. See format of a peptide sequence in Format of Sequence Databases.
The tool automatically checks if the syntax of an entered sequence corresponds to the selected peptide type. If the syntax is correct then a green message "correct" is shown else a red message "incorrect" is shown at the end of the input line.
Copy to Clipboard
The button has following functions which are accessible under an arrow on its right side:

- Copy to Clipboard - copy the sequence into the clipboard; the default action of the button.
- Set as Searched Sequence - the dialog "Search -> Settings..." is opened; the values of "Peptide Type" and "Sequence" are changed to the values defined in the "Draw Peptide Tool".
- Set as Sequence Tag - the dialog "Search -> Settings..." is opened; the values of "Peptide Type" and "Peptide Sequence Tag" are changed to the values defined in the "Draw Peptide Tool".
- Insert into Sequence Database - the tool "Tools -> Sequence/Compound Database Editor" is opened and a new row is added at the end of the table; values of "Type" and "Sequence" on the new row are set up to the values defined in the "Draw Peptide Tool".

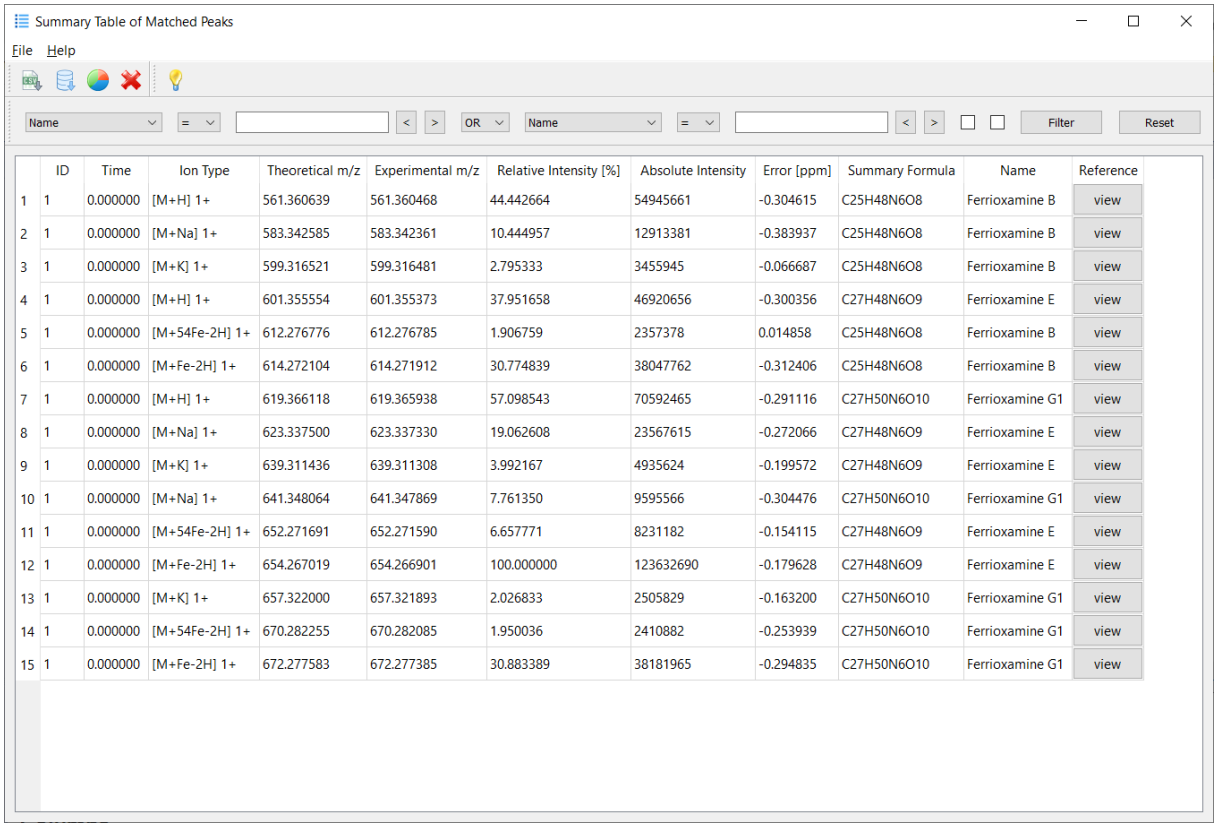
Define Blocks Separately
If the box is checked, the sequence to be drawn is specified by the fields "Number of Blocks at Backbone", "↩ Number of Blocks at Branch", "Branch Connection Block No.", "Backbone Blocks" and "Branch Blocks". The field "Sequence" is disabled and updated automatically.
If the box is unchecked, the sequence to be drawn is specified in the field "Sequence". The fields "Number of Blocks at Backbone", "Number of Blocks at Branch", "Branch Connection Block No.", "Backbone Blocks" and "Branch Blocks" are disabled and updated automatically.
Number of Blocks at Backbone

A number of blocks forming a backbone of a peptide.
Number of Blocks at Branch
A number of blocks forming a branch of a branched or a branch-cyclic peptide.
Branch Connection Block No.
A number of a block where a backbone of a branched or a branch-cyclic peptide is connected to a branch.
Backbone Blocks
Names of building blocks forming a backbone of a peptide. The number of fields is updated when the value of "Number of Blocks at Backbone" is changed. One name of a building block per field must be entered.
Branch Blocks
Names of building blocks forming a branch of a branched or a branch-cyclic peptide. The number of fields is updated when the value of "Number of Blocks at Branch" is changed. One name of a building block per field must be entered.

2.15 Summary Table of Matched Peaks

2.15.1 Summary Table of Matched Peaks

The table is opened using the command "Tools -> Summary Table of Matched Peaks" from the Main Window. Use F1 to open this help.
The table is refreshed automatically when the content of the Output Report in the main window is changed. The content of the table is generated from the rows in the Output Report which are currently visible, i.e., you can use the 'Filter' button in the main window to limit the rows from which the summary table of peaks is generated.



	ID	Time	Ion Type	Theoretical m/z	Experimental m/z	Relative Intensity [%]	Absolute Intensity	Error [ppm]	Summary Formula	Name	Reference
1	1	0.000000	[M+H] 1+	561.360639	561.360468	44.442664	54945661	-0.304615	C25H48N6O8	Ferrioxamine B	view
2	1	0.000000	[M+Na] 1+	583.342585	583.342361	10.444957	12913381	-0.383937	C25H48N6O8	Ferrioxamine B	view
3	1	0.000000	[M+K] 1+	599.316521	599.316481	2.795333	3455945	-0.066687	C25H48N6O8	Ferrioxamine B	view
4	1	0.000000	[M+H] 1+	601.355554	601.355373	37.951658	46920656	-0.300356	C27H48N6O9	Ferrioxamine E	view
5	1	0.000000	[M+54Fe-2H] 1+	612.276776	612.276785	1.906759	2357378	0.014858	C25H48N6O8	Ferrioxamine B	view
6	1	0.000000	[M+Fe-2H] 1+	614.272104	614.271912	30.774839	38047762	-0.312406	C25H48N6O8	Ferrioxamine B	view
7	1	0.000000	[M+H] 1+	619.366118	619.365938	57.098543	70592465	-0.291116	C27H50N6O10	Ferrioxamine G1	view
8	1	0.000000	[M+Na] 1+	623.337500	623.337330	19.062608	23567615	-0.272066	C27H48N6O9	Ferrioxamine E	view
9	1	0.000000	[M+K] 1+	639.311436	639.311308	3.992167	4935624	-0.199572	C27H48N6O9	Ferrioxamine E	view
10	1	0.000000	[M+Na] 1+	641.348064	641.347869	7.761350	9595566	-0.304476	C27H50N6O10	Ferrioxamine G1	view
11	1	0.000000	[M+54Fe-2H] 1+	652.271691	652.271590	6.657771	8231182	-0.154115	C27H48N6O9	Ferrioxamine E	view
12	1	0.000000	[M+Fe-2H] 1+	654.267019	654.266901	100.000000	123632690	-0.179628	C27H48N6O9	Ferrioxamine E	view
13	1	0.000000	[M+K] 1+	657.322000	657.321893	2.026833	2505829	-0.163200	C27H50N6O10	Ferrioxamine G1	view
14	1	0.000000	[M+54Fe-2H] 1+	670.282255	670.282085	1.950036	2410882	-0.253939	C27H50N6O10	Ferrioxamine G1	view
15	1	0.000000	[M+Fe-2H] 1+	672.277583	672.277385	30.883389	38181965	-0.294835	C27H50N6O10	Ferrioxamine G1	view

Figure 2.48 Summary Table of Matched Peaks (MS data)

2.15.1.1 Toolbar

Export to CSV (CTRL + E) 
Export the table into a .csv (comma-separated values) file.

Export to Database (CTRL + D) 
Export matched formulas to a Sequences/Compounds Database - the feature is available only if Compound Search - MS, LC-MS, MSI mode was used.

Export Statistics to CSV (CTRL + I) 
Export statistics into a .csv (comma-separated values) file. Duplicate matches of peaks across all spectra are removed and counted.

Close (Esc) 
Close window.

HTML Documentation (F1) 
Show HTML documentation.

2.15.1.2 Text Filter

See details here. The text filter contains a suggester of names of identified compounds if the mode "Compare Peaklist(s) with Database - MS, LC-MS, MSI" (see Mode) was used. The molecular formulas are used instead of the names if the mode "Compound Search - MS, LC-MS, MSI" was used. The arrows $<$ and $>$ can be used to browse all identified compounds. A compound can also be directly seen in Chromatogram window (LC-MS) or Image Fusion tool (MSI data).

2.15.1.3 Columns

The following columns are reported for MS, LC-MS and MSI data:

- ID - Spectrum ID in the Output Report
- Ion Type - a name of a theoretical peak including information about its charge (if Generate Full Isotope Patterns is disabled)
- Pattern Type - a name of a theoretical isotope pattern including information about its charge (if Generate Full Isotope Patterns is enabled)
- Theoretical m/z - a theoretical mass-to-charge ratio of the peak
- Theoretical intensity [%] - theoretical intensity of the peak (if Generate Full Isotope Patterns is enabled)
- Experimental m/z - a mass-to-charge ratio of a matched peak
- Relative Intensity [%] - a relative intensity of a matched peak
- Absolute Intensity - an absolute intensity of a matched peak
- Error [ppm] - a ppm error
- Score - the angle distance calculated between theoretical and experimental isotopic patterns (if Generate Full Isotope Patterns is enabled)
- FDR - false discovery rate of a matched isotopic pattern (if Generate Full Isotope Patterns is enabled)
- Summary Formula - a summary formula of a neutral molecule
- Name - a name of a compound
- Reference - a link into a database if available (ChemSpider, PubChem or Norine)

The following column is reported for LC-MS data:

- Time - retention time (time units are dataset dependent)

The following columns are reported for MSI data:

- Coordinate X - x coordinate of a single mass spectrum

- Coordinate Y - y coordinate of a single mass spectrum

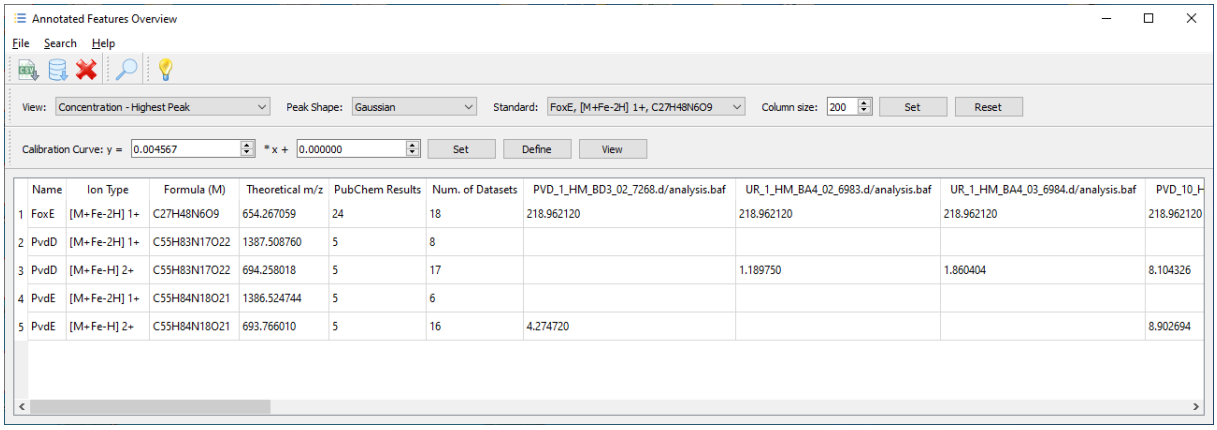
The following columns are reported for MS/MS data:

- ID - Result ID in the Output Report
- Group ID - identifier of a group of peptide sequence candidates generated from the same path in a de novo graph (only if mode is set up to "De Novo Search Engine"; see Mode)
- Fragment Type - a name of a theoretical peak including information about its charge
- Theoretical m/z - a theoretical mass-to-charge ratio of the peak
- Theoretical intensity [%] - theoretical intensity of the peak (if Generate Full Isotope Patterns is enabled)
- Experimental m/z - a mass-to-charge ratio of a matched peak
- Relative Intensity [%] - a relative intensity of a matched peak
- Absolute Intensity - an absolute intensity of a matched peak
- Error [ppm] - a ppm error
- Summary Formula - a summary formula of a fragment ion
- Sequence - a sequence of building blocks corresponding to the peak

2.16 Annotated Features Overview

2.16.1 Annotated Features Overview

The table is opened using the command "Tools -> Annotated Features Overview" from the Main Window. Use F1 to open this help.



	Name	Ion Type	Formula (M)	Theoretical m/z	PubChem Results	Num. of Datasets	PVD_1_HM_BD3_02_7268.d/analysis.baf	UR_1_HM_BA4_02_6983.d/analysis.baf	UR_1_HM_BA4_03_6984.d/analysis.baf	PVD_10_H
1	FoxE	[M+Fe-2H] 1+	C27H48N6O9	654.267059	24	18	218.962120	218.962120	218.962120	218.962120
2	PvdD	[M+Fe-2H] 1+	C55H83N17O22	1387.508760	5	8				
3	PvdD	[M+Fe-H] 2+	C55H83N17O22	694.258018	5	17		1.189750	1.860404	8.104326
4	PvdE	[M+Fe-2H] 1+	C55H84N18O21	1386.524744	5	6				
5	PvdE	[M+Fe-H] 2+	C55H84N18O21	693.766010	5	16	4.274720			8.902694

Figure 2.49 Annotated Features Overview

2.16.1.1 Toolbar

Export to CSV (CTRL + E) 
Export the table into a .csv (comma-separated values) file.

Export to Database (CTRL + D) 
Export matched formulas to a Sequences/Compounds Database - the feature is available only if Compound Search - MS, LC-MS, MSI mode was used.

Close (Esc) 
Close window.

Search PubChem... (CTRL + P) 
Search molecular formulas of compounds in PubChem. Available only in Compound Search - MS, LC-MS, MSI mode.

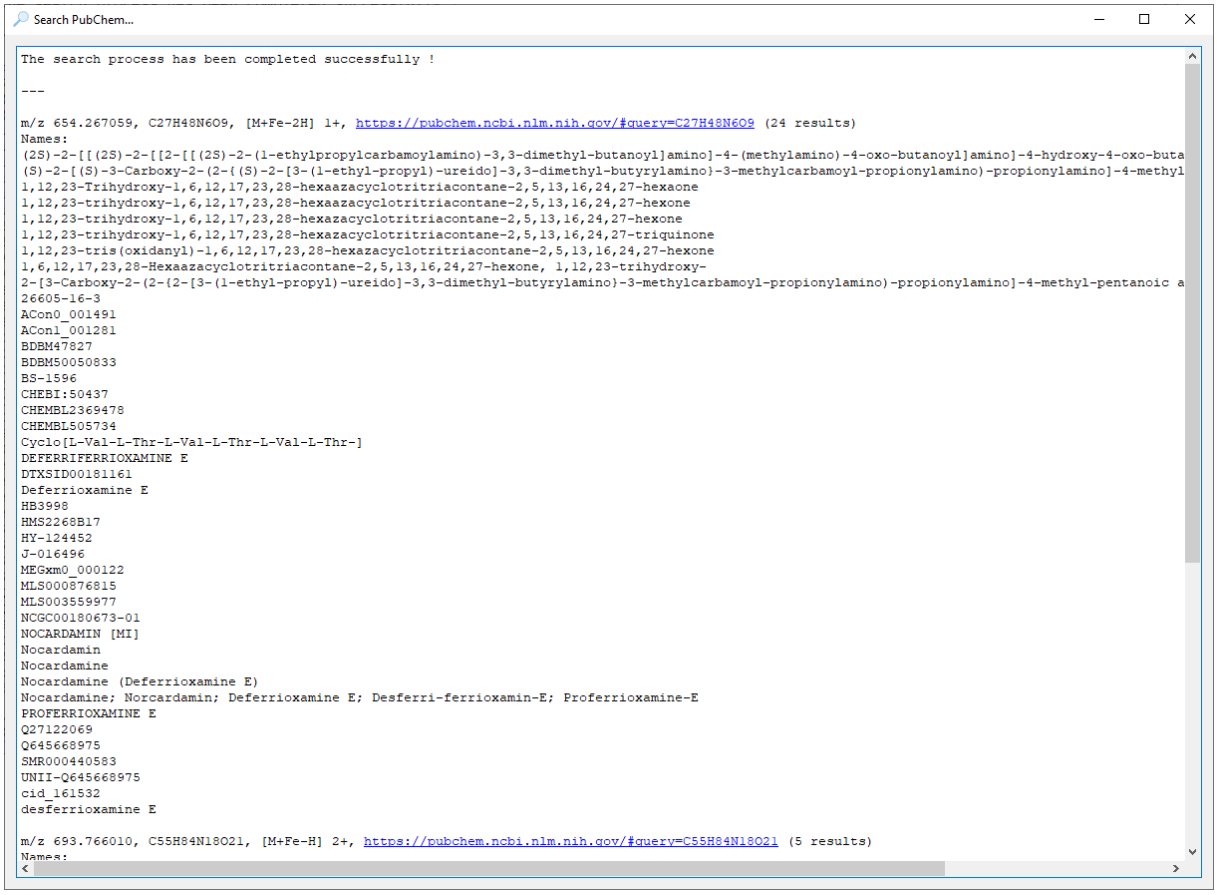


Figure 2.50 Example - names of compounds identified by CycloBranch were searched in PubChem.

HTML Documentation (F1) 
Show HTML documentation.

View
Select a value type which is reported for every compound and every dataset.

- Maximum Relative Intensity - maximum relative intensity of a given compound in the respective LC-MS data file.
- Maximum Absolute Intensity - maximum absolute intensity of a given compound in the respective LC-MS data file.
- Peak Area (Chromatogram) - All Isotopes - peak area under the extracted ion chromatogram of a given compound
  (the extracted ion chromatogram is constructed from the sum of absolute intensities of all isotopes of a given compound in every spectrum).
- Concentration - All Isotopes - concentration of a given compound calculated using the respective peak area
  (the extracted ion chromatogram is constructed from the sum of absolute intensities of all isotopes of a given compound in every spectrum).

- Peak Area (Chromatogram) - Highest Peak - peak area under the extracted ion chromatogram of a given compound
  (the extracted ion chromatogram is constructed from the highest intensity peak of a given compound in every spectrum).
- Concentration - Highest Peak - concentration of a given compound calculated using the respective peak area
  (the extracted ion chromatogram is constructed from the highest intensity peak of a given compound in every spectrum).
- Retention Time

Peak Shape
Select a function which is used to estimate shape of a chromatographic peak in profile mode from peaks in line mode.

- Gaussian - common Gaussian function
- ex-Gaussian (tailing) - exponentially modified Gaussian function with tailing
- ex-Gaussian (fronting) - exponentially modified Gaussian function with fronting

Standard
All compounds detected in at least one LC-MS datafile are listed.
Calibration Curve
The equation is used if View is set to "Concentration - All Isotopes" and "Concentration - Highest Peak". Enter the input parameters a and b for the equation y = a $*$ x + b, where x is concentration and y is peak area. If Standard is selected, the peak area of a given compound is automatically divided by peak area of standard.
Buttons:

- "Set" - set the parameters a and b, and recalculate the table.
- "Define" - define samples, calibration curve and recalculate the table.
- "View" - view the calibration curve and statistics.

Define Samples and Calibration Curve

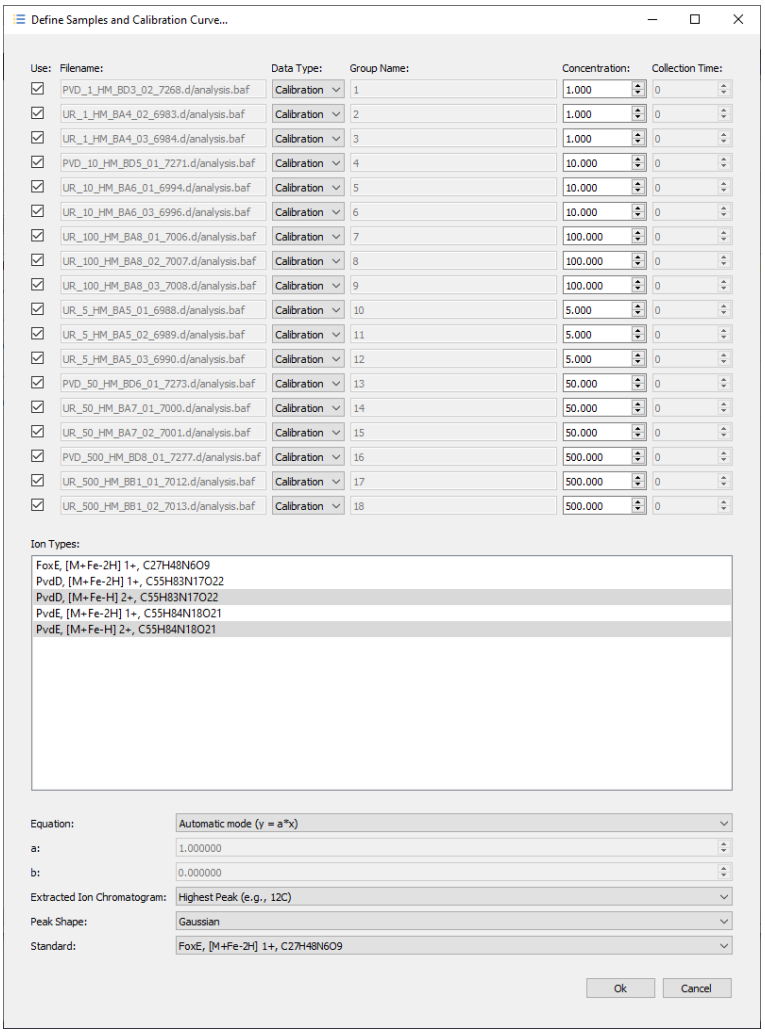


Figure 2.51 Define parameters of calibration curve.

The parameters a and b of a calibration curve can be estimated from an input set of LC-MS data files. Check the filenames which should be used to estimate the parameters of curve. In our example, a mixture of PvdD and PvdE was spiked into the respective samples in concentrations 1-500 ng/mL. The samples were collected in triplicates. The values of concentration were entered into respective fields. If a value is used more than once, the results were averaged. The areas of chromatographic peaks of selected ions (i.e., PvdD and PvdE) were summed. The resulting area was divided by the area of selected internal standard (i.e., FoxE).

If the slope (parameter a) is the only parameter which should be estimated, select the equation y = a∗x. If the slope (parameter a) and y-intercept (parameter b) should be estimated, select the equation y=a∗x + b. The extracted ion chromatogram of a given compound can be created by two different approaches - using 'Highest Peak (e.g., 12C)' (the extracted ion chromatogram is constructed from the highest intensity peak of a given compound in every spectrum) or 'All Isotopes (e.g., sum of 12C + 13C + 13C2 + ...)' (the extracted ion chromatogram is constructed from the sum of absolute intensities of all isotopes of a given compound in every spectrum). When calculating the area of chromatographic peaks, the peak shape can be estimated using Gaussian function, exponentially modified Gaussian function with tailing, and exponentially modified Gaussian function with fronting.

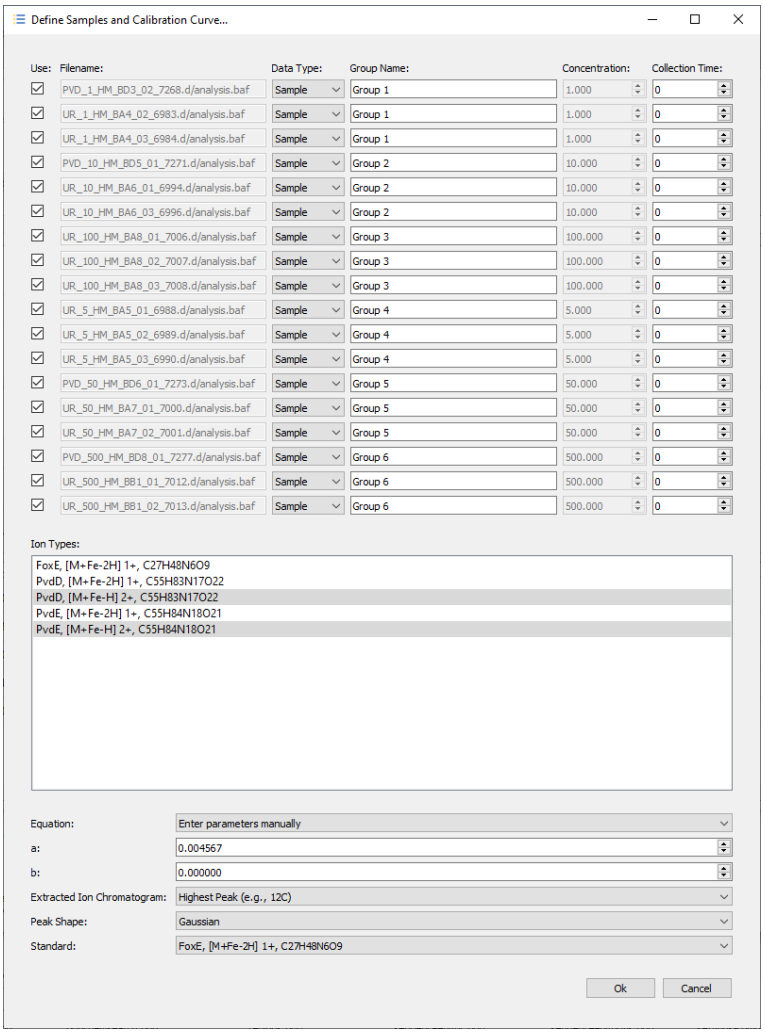
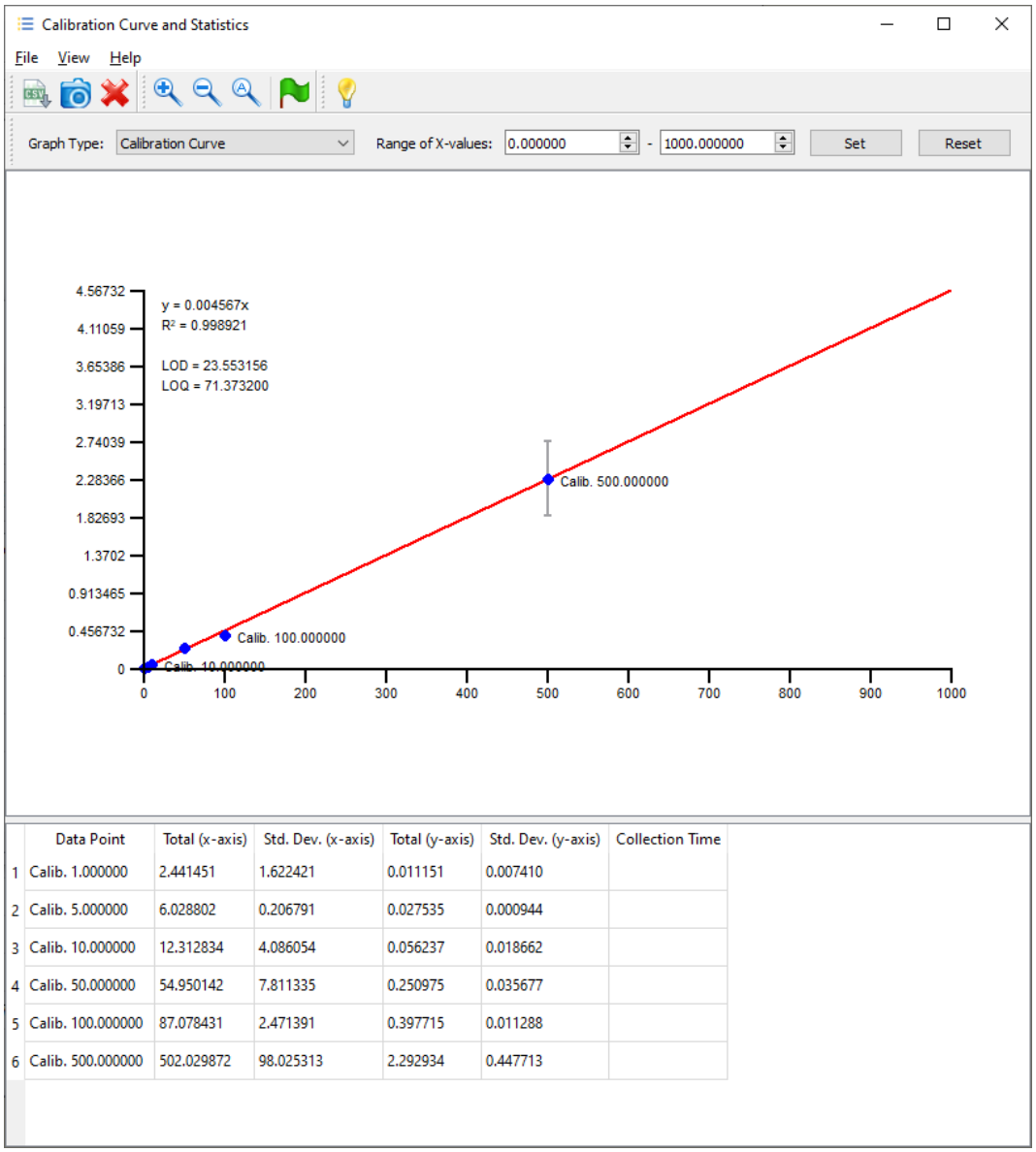


Figure 2.52 Define groups of samples.

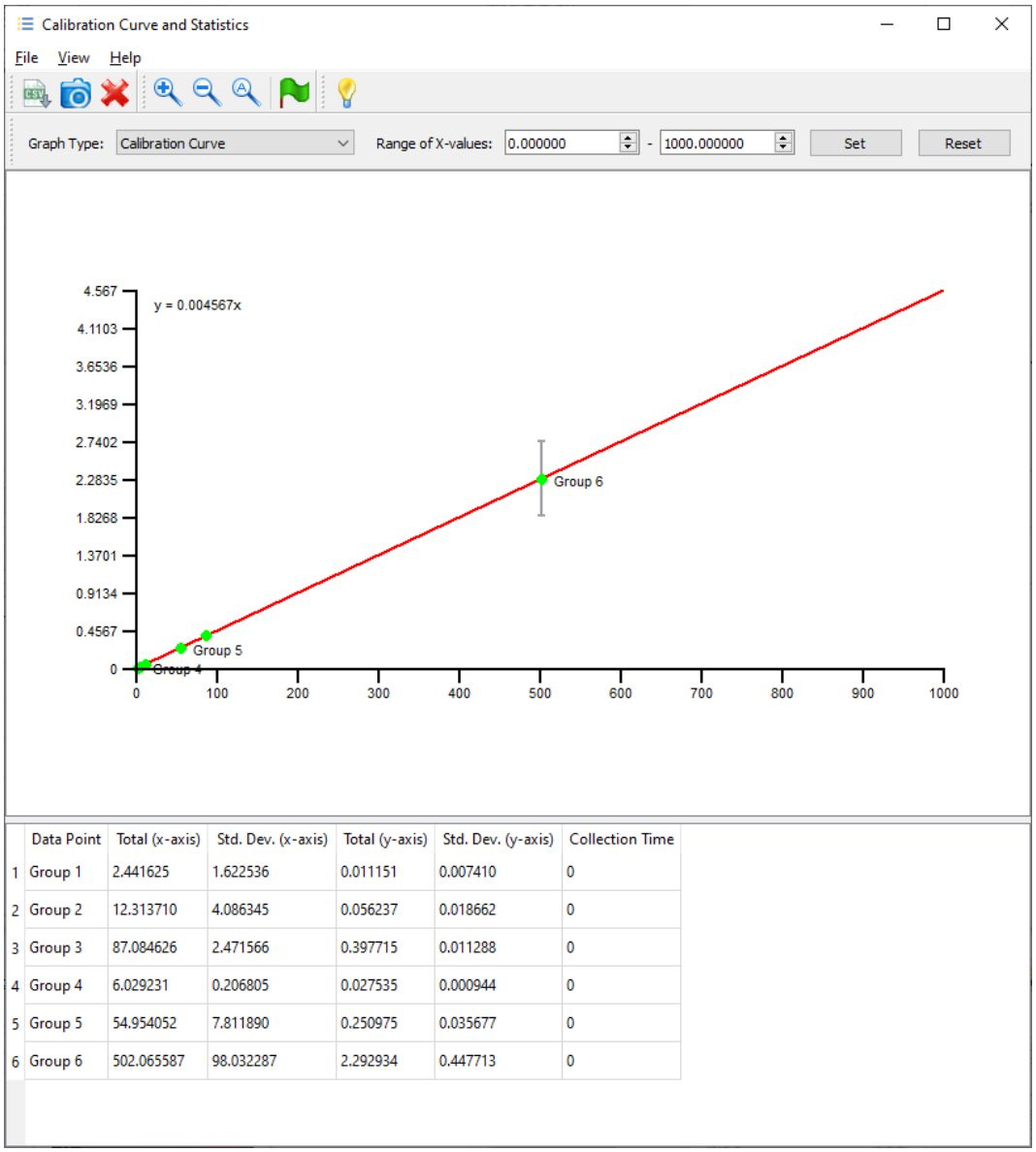
If data files with unknown samples are analyzed, choose the value 'Sample' in 'Data Type'. Annotate technical replicates with the same Group number. The results are then automatically averaged. If calibration data files are selected at the same time, the parameters of calibration curve can be estimated automatically. Optionally, choose 'Enter parameters manually' in 'Equation' field and type the parameters 'a' and 'b' manually.

View Calibration Curve and Statistics



	Data Point	Total (x-axis)	Std. Dev. (x-axis)	Total (y-axis)	Std. Dev. (y-axis)	Collection Time
1	Calib. 1.000000	2.441451	1.622421	0.011151	0.007410	
2	Calib. 5.000000	6.028802	0.206791	0.027535	0.000944	
3	Calib. 10.000000	12.312834	4.086054	0.056237	0.018662	
4	Calib. 50.000000	54.950142	7.811335	0.250975	0.035677	
5	Calib. 100.000000	87.078431	2.471391	0.397715	0.011288	
6	Calib. 500.000000	502.029872	98.025313	2.292934	0.447713	

Figure 2.53 View calibration curve and statistics.



	Data Point	Total (x-axis)	Std. Dev. (x-axis)	Total (y-axis)	Std. Dev. (y-axis)	Collection Time
1	Group 1	2.441625	1.622536	0.011151	0.007410	0
2	Group 2	12.313710	4.086345	0.056237	0.018662	0
3	Group 3	87.084626	2.471566	0.397715	0.011288	0
4	Group 4	6.029231	0.206805	0.027535	0.000944	0
5	Group 5	54.954052	7.811890	0.250975	0.035677	0
6	Group 6	502.065587	98.032287	2.292934	0.447713	0

Figure 2.54 View groups of samples.

Export to CSV (CTRL + E) 
Export the table into a .csv (comma-separated values) file.

Export Image (CTRL + G) 
Export the image into a PDF, PS (Windows only), PNG or SVG file. PS files in Linux can be generated outside of the application, e.g., using the pdf2ps tool.

Close (Esc) 
Close window.

Zoom In (CTRL +) 
Zoom in.

Zoom Out (CTRL -) 
Zoom out.

Zoom Reset (CTRL + R) 

Reset zoom.

Hide Labels (CTRL + L) 
Hide/show labels of calibration points, limit of detection and limit of quantification.

HTML Documentation (F1) 
Show HTML documentation.
Concentration

- First field - a minimum value of concentration to be shown on x-axis.
- Second field - a maximum value of concentration to be shown on x-axis.
- "Set" button (or Enter key if the field is focused) - apply the changes.
- "Reset" button - reset the range of concentration.

2.16.1.2 Columns

The following columns are reported:

- Name - compound name
- Ion Type - detected ion type
- Formula (M) - molecular formula (M) of the compound
- Theoretical m/z - theoretical mass-to-charge ratio of the highest intensity peak
- PubChem Results - number of results with the same formula which were found in PubChem
- Num. of Datasets - number of datasets in which the respective ion of the compound was detected

The other columns correspond to analyzed datasets. If a user clicks on an item, the corresponding extracted ion chromatogram is shown in Chromatogram window. If a user clicks on the item and CTRL is pressed, the detail of a mass spectrum is opened where the peak with the highest absolute intensity was found. If a user clicks on an item in the columns Formula (M) and PubChem Results, the formula is searched in PubChem and the results are opened in a web-browser. If a user clicks on the column Name, a reference is opened in a web-browser if it was defined in Sequence/Compound Database Editor. If any reference was not defined, the compound name is searched on PubChem.

2.17 Chromatogram

2.17.1 Chromatogram

The tool is opened using the command "Tools -> Chromatogram" from the Main Window. The function is available if LC-MS data are processed. Use F1 to open this help.

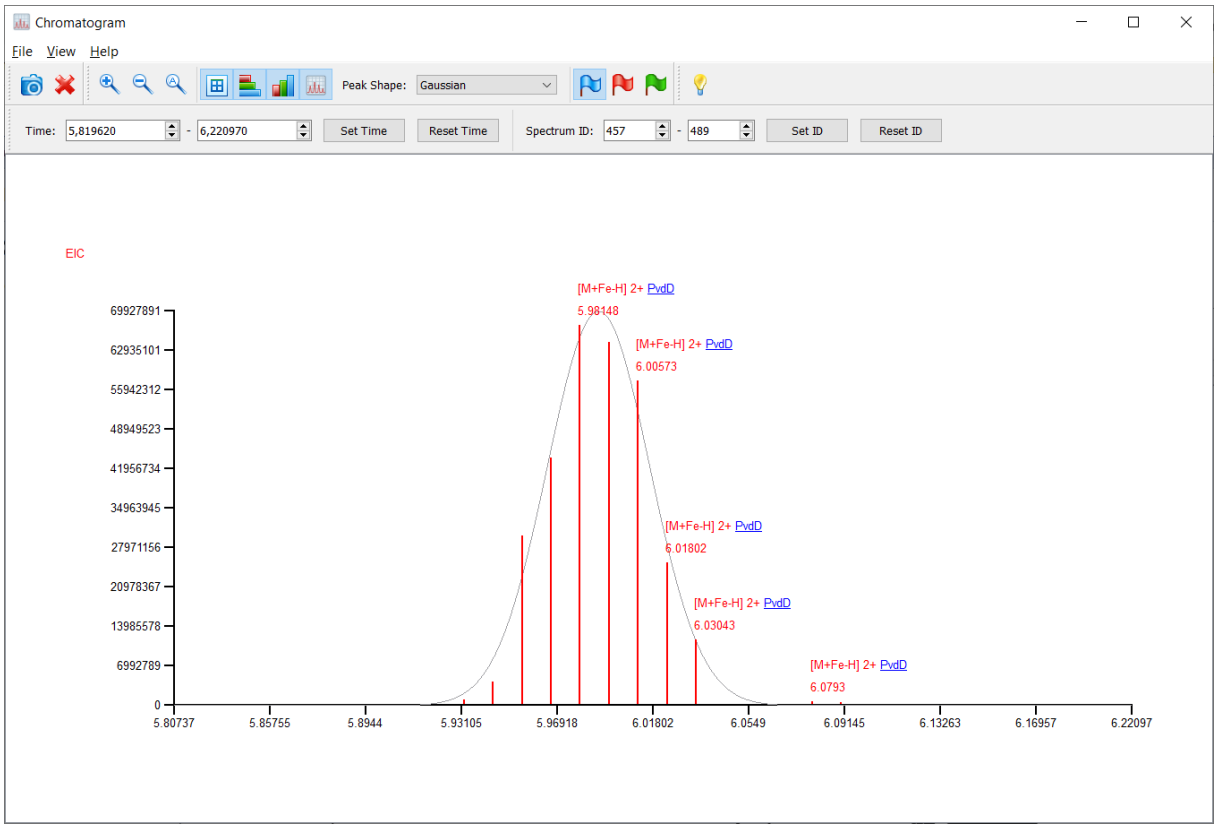


Figure 2.55 Example Chromatogram

2.17.1.1 Toolbar

Export Image (CTRL + E) 
Export the image into a PDF, PS (Windows only), PNG or SVG file. PS files in Linux can be generated outside of the application, e.g., using the pdf2ps tool.

Close (Esc) 
Close window.

Zoom In (CTRL +) 
Zoom in.

Zoom Out (CTRL -) 
Zoom out.

Zoom Reset (CTRL + R) 
Reset zoom.

Mouse Selection Tool (CTRL + T) 
If the button is enabled, the mouse can be used to select a range retention time (or IDs) to be shown.

- Left mouse button - press and hold the button to draw a rectangle determining the minimum and maximum values of retention time (IDs); release the button to apply the changes.
- Right mouse button - press the button during the drawing of a rectangle to cancel the drawing.
- Middle mouse button - reset the range of retention time (IDs).
- Mouse wheel up - zoom the interval in.

- Mouse wheel down - zoom the interval out.

If the button is disabled, the mouse can be used to move the interval of retention time (IDs) to the left or right. Press and hold the left mouse button to shift the chromatogram.

Retention Time (CTRL + M) 
View retention time or IDs of spectra.

Absolute Intensity (CTRL + I) 
View absolute/relative intensities.

Profile Chromatogram (CTRL + P) 
View profile chromatogram. The shape of peak in profile mode is estimated from peaks in line mode. The following functions are available to estimate the shape - Gaussian, ex-Gaussian (tailing), and ex-Gaussian (fronting).

Hide TIC (CTRL + Shift + T) 
Hide/show total ion current chromatogram (TIC).

Hide Matched Peaks (CTRL + Shift + E) 
Hide/show extracted ion chromatogram (EIC). EIC is calculated from the ions which are currently listed in Summary Table of Matched Peaks.

Hide Labels (CTRL + L) 
The labels of chromatographic peaks are hidden if the button is checked.

HTML Documentation (F1) 
Show HTML documentation.

Time

- First field - a minimum value of retention time to be shown.
- Second field - a maximum value of retention time to be shown.
- "Set Time" button (or Enter key if the field is focused) - apply the changes.
- "Reset Time" button - reset the range of retention time.

Spectrum ID

- First field - a minimum value of ID to be shown.
- Second field - a maximum value of ID to be shown.
- "Set ID" button (or Enter key if the field is focused) - apply the changes.
- "Reset ID" button - reset the range of IDs.

2.18 Image Fusion

2.18.1 Image Fusion

The tool is opened using the command "Tools -> Image Fusion" from the Main Window. The function is available if the imaging mass spectra are processed (∗.imzML file). Use F1 to open this help.

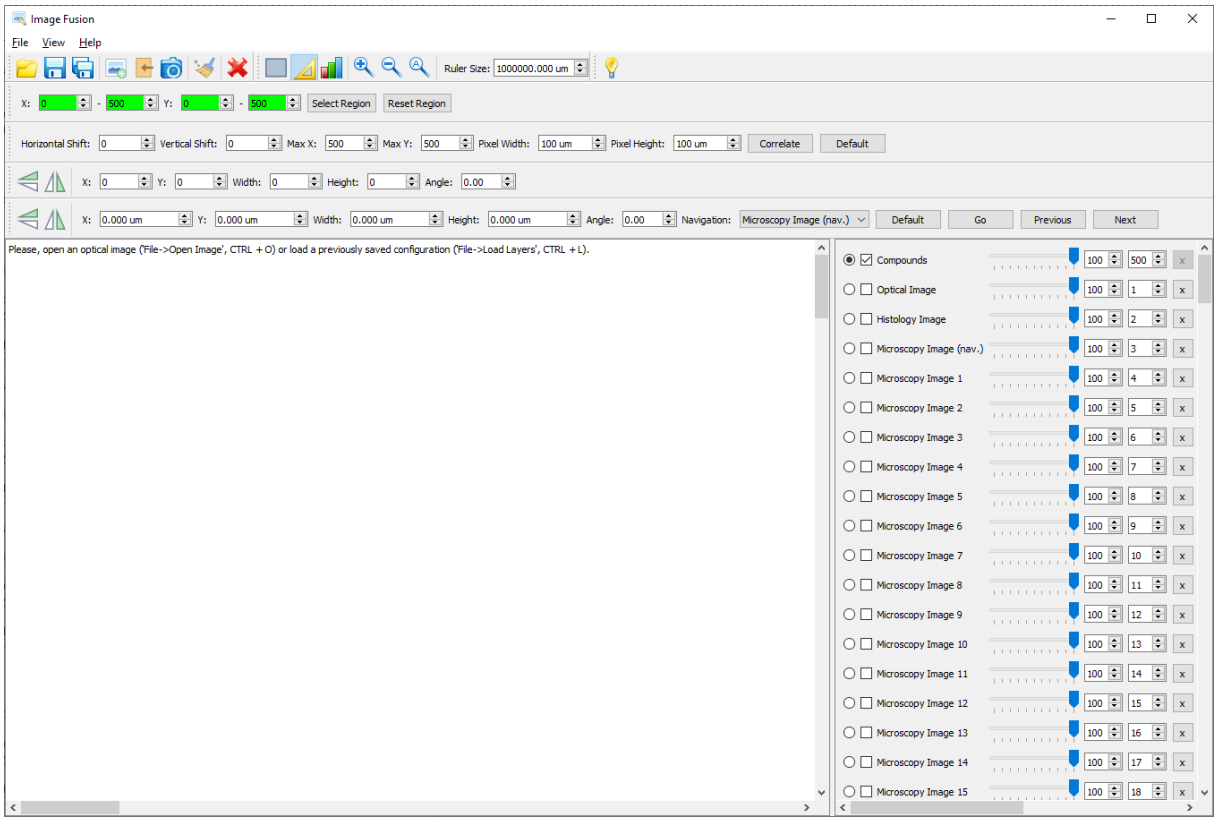


Figure 2.56 Image Fusion window.

2.18.1.1 Toolbar

Load Layers (CTRL + L) 
Load a layers file.

Save Layers (CTRL + S) 
Save a layers file.

Save Layers As... (CTRL + D) 
Save the layers file using another name.

Open Image (CTRL + O) 
Open an optical/histology/microscopy image.

Import Images... (CTRL + M) 
Import scanning electron microscopy images (FEI) or optical and fluorescence images (Leica) in a batch.

Export Image (CTRL + G) 
Export the current image (fusion of different layers).

Clear All Layers (CTRL + W) 
Remove all images.

Close (Esc) 
Close window.

Show Selection (Ctrl + E) 

Show current selection rectangle.

Color Scale and Ruler (CTRL + C) 
Show/Hide color scale and ruler.

Absolute Intensity (CTRL + I) 
View absolute/relative intensities of peaks.

Zoom In (CTRL +; mouse wheel up) 
Zoom in.

Zoom Out (CTRL -; mouse wheel down) 
Zoom out.

Zoom Reset (CTRL + R; mouse wheel pressed) 
Reset zoom.

Ruler Size
The current size of ruler [micrometers].

HTML Documentation (F1) 
Show HTML documentation.

2.18.1.2 Layers

The list of layers is on the right side of the Image Fusion tool. The radio button is used to select an active layer. The checkbox is used to show/hide a layer. The slider and the number in the first spin box is the opacity of a layer (0 = layer is not visible; 100 = layer is not transparent). The second spin box is the order of a layer (z-value).
Each layer has an associated toolbar. The toolbar associated with a layer is green if the corresponding layer is selected (i.e., the corresponding radiobutton is checked). The green values can be also modified by mouse selection.

Compounds
The range of X and Y coordinates to be selected. The button 'Select Region' applies the changes (Enter key can also be used if a corresponding text field is focused). The 'Reset Region' button clears the selection. The selection can also be performed using mouse. To draw a rectangle, press and hold the left mouse button. Draw a rectangle as shown in the following figure and then release the left mouse button. You can click on the right mouse button during the rectangle drawing to cancel the selection. The size of the region can be adjusted and the region can be moved by mouse.
The number of selected points and the minimum bounding region of the selection is reported under the image. The selection of X and Y intervals automatically filters rows in the Summary Table of Matched Peaks. If a compound was filtered by its name or m/z value in Summary Table of Matched Peaks, the filtered text is shown. Both selection methods can be used together.
The standard HSL (hue, saturation, lightness) color model is used to visualize the points. The most intense point in a selected region is red and minimum intensity points are violet. A color represents a sum of relative/absolute intensities of matched peaks in a single spectrum; or the intensity of a specific compound if it was filtered by its name or m/z value in the Summary Table of Matched Peaks.

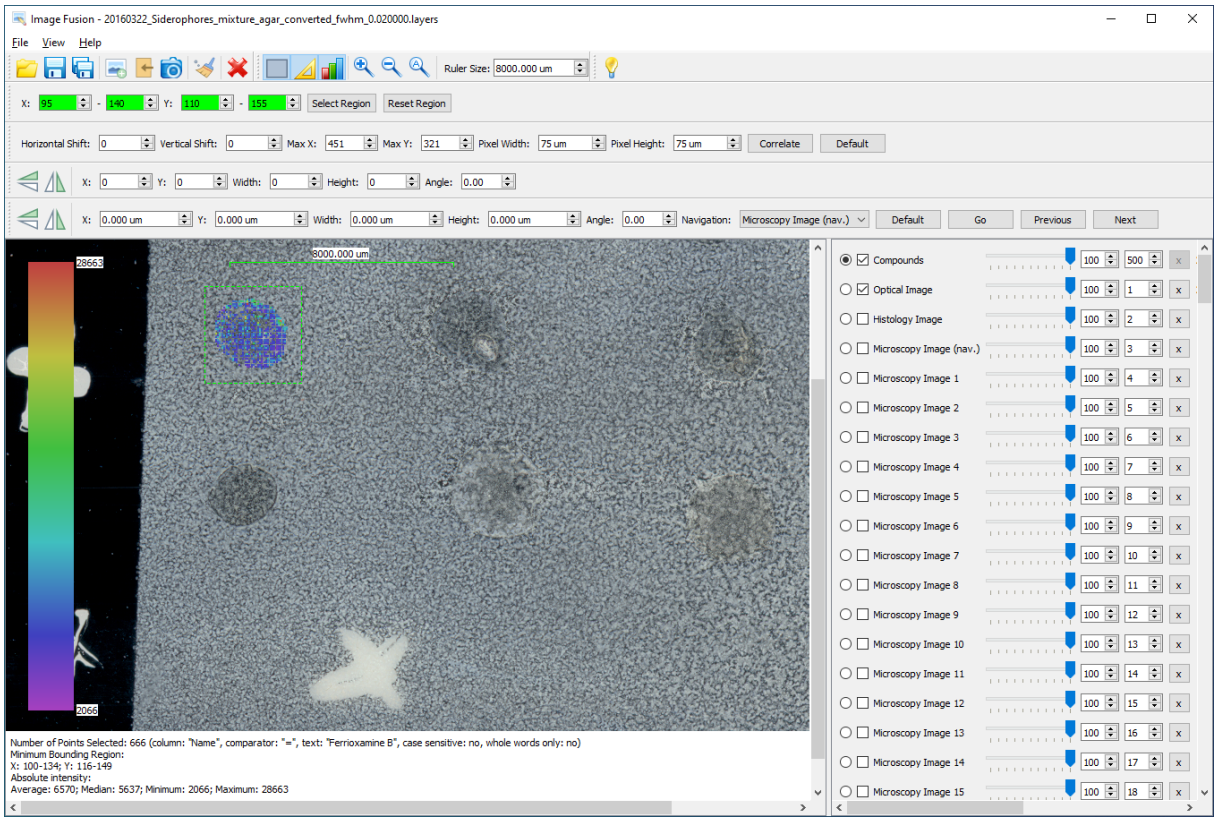


Figure 2.57 Image Fusion - selection of compounds

Optical Image

The toolbar is used to correlate the measured region of compounds with optical image (see below for vendor specific details). The region can also be adjusted or moved by mouse as stated above.

- 'Horizontal Shift' - horizontal shift of points between measured area and the optical image [number of points]
- 'Vertical Shift' - vertical shift of points between measured area and the optical image [number of points]
- 'Max X' and 'Max Y' - the [X,Y] coordinates of the bottom-right corner of an optical image [number of points] (see below for vendor specific details)
- 'Pixel Width' - width of a pixel [micrometers]
- 'Pixel Height' - height of a pixel [micrometers]
- 'Correlate' button (or Enter key if 'Horizontal Shift', 'Vertical Shift', 'Max X' or 'Max Y' are focused) - correlate the measured region of compounds with optical image
- 'Default' - use the default values from input imzML file (see below for vendor specific details)

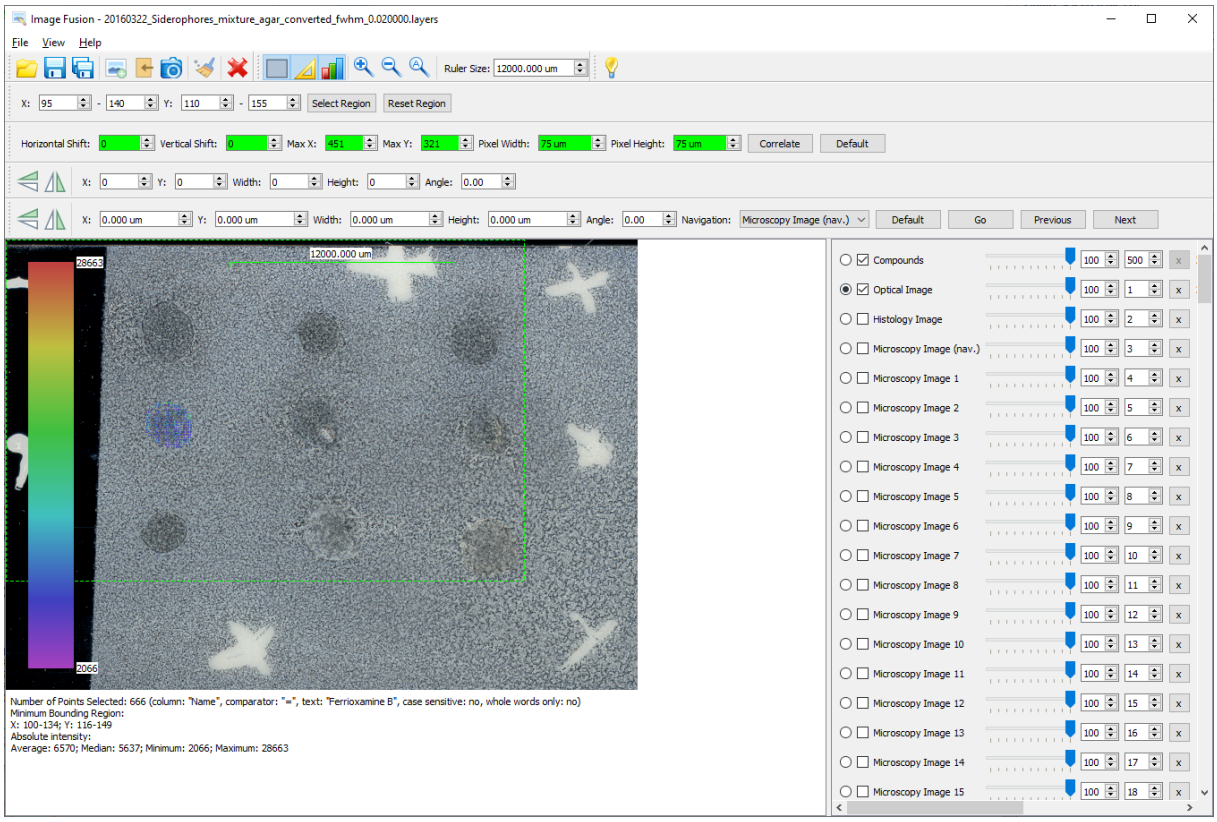


Figure 2.58 Image Fusion - correlation of imzML exported by FlexImaging 4.1 with optical image

Histology Image

The toolbar is used to correlate the histology image with an optical image. Then the identified compounds are visualized in histology image. The size and position of histology image can also be adjusted by mouse.

Flip Histology Horizontally (Ctrl + H)



Flip histology image horizontally.

Flip Histology Vertically (Ctrl + V)



Flip histology image vertically.

- 'X' - horizontal position of histology image [pixels]
- 'Y' - vertical position of histology image [pixels]
- 'Width' - width of histology image [pixels]
- 'Height' - height of histology image [pixels]
- 'Angle' - rotation angle of histology image [degrees]

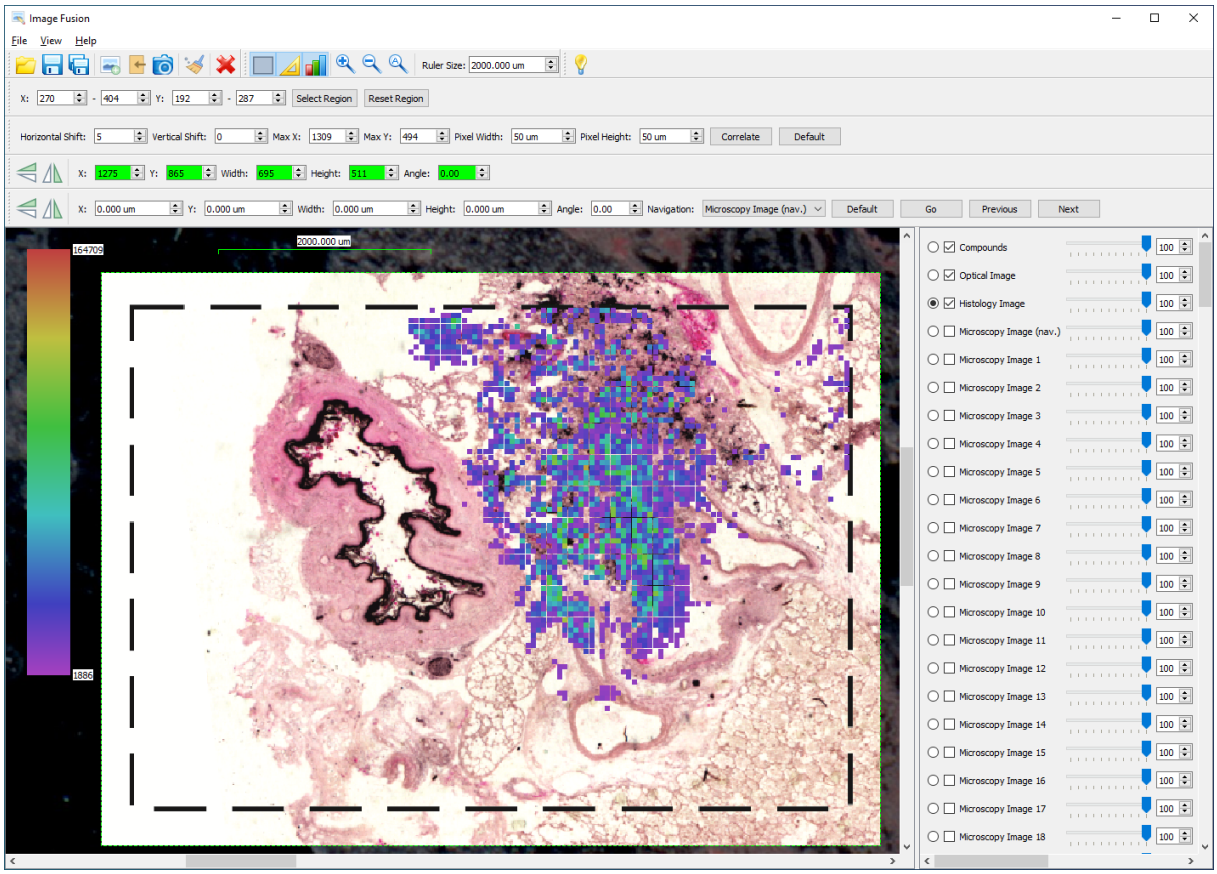


Figure 2.59 Image Fusion - positioning of histology image with an optical image

Microscopy Image

The toolbar is used to correlate a microscopy image with an optical image. Then the identified compounds are visualized in microscopy image. The toolbar is associated with one navigation microscopy image "Microscopy image (nav.)" having a small magnification and multiple non-navigation microscopy images with bigger magnifications. The values of X, Y, Width, Height, Angle, and Navigation correspond to an active microscopy image which is currently selected in the list of layers on the right side.

Flip Microscopy Horizontally (Ctrl + Shift + H) 

Flip microscopy image horizontally.

Flip Microscopy Vertically (Ctrl + Shift + V) 

Flip microscopy image vertically.

- 'X' - horizontal position of microscopy image [micrometers]
- 'Y' - vertical position of microscopy image [micrometers]
- 'Width' - width of microscopy image [micrometers]
- 'Height' - height of microscopy image [micrometers]
- 'Angle' - rotation angle of microscopy image [degrees]
- 'Navigation' dropdown menu - change navigation layer of the current image
- 'Default' - use the default values from the TIFF file
- 'Go' - go to the coordinates of microscopy image
- 'Previous' - go to the coordinates of previous microscopy image
- 'Next' - go to the coordinates of next microscopy image

Prefered input file formats:

- TIFF - generated by FEI Nova NanoSEM 450
- LIF - Leica image file format (the file may contain multiple images; images from optical and fluorescence microscopy are supported)

If the images are in the above listed file formats, then the size in micrometers is calculated automatically from metadata stored in the input file. If a different image type is used, the size of the image must be adjusted manually. The size and position of microscopy image can also be adjusted by mouse. If the image is being resized by mouse, press and hold the CTRL key to keep the original aspect ratio of the image.
It is expected that the navigation image has a small maginification (e.g., 17x) and that non-navigation images have bigger magnifications (e.g. 50x, 100x, 1000x). The size of the image in micrometers is adjusted automatically if the input file contains corresponding metadata. In case of a navigation image, the size is adjusted automatically but the position in the optical image must be adjusted manually. In case of a SEM non-navigation image, both - the size and the position are adjusted automatically. In case of a non-navigation image in LIF file format, the size is adjusted automatically but the position may be aproximate and may require a manual correction.

2.18.1.3 Vendor-specific Details

imzML exported from FlexImaging 4.1/5.0 (Bruker)
The correct values of 'Max X' and 'Max Y' must be determined by hand as they are not included in the imzML file. Open the dataset in FlexImaging 4.1 and determine the coordinates of the bottom-right corner of an optical image as shown in the following image. R00X451Y321 means that Max X = 451 and Max Y = 321.

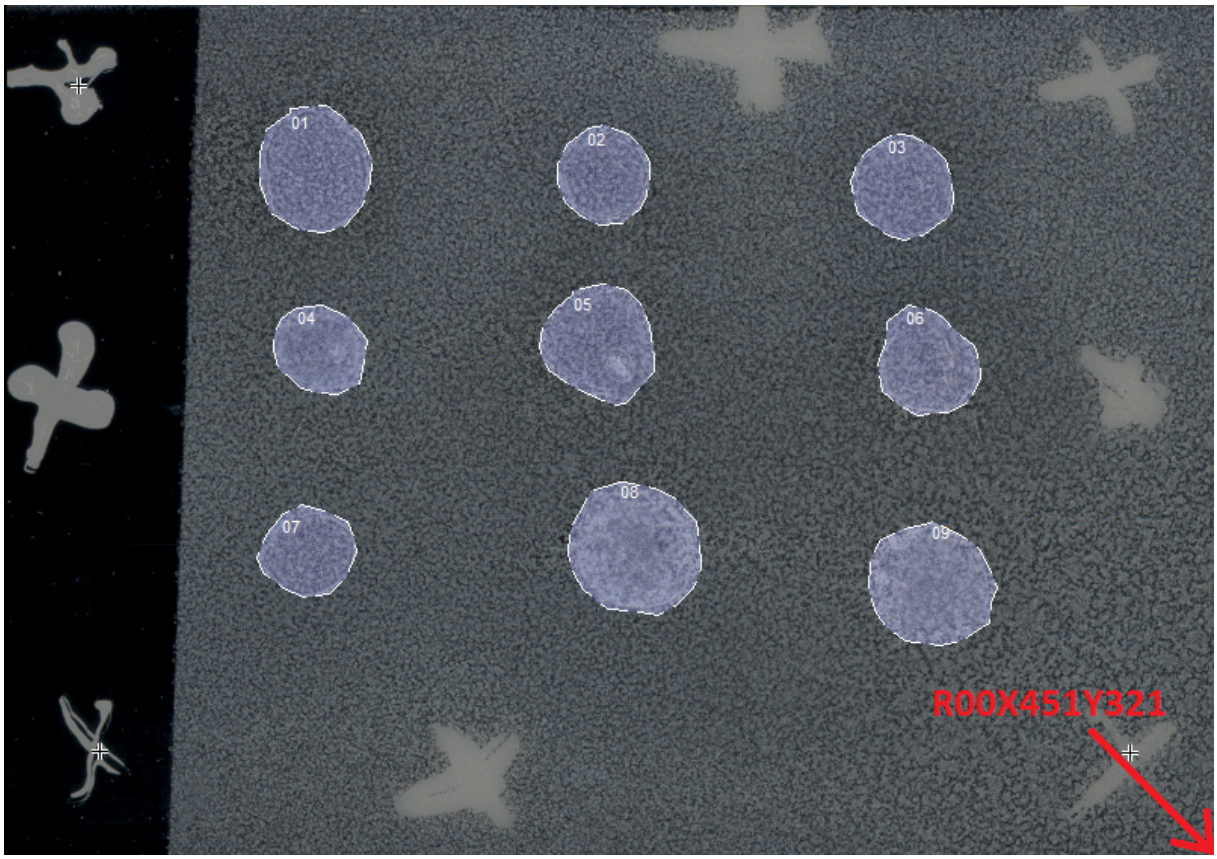


Figure 2.60 Coordinates of the bottom-right image corner.

For the above example, the correct settings are

- Horizontal Shift = 0
- Vertical Shift = 0
- Max X = 451

- Max Y = 321

imzML exported from HDImaging v1.4 (Waters)
In HDImaging v1.4 click on the 'Acquire' tab, then open a pattern definition metafile (∗.pdm). Click on the measured region and then click on the 'Export' icon. In the image export dialog, check the option 'Crop to selection' and uncheck the option 'Overlay regions'. Then click 'Export' and save the file. Hint: Enlarge the main window of HDImaging as much as possible to make the exported image bigger.
The correct values of 'Max X' and 'Max Y' are determined automatically from the imzML file. It may be necessary to modifify the values of 'Horizontal Shift' and 'Vertical Shift' manually to correlate the points with the optical image correctly.

2.19 Simulate Isotopic Pattern

2.19.1 Simulate Isotopic Pattern

The tool is opened using the command "Tools -> Simulate Isotopic Pattern" from the Main Window. Use F1 to open this help.

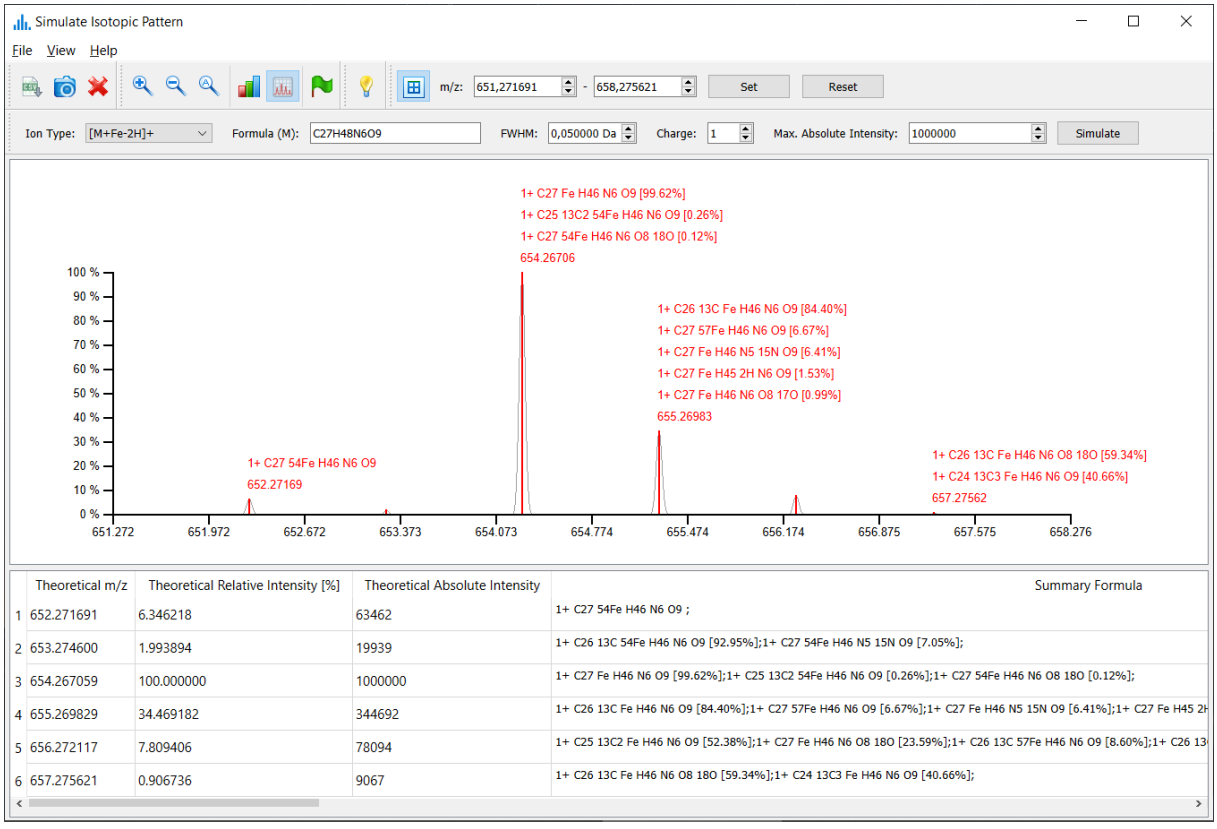


Figure 2.61 Simulated isotopic pattern of ferrioxamine E.

2.19.1.1 Toolbar

Export Table to CSV (CTRL + E) 
Export the table of peaks into a csv (comma separated values) file.

Export Image (CTRL + G) 
Export the image into a PDF, PS (Windows only), PNG or SVG file. PS files in Linux can be generated outside of the application, e.g., using the pdf2ps tool.



Close (Esc)
Close window.



Zoom In (CTRL +)
Zoom in.



Zoom Out (CTRL -)
Zoom out.



Zoom Reset (CTRL + R)
Reset zoom.



Absolute Intensity (CTRL + I)
View absolute/relative intensities of peaks.



Profile Spectrum (CTRL + P)
View profile spectrum.



Hide Peak Labels (CTRL + L)
The labels of peaks are hidden if the button is checked.



HTML Documentation (F1)
Show HTML documentation.



Mouse m/z Selection Tool (CTRL + T)
If the button is enabled, the mouse can be used to select a range of m/z ratios to be shown.

- Left mouse button - press and hold the button to draw a rectangle determining the minimum and maximum values of m/z ratios; release the button to apply the changes.
- Right mouse button - press the button during the drawing of a rectangle to cancel the drawing.
- Middle mouse button - reset the range of m/z ratios.
- Mouse wheel up - zoom the interval in.
- Mouse wheel down - zoom the interval out.

If the button is disabled, the mouse can be used to shift the interval of m/z ratios to the left or right. Press and hold the left mouse button to shift the spectrum.

m/z

- First field - a minimum value of m/z ratio to be shown.
- Second field - a maximum value of m/z ratio to be shown.
- "Set" button (or Enter key) - apply the changes.
- "Reset" button - reset the range of m/z ratios.

Ion type
Select the ion type whose isotopic pattern will be generated. The custom ion types can be defined in the Preferences dialog.

Formula
The molecular formula of a neutral molecule.

FWHM
Enter a value of full width at half maximum.

Charge
Enter a value of charge. Use the positive value for positive ion and negative value for negative ion.

Max. Absolute Intensity

Enter an absolute intensity of highest peak.
Simulate
Click on the button to recalculate the isotopic pattern.

2.20 Calculate Molecular Formula

2.20.1 Calculate Molecular Formula

The tool is opened using the command "Tools -> Calculate Molecular Formula" from the Main Window. Use F1 to open this help.

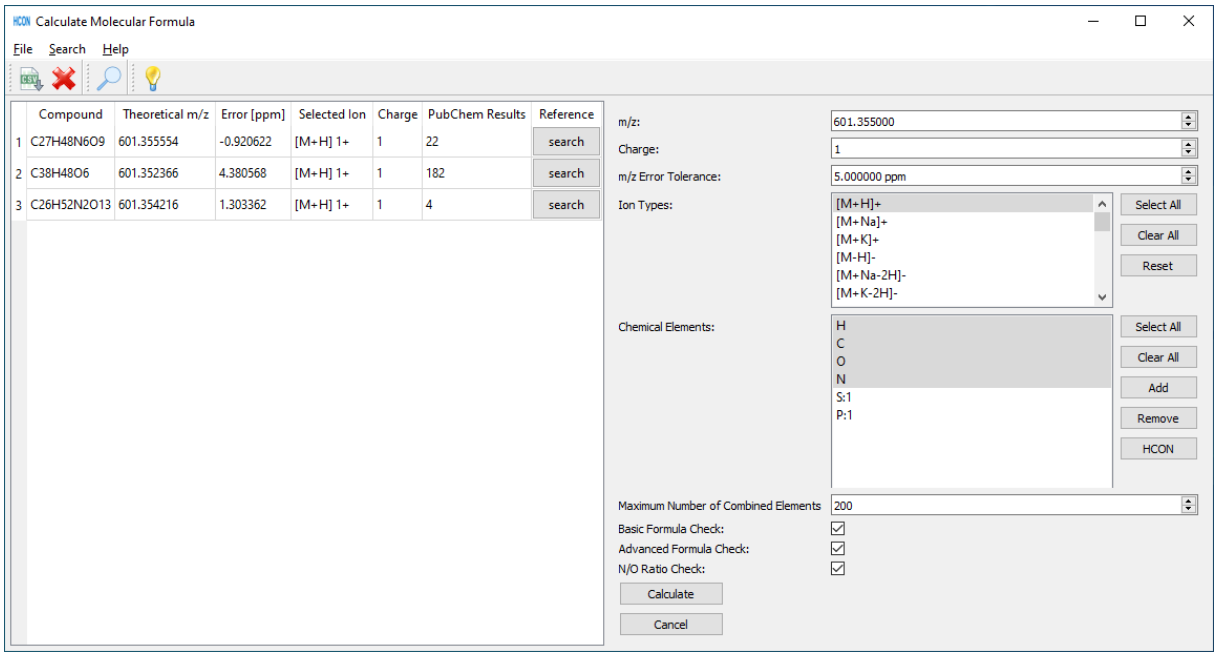


Figure 2.62 Calculate molecular formula of desferrioxamine E.

2.20.1.1 Toolbar

Export Table to CSV (CTRL + E)
Export the table of peaks into a csv (comma separated values) file.

Close (Esc)
Close window.

Search PubChem... (CTRL + P)
Search molecular formulas of compounds in PubChem.

HTML Documentation (F1)
Show HTML documentation.
m/z
Enter an experimental m/z value.
Charge
Enter a maximum value of charge. If charge is 2, singly and doubly charged ions are generated and tested. Use negative values if negative ions are selected (e.g., -1).
m/z Error Tolerance
Enter m/z error tolerance value.

Ion Types
Select the types of ions to be calculated. The custom ion types can be defined in the Preferences dialog.
Buttons:

- "Select All" - select all ions in the list.
- "Clear All" - unselect all ions in the list.
- "Reset" - select [M+H]+ ion if defined.

Chemical Elements
Define and select the chemical elements which are used for prediction of compounds. Limit the maximum number of occurrences of an element using a colon (e.g., S:2 means that each formula will contain at the most two atoms of sulfur).
Buttons:

- "Select All" - select all elements in the list.
- "Clear All" - unselect all elements in the list.
- "Add" - add a new item at the end of the list.
- "Remove" - remove selected items from the list.
- "HCON" - clear the list and set the default chemical elements H, C, O, N, S, and P.

Maximum Number of Combined Elements
Enter the maximum number of elements in a formula (M).
Basic Formula Check
Apply Senior's filtering rules. See https://doi.org/10.1186/1471-2105-8-105 for more details.
Advanced Formula Check
Advanced filtering rules are applied if molecular formulas of compounds are generated. See the tables 1, 2, and 3 in https://doi.org/10.1186/1471-2105-8-105 for more details.
N/O Ratio Check
Check if the number of nitrogen atoms is less or equal to the number of oxygen atoms (Advanced Formula Check must be enabled).
Calculate
Click on the button to calculate molecular formulas corresponding to the input m/z value.
Cancel
Click on the button to cancel the current calculation process.

2.21 Text Filter

2.21.1 Text Filter



Figure 2.63 Text filter.

The text filter is embedded in the Main Window, Summary Table of Matched Peaks, Building Blocks Editor, Sequence/Compound Database Editor, and Modifications Editor.
Dropdown menu - column 1
Select the name of column 1 to be searched.
Dropdown menu - comparator 1
Select the type of comparator for the column number 1. The value in a selected column can be =, <, <=, >, >= than a value entered in the input line.
Input Line - Text to Find 1 (CTRL + F)
Enter the text to be found.
Arrows 1 (< and >)

Browse identified compounds (Summary Table of Matched Peaks only).
Dropdown menu - logical operator
Select the logical operator OR/AND for the conditions 1 and 2.
Dropdown menu - column 2
Select the name of column 2 to be searched.
Dropdown menu - comparator 2
Select the type of comparator for the column number 2. The value in a selected column can be =, <, <=, >, >= than a value entered in the input line.
Input Line - Text to Find 2
Enter the text to be found into the input line.
Arrows 2 (< and >)
Browse identified compounds (Summary Table of Matched Peaks only).
Checkbox - Case In/sensitive (CTRL + T)
If the checkbox is checked, the text is searched case sensitively, otherwise it is searched case insensitively.
Checkbox - Whole Words Only (CTRL + W)
If the checkbox is checked, only whole words are searched.
Filter (Enter key, if the input line is focused)
Press the button to select only the rows of the table which meet the defined criteria.
Reset (CTRL + R)
Press the button to reset the input line and to show the entire report.

2.22 Preferences

2.22.1 Preferences Dialog

The preferences dialog is opened using the command "File -> Preferences". Use F1 to open this help.
In this dialog, you can define global preferences of the application, which are stored in the "CycloBranch.ini" file.
The preferences dialog has the following tabs:

- Ion Types
- Data Processing
- Directories
- Bookmarks

2.22.1.1 Ion Types

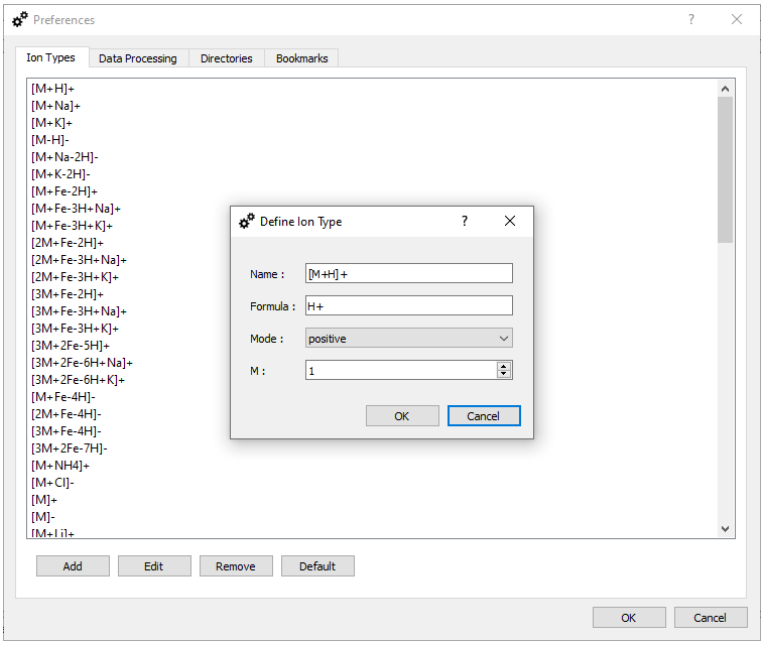


Figure 2.64 Ion Types.

In this tab you can define a list of custom ions, which are shown in the Settings dialog if MS, LC-MS, and MSI data is processed. Buttons:

- 'Add' - add a new entry at the end of the list
- 'Edit' - edit selected item
- 'Remove' - remove selected item
- 'Default' - clear the current list and define a default list of ions (see below)

The dialog 'Define Ion Type' includes the following rows:

- 'Name' - a unique name of ion, e.g., [M+H]+.
- 'Formula' - the formula 'H+' determines the difference between a neutral molecule and the ion. The symbol '+' has a special meaning and it determines that a mass of electron is subtracted. For example, if you would like to define the ion [M-H]-, the correct syntax is 'H-1+-1', where '+-1' means that the mass of electron is added.
- 'Mode' - define mode in which the ion type can be used - positive or negative.
- 'M' - define the number *n* of connected molecules in [*n*M+H]+, e.g., *n*=2 in [2M+Fe-3H+Na]+.

The default list of ion types includes the following items:

- [M+H]+
- [M+Na]+
- [M+K]+
- [M-H]-
- [M+Na-2H]-
- [M+K-2H]-
- [M+Fe-2H]+

- [M+Fe-3H+Na]+
- [M+Fe-3H+K]+
- [2M+Fe-2H]+
- [2M+Fe-3H+Na]+
- [2M+Fe-3H+K]+
- [3M+Fe-2H]+
- [3M+Fe-3H+Na]+
- [3M+Fe-3H+K]+
- [3M+2Fe-5H]+
- [3M+2Fe-6H+Na]+
- [3M+2Fe-6H+K]+
- [M+Fe-4H]-
- [2M+Fe-4H]-
- [3M+Fe-4H]-
- [3M+2Fe-7H]-
- [M+NH4]+
- [M+Cl]-
- [M]+
- [M]-
- [M+Li]+
- [M+Mg-H]+
- [M+Mg-2H+Na]+
- [M+Mg-2H+K]+
- [M+Mg-3H]-
- [M+Al-2H]+
- [M+Al-3H+Na]+
- [M+Al-3H+K]+
- [M+Al-4H]-
- [M+Si-3H]+
- [M+Si-4H+Na]+
- [M+Si-4H+K]+
- [M+Si-5H]-
- [M+Ca-H]+
- [M+Ca-2H+Na]+
- [M+Ca-2H+K]+
- [M+Ca-3H]-

- [M+Sc-2H]+
- [M+Sc-3H+Na]+
- [M+Sc-3H+K]+
- [M+Sc-4H]-
- [M+Cr-2H]+
- [M+Cr-3H+Na]+
- [M+Cr-3H+K]+
- [M+Cr-4H]-
- [M+Mn-H]+
- [M+Mn-2H+Na]+
- [M+Mn-2H+K]+
- [M+Mn-3H]-
- [M+Co-H]+
- [M+Co-2H+Na]+
- [M+Co-2H+K]+
- [M+Co-3H]-
- [M+Ni-H]+
- [M+Ni-2H+Na]+
- [M+Ni-2H+K]+
- [M+Ni-3H]-
- [M+Cu-H]+
- [M+Cu-2H+Na]+
- [M+Cu-2H+K]+
- [M+Cu-3H]-
- [M+Zn-H]+
- [M+Zn-2H+Na]+
- [M+Zn-2H+K]+
- [M+Zn-3H]-
- [M+Ga-2H]+
- [M+Ga-3H+Na]+
- [M+Ga-3H+K]+
- [M+Ga-4H]-

2.22.1.2 Data Processing

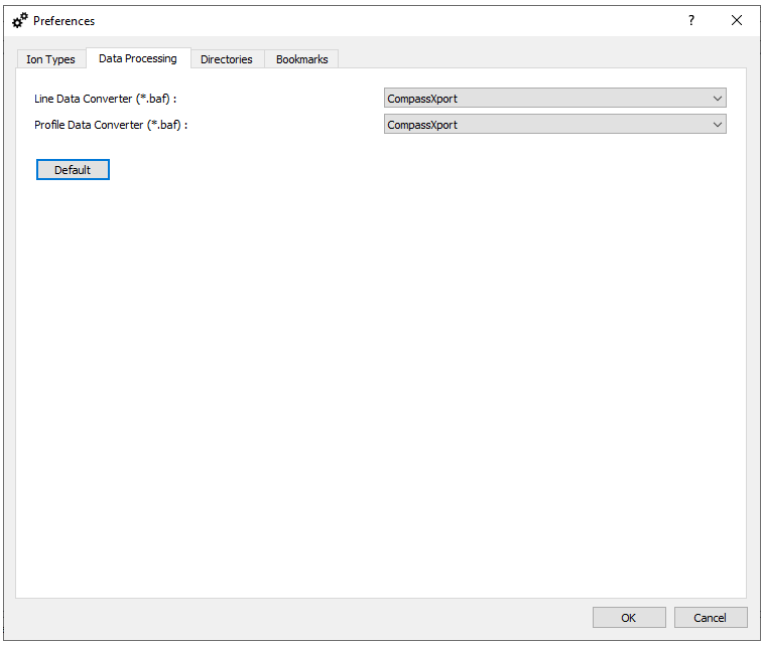


Figure 2.65 Data processing options.

Define a tool used for extraction of line/profile mass spectra from ∗.baf files. The corresponding tool (`CompassXport 3.0.13.1 or CompassXtract 3.2.201 (64-bit)`) must be installed separately.

2.22.1.3 Directories

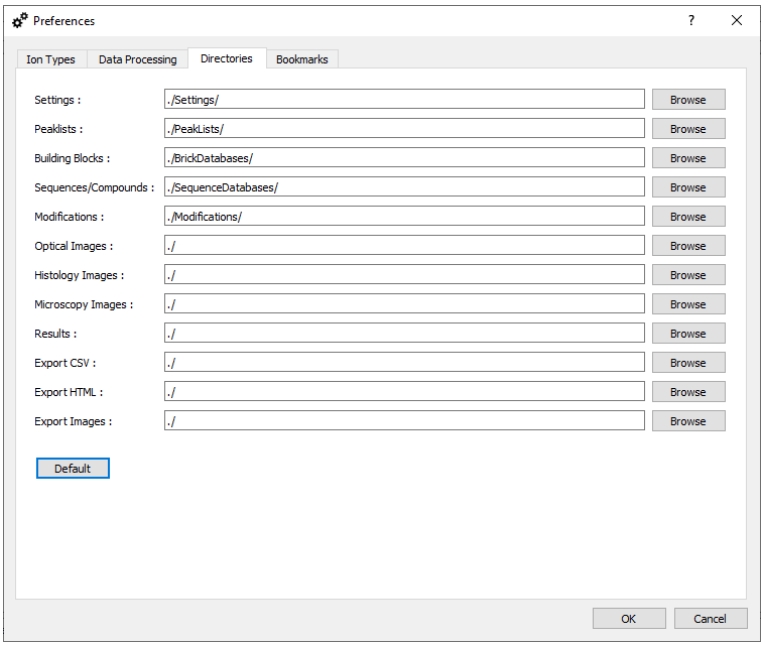


Figure 2.66 Default Directories.

In this tab, you can define the default folders for various types of input/output files.

2.22.1.4 Bookmarks

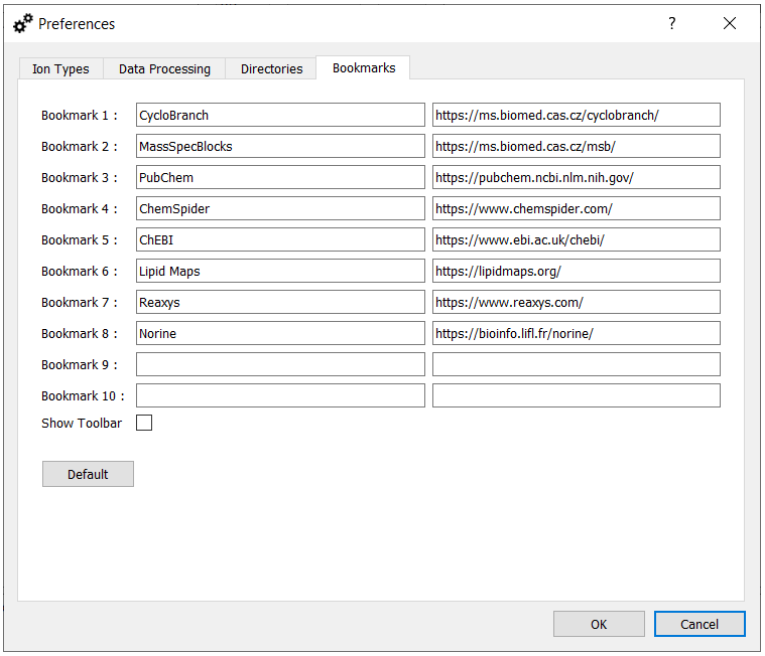


Figure 2.67 Default bookmarks.

A custom list of up to 10 bookmarks can be created. The bookmarks can then be opened in an external web browser using CTRL + 1, CTRL + 2, ..., CTRL + 9, and CTRL + 0. Select "Show Toolbar" to display a toolbar with bookmarks in the main window of the application.

Chapter 3

Tutorials

Tutorials (CycloBranch 2.0.x)

- Tutorial 1: Dereplication in MS1 data
- Tutorial 2: Dereplication in LC-MS data
- Tutorial 3: Identification of unknown compounds in LC-MS data
- Tutorial 4: Dereplication in MSI data and fusion with histology image
- Tutorial 5: Identification of unknown compounds in MSI data
- Tutorial 6: Annotate MS/MS spectrum of a metabolite
- MassSpecBlocks: **Tutorial**, **Homepage**, **Reference and more information**

If you would like to annotate single MS/MS spectra of (non-)ribosomal peptides and siderophores, see the tutorials for CycloBranch (1.x). See also the extra pages for linear, cyclic, branched, and branch-cyclic peptides; the building blocks editor, sequence/compound database editor, modifications editor, and draw peptide tool.

Tutorials (CycloBranch 1.x)

- Tutorial 1: Does a peptide have a cycle ?
- Tutorial 2: How to determine a tag ?
- Tutorial 3: How to determine a complete sequence ?
- Tutorial 4: How to determine a branched sequence ?
- Tutorial 5: How to determine terminal modifications ?
- Tutorial 6: How to compare a theoretical spectrum with a peaklist ?
- Tutorial 7: How to search in a database of MS/MS spectra ?
- Tutorial 8: How to search MS spectrum against a database of compounds ?
- Tutorial 9: Draw Peptide Tool
- Tutorial 10: Identification of Branch-Cyclic Peptide with a Long-sized Branch
- Tutorial 11: Identification of Peptide Family using Similarity Search
- Tutorial 12: Identification of Siderophores in MS/MS data
- Tutorial 13: Identification of Siderophores in MALDI-MS and LC-MS data
- Tutorial 14: Identification of Siderophores in Imaging Mass Spectra

See also a recent protocol on how to configure CycloBranch for different use-cases:

Tomas Pluhacek, Anton Skriba, Jiri Novak, Dominika Luptakova and Vladimir Havlicek.
Analysis of Microbial Siderophores by Mass Spectrometry.
In: Sanjoy K. Bhattacharya (ed.), *Metabolomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1996, pp. 131-153, Springer, 2019. DOI: 10.1007/978-1-4939-9488-5_12.

Chapter 4

Download

If you use CycloBranch in your work, please cite us using

- *Anal. Chem.* (2020), 92(10):6844-6849. DOI: 10.1021/acs.analchem.0c00170
  Download citation: [**ris**], [**bib**]

or another related reference.

21.8.2023

CycloBranch 2.1.35

- Changes:
  - The annotation of compounds with metal complexes has been improved.
  - The performance of Compound Search has been improved.
  - Annotated Features Overview - retention time is now listed for LC-MS data, the width of columns can be modified.
  - An average spectrum can now be calculated for LC-MS data in Chromatogram Window and the peaks can be annotated.
  - The ids of groups were replaced by names in Define Samples and Calibration Curve dialog.
  - The concentrations of compounds can now be visualized in the dependency on the sample collection time.
  - Some other minor bugs and performance issues were fixed and minor improvements were made.
- Download (Windows, 64 bit - tested on Windows 10)
- Download (Linux, 64 bit - tested on Ubuntu 20.04)
- Download (macOS, 64 bit - tested on macOS 10.12 Sierra)
- PDF manual
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids

10.8.2022

CycloBranch 2.1.21

- Changes:
  - Quantification of compounds in LC-MS datasets is now possible.
  - "Annotated Features Overview" added (a table of all features matched in all LC-MS files).
  - "Simulate Isotopic Pattern" feature added.
  - "Calculate Molecular Formula" feature added.

  - The names corresponding to molecular formulas of compounds identified in "Compound Search - MS, LC-MS, MSI" mode can now be searched in PubChem.
  - The range of retention time can now be limited in the settings dialog if LC-MS data is processed.
  - Retention time of individual database compounds can also be limited if LC-MS data is processed.
  - "Minimum Intensity of Highest Peak in Isotopic Pattern" parameter was added into the Settings dialog.
  - "54Fe/56Fe Ratio" parameter was added into the Settings dialog.
  - Line/Profile mass spectra can now optionally be converted by CompassXtract from baf files.
  - Waters data support improved.

  - EIC chromatogram is now visible by default.
  - Calculation of FDRs (LC-MS and MSI data) can optionally be disabled.
  - Copy to clipboard function (CTRL+C) improved in Building Block, Sequence/Compound, and Modification editors.
  - Remove duplicate rows feature added in Sequences/Compounds Editor.
  - Export matched formulas to a compound database - feature available in "Summary Table of Matched Peaks" and "Annotated Features Overview" if "Compound Search - MS, LC-MS, MSI" mode was used.
  - Results files (∗.res) are now backwards compatible (starting from version 2.0.8).
  - Data processing tab added in Preferences dialog.
  - Compounds in NP Atlas and COCONUT can now be referenced from Sequence/Compound Editor.
  - Bookmarks toolbar was added and corresponding tab was fixed in Preferences dialog.
  - Unused parameters are now automatically hidden in the Settings dialog.
  - Multiple LC-MS files can now be processed in a batch.
  - Multiple threads can now be used if LC-MS or MSI data are processed.
  - Image fusion improved:
    * Current configuration of layers can now be loaded/saved.
    * Open image function was improved.
    * Import dialog was added to open a batch of microscopy images FEI (SEM) / Leica (optical and fluorescence).
    * Optical and fluorescence microscopy images can now be opened directly from ∗.lif files (Leica).
    * Horizontal and vertical flip feature added for microscopy images.
    * Name and magnification is now visible for every microscopy layer.
    * Navigation buttons were added to browse microscopy layers (Previous, Next).
    * Maximum number of microscopy layers was increased to 200.
    * A dropdown menu was added to change the navigation layer of any microscopy layer.
    * Keep aspect ratio feature added when resizing a region (CTRL + resize by mouse).
    * Clear all/individual layers features added.
  - Some other minor bugs and performance issues were fixed and minor improvements were made.

- Download (Windows, 64 bit - tested on Windows 10)
- Download (Linux, 64 bit - tested on Ubuntu 20.04)
- Download (macOS, 64 bit - tested on macOS 10.12 Sierra)
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids

11.3.2020
CycloBranch 2.0.8

- Changes:
  - Custom ions are now supported for MS, LC-MS, and MSI data.
  - Global preferences dialog added (custom ions, default directories, default URLs).
  - Chromatogram window improved - names/molecular formulas of matched compounds are now reported.
  - Support of fine isotopic patterns of metals improved.
  - New scoring function added - weighted ratio of matched peaks.
  - m/z values of unmatched theoretical peaks in single spectrum detail window can now be focused in a spectrum using a double-click.
  - Bug fixed: intensities in chromatogram window and image fusion window were summed if the same experimental peak was matched by multiple compounds.
  - A bug was fixed when peaks of scrambled ions were generated.
  - Link to PubChem was fixed.

- Download (Windows, 64 bit - tested on Windows 7/10) (updated 29.4.2020)
- Download (Linux, 64 bit - tested on Ubuntu 16.04)
- Download (macOS, 64 bit - tested on macOS 10.12 Sierra)
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids

20.11.2019
CycloBranch 2.0.0

- Changes:
  - Compound search mode added (a database-free search mode for MS1, LC-MS, and MSI data).
  - Fragmentation of metabolites is now supported (peptide type "Other" in MS/MS mode).
  - A list od neutral losses can now be defined for individual building blocks.
  - The databases of blocks were updated to a new file format (the column neutral losses was added).
  - Spectrum detail window can now be opened direcly from chromatogram or image fusion window by a double-click.
  - Filter options improved - main window, summary table of matched peaks, and editors (building blocks, sequences/compounds, and modifications).
  - Processing of Waters data improved and visualization of profile data added.
  - Visualization of profile mzML/Thermo raw files added and a bug fixed during profile data processing.
  - Retention time is now reported for LC-MS data.
  - Minimum number of spectra can now be defined for LC-MS and MSI data.
  - Minimum number of matched ions can now be defined for MS, LC-MS, and MSI data.
  - New parameters added: Basic Formula Check, Advanced Formula Check, N/O Ratio Check, Isotope m/z Tolerance, and Isotope Intensity Tolerance.
  - Percentage contributions of peak components (e.g. 13C, 17O) are now reported if theoretical isotopic patterns are calculated.
  - Copy to clipboard function (CTRL+C) improved in spectrum detail window, main window, and summary table of matched peaks (multiple rows and columns can be selected in the table).
  - Visualization of mass spectra fixed (occasional flashing).
  - A bug fixed - existing items in the list of neutral losses were not editable if the settings dialog was re-opened.

  - A bug was fixed when too many combinations of neutral losses were generated.
  - Internal ions can now be generated for branch-cyclic peptides.
  - DOI can now be used as a reference in a compound/sequence database.
  - [M+Cl]- ion added in MS mode.
  - Link to BBDGNC added.
  - zlib compression is now supported in mzML files.
  - Some other bugs were fixed and minor improvements were made.

- Download (Windows, 64 bit - tested on Windows 7/10) (updated 29.4.2020)
- Download (Linux, 64 bit - tested on Ubuntu 16.04)
- Download (macOS, 64 bit - tested on macOS 10.12 Sierra)
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids

22.5.2018

CycloBranch 1.3.1

- Changes:
  - Spectrum detail window improved:
    - Profile spectra can now be visualized.
      - Bruker's baf files on (Windows only)
      - imzML
    - Visualization of mass spectra improved.
    - Hide labels of peaks feature added.
    - Mouse wheel function improved.
    - Summary formulas of experimental peaks are now reported also in MS/MS mode.
    - An experimental peak can now be focused in using a double-click.
  - Chromatogram - visualization and mouse wheel function improved.
  - Summary Table of Matched Peaks - an experimental peak can now be focused using a double-click.
  - "Compare Peaklist with Database - MS/MS data" mode - a bug fixed; different theoretical spectra with the same score were reported only once.
  - LC-MS/MS data - all spectra in a dataset are now compared with a theoretical spectrum of a searched sequence if "Compare Peaklist(s) with Spectrum of Searched Sequence" mode is used (the parameter "Scan no." is not used in this mode).
  - Settings dialog - layout fixed for macOS.
  - FileConverter - collision of filenames between OpenMS and Xcalibur fixed.

- Download (Windows, 64 bit - tested on Windows 7/10)
- Download (Linux, 64 bit - tested on Ubuntu 16.04)
- Download (macOS, 64 bit - tested on macOS 10.12 Sierra)
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids

26.2.2018
CycloBranch 1.2.748

- Changes:
  - A custom list of neutral losses can now be defined for MS, MSI and MS/MS data.
  - Details of MS/MS spectra improved - matched peaks can now be filtered using a name of a ion series or using name of a neutral loss.
  - Settings dialog improved:
    - "Neutral Losses" - a modifiable list added.
    - "Maximum Number of Combined Losses" - settings added.
    - "Report Unmatched Theoretical Peaks" - checkbox added; performance improved if unchecked.
    - "m/z Error Tolerance for Deisotoping" - obsolete feature removed.
    - "Remove Hits of Fragments without Hits of Parent Fragments" - obsolete feature removed.
    - Obsolete scoring functions removed.
  - Unmatched theoretical isotope patterns can now be reported in MS, MSI and MS/MS mode.
  - Memory performance was improved.
  - OpenMS 2.3 is now supported.
  - Thermo raw files are now supported.
  - mzML support improved.
  - [M]+, [M]- ions added.
  - Bug fixed: the number of electrons was wrong if the m/z ratio of negative ions was calculated in "↩ Generate Full Isotope Patterns" mode.
- Download (Windows, 64 bit - tested on Windows 7/10)
- Download (Linux, 64 bit - tested on Ubuntu 16.04)
- Download (macOS, 64 bit - tested on macOS 10.12 Sierra)
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids

15.12.2017
CycloBranch 1.2.660

- Changes:
  - *De novo* sequencing improved - sequence candidates with the same score and identical path in the *de novo* graph are now reported only once.
  - Summary formulas of fragment ions are now reported.
  - Fine isotope patterns of fragment ions are now generated.
  - Visualization of mass spectra and chromatogram improved - they can be shifted left/right by mouse now.
  - The list of peaks in a spectrum detail window can now be filtered using a ring break-up point (cyclic peptides) or linearized sequence type (branched peptides).
  - Image Fusion - support of microscopy images fixed; image export fixed; pixel width and height fields added.
  - Summary Table of Matched Peaks - calculation of statistics fixed.
  - Multiply dehydrated and deamidated fragment/precursor ions are now reported.
  - imzML support improved.
  - CompassXPort support improved.

  - [M+Si-3H]+, [M+Si-4H+Na]+, [M+Si-4H+K]+, and [M+Si-5H]- ions added.
  - Some minor bugs were fixed and performance was improved.

- Download (Windows, 64 bit - tested on Windows 7/10)
- Download (Linux, 64 bit - tested on Ubuntu 16.04)
- Download (macOS, 64 bit - tested on macOS 10.12 Sierra)
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids

22.6.2017

CycloBranch 1.2.495

- Changes:
  - MALDI-MSI and DESI-MSI data can now be correlated with histology and SEM images.
  - Waters raw data folders can now be processed (Windows only).
  - [M+Na-2H]-, [M+K-2H]-, and [M+NH4]+ ions added.
  - Single spectra can now be opened also from the Summary Table of Matched Peaks using Enter key or a row double click.
  - Filter options in Main Window, Summary Table of Matched Peaks, Building Blocks Editor, Sequence Database Editor and Modifications Editor improved.
  - Minimum Threshold of Absolute Intensity option added into the Settings Dialog.
  - Minimum Pattern Size option added into the Settings Dialog.
  - Generate Permutations of Building Blocks option removed from Settings Dialog; the feature is now always enabled.
  - Scores and FDRs of matched isotope patterns are computed for MS/MSI data.
  - Summary Table of Matched Peaks - exported statistics improved.
  - Show Isomers feature now affects Main Window, Summary Table of Matched Peaks, Spectra Details, De Novo Graph window, and HTML export.
  - Single spectrum detail window and visualization of single spectra improved.
  - Ppm error is now reported as a positive/negative number.
  - Chromatogram window added.
  - Icon sizes increased.
  - Some bugs were fixed and some other minor improvements were made.

- Download (Windows, 64 bit - tested on Windows 7/10)
- Download (Linux, 64 bit - tested on Ubuntu 16.04)
- Download (macOS, 64 bit - tested on macOS 10.12 Sierra)
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids

3.10.2016

CycloBranch 1.2.192

- Changes:

  - mzML file format support fixed.
  - LC-MS/MS spectra support added - id of a scan can now be entered in the settings dialog.
  - Absolute intensities of peaks are now reported.
  - Summary Table of Matched Peaks improved - statistics of matched ions can now be exported.
  - Image Window improved - number of selected points, color scale, average intensity, median intensity, and searched text are reported.
  - Full isotope patterns of compounds can now be generated (MS and MSI data).
  - Visualization of Waters imzML datasets is now supported.
  - Some bugs and performance issues were fixed and some other minor improvements were made.

- Download (Windows, 64 bit - tested on Windows 7/8/10)
- Download (Linux, 64 bit - tested on Ubuntu 14.04)
- Download (OS X, 64 bit - tested on OS X 10.8 (Mountain Lion))
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids

19.5.2016
CycloBranch 1.1.570

- Changes:
  - New features have been added in the building blocks editor, sequence database editor and modifications editor - create a new database, import a database, un/select all rows.
  - Menubars have been included in the building blocks editor, sequence database editor, modifications editor and summary table of matched peaks.
  - Some bugs and multiple performance issues have been fixed.
  - Some keyboard shortcuts have been fixed.
  - The support of biometals and polyketide siderophores have been improved.
  - Group id is now reported for de novo sequence candidates.
  - Imaging mass spectra and imzML data format are now supported.
- Download (Windows, 64 bit - tested on Windows 7/8/10)
- Download (Linux, 64 bit - tested on Ubuntu 14.04)
- Download (OS X, 64 bit - tested on OS X 10.8 (Mountain Lion))
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)
- Additional databases: Lipids, Metabolites

15.12.2015
CycloBranch 1.1.170

- Changes:
  - Support of siderophores and polyketides improved.
  - Feature added - summary table of matched peaks.
  - Feature added - cancel buttons have been added into some progress dialogs.

  - Bug fix - program crashed from time to time when combinations of building blocks were generated.
  - Bug fix - detection of branch-cyclic NRPs using series D) and H) fixed.

- Download (Windows, 64 bit - tested on Windows 7/8/10)
- Download (Linux, 64 bit - tested on Ubuntu 14.04)
- Download (OS X, 64 bit - tested on OS X 10.8 (Mountain Lion))
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)

21.9.2015
CycloBranch 1.0.1512

- Changes:
  - Peaks in a spectrum detail window are now reported in a table.
  - The table of peaks can be exported into a csv file.
  - Columns containing real numbers are now sorted correctly.
  - HTML export was improved.
  - Peaklist from multiple scans can now be processed in MS mode.
  - Support of identification of NRP siderophores added.
  - Support of linear and cyclic polyketide siderophores added.
  - Support of common biometals added (MS and MS/MS mode).
  - Some keyboard shortcuts were added in the building blocks editor, sequence database editor, modifications editor, draw peptide tool, main window, settings dialog, and de novo graph window.
  - Terminal modifications are now reported in spectra detail windows.
  - Link to Norine added.
  - A compatibility issue with CompassXPort was fixed.
  - Some bugs were fixed and minor improvements were made.
- Download (Windows, 64 bit - tested on Windows 7/8/10)
- Download (Linux, 64 bit - tested on Ubuntu 14.04)
- Download (OS X, 64 bit - tested on OS X 10.8 (Mountain Lion))
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)

15.5.2015
CycloBranch 1.0.1216

- Changes:
  - Export of graphical visualization of spectra and peptides into SVG fixed for Windows.
  - Feature added: show/hide isomers of building blocks in the output report.
  - Bug fixed: HTML report was exported incorrectly when the order of rows was changed.
  - Quit confirmation dialog added.
  - Names of precursor ions of multiply charged ions and precursor ions with adducts fixed for MS/MS spectra.
  - Peptide identification details in the main window can be also opened using Return/Enter key.
  - Image export dialog improved in the spectrum detail window.

  - Find text option added into the spectrum detail window.
  - Similarity search can be performed when a peklist is searched against a database of MS/MS spectra.
  - Monoisotopic masses of sequence candidates are now reported in the main window.
  - Settings dialog was improved for monitors with small resolutions (scrollbars added).
  - Export of PS images dropped in Linux (PDF files were generated instead of PS files; pdf2ps tool is now recommended to generate PS files).
  - OS X is now supported.
  - A text filter of rows was added into the main window, building blocks editor, sequence database editor and modifications editor.
  - Save confirmation dialogs added into editors when closing editors.
  - The internal periodic table has been extended.
  - Some bugs and performance issues fixed.

- Download (Windows, 64 bit - tested on Windows 7/8)
- Download (Linux, 64 bit - tested on Ubuntu 14.04)
- Download (OS X, 64 bit - tested on OS X 10.8 (Mountain Lion))
- Download Sources (Windows)
- Download Sources (Linux)
- Download Sources (OS X)

3.3.2015

CycloBranch 1.0.1106

- Changes:
  - Negatively charged ions are supported.
  - c, x and z ions are supported.
  - Precursor ion adducts are supported.
  - Detection of scrambled peaks improved.
  - Linux is now supported.
  - Modifications editor added.
  - A few building blocks were added which cannot eliminate H2O.
  - Support of branched and branch-cyclic peptides improved.
  - "Draw Peptide Tool" added.
  - Keyboard shortcuts and toolbars added.
  - Export of graphical visualization of spectra and peptides into PDF, PS, PNG and SVG (Linux only) added.
  - Visualization of mass spectra improved.
  - Some bugs and performance issues fixed.

- Download (Windows, 64 bit - tested on Windows 7/8)
- Download (Linux, 64 bit - tested on Ubuntu 14.04)
- Download Sources (Windows)
- Download Sources (Linux)

19.11.2014

CycloBranch 1.0.850

- Changes:

  - Database search added (for MS and MS/MS spectra).
  - Databases of building blocks updated.
  - Building blocks editor added.
  - Sequence database editor added.
  - A number of paths in a de novo graph forming sequence candidates is calculated before the search.
  - The version of the application is checked when a results file (∗.res) is opened.
  - Number of matched scrambled fragment ions is reported when a cyclic peptide is searched.
  - Small bugs fixed.

- Download (Windows, 64 bit)
- Download Sources (Windows)

3.10.2014
CycloBranch 1.0.739

- Changes:
  - Visualization of branch-cyclic peptides added.
  - Visualization of branched peptides fixed.
  - Visualization of mass spectra improved.
  - Summary formulas of peptide sequence candidates added.
- Download (Windows, 64 bit)
- Download Sources (Windows)

3.9.2014
CycloBranch 1.0.696

- Initial release.
- Download (Windows, 64 bit)
- Download Sources (Windows)

Chapter 5

Datasets

5.1 List of Datasets

- Dataset 1: MS/MS of Nonribosomal Peptides
- Dataset 2: MALDI-MS of Siderophores
- Dataset 3: MALDI-MSI and DESI-MSI of Siderophores
- Dataset 4: LC-MS of Siderophores
- Dataset 5: MS/MS of Siderophores
- Dataset 6: Correlation of MALDI-MSI with Histology Image

If you use these datasets in your work, please cite us using

- *Anal. Chem.* (2020), 92(10):6844-6849. DOI: 10.1021/acs.analchem.0c00170
  Download citation: [**ris**], [**bib**]

or another related reference.

5.2 Dataset 1: MS/MS of Nonribosomal Peptides

Peptide type	Peptide	Spectrum
Linear	Gramicidin C	mzXML, mzML, mgf, txt, baf (raw data)
	Substance P	mzXML, mzML, mgf, txt
Cyclic	Beauverolide I	mzXML, mzML, mgf, txt, baf (raw data)
	Roseotoxin A	mzXML, mzML, mgf, txt, baf (raw data)
	Cyclosporin A	mzXML, mzML, mgf, txt, baf (raw data)
	Surfactin C	mzXML, mzML, mgf, txt, baf (raw data)
	Valinomycin	mzXML, mzML, mgf, txt, baf (raw data)
Branched	Linearized pseudacyclin A	mzXML, mzML, mgf, txt
	N-acetyl-ESL(KNFI)DQYG-NH2	mzXML, mzML, mgf, txt, baf (raw data)
Branch-cyclic	Pseudacyclin A	mzXML, mzML, mgf, txt, baf (raw data)
	Pyoverdin Pa A	mzXML, mzML, mgf, txt, baf (raw data)

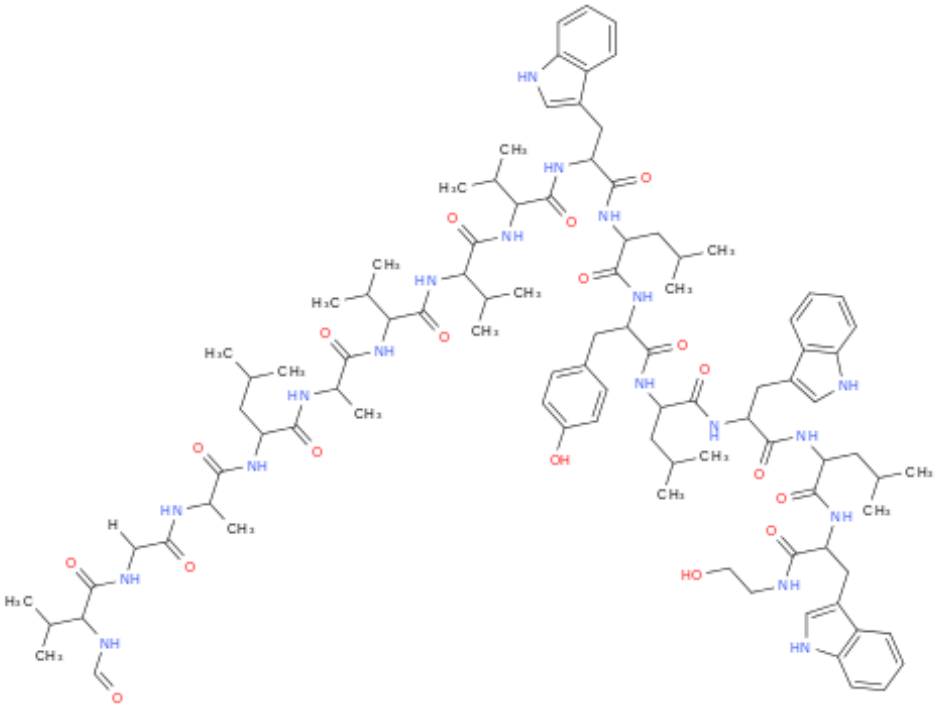


Figure 5.1 Gramicidin C

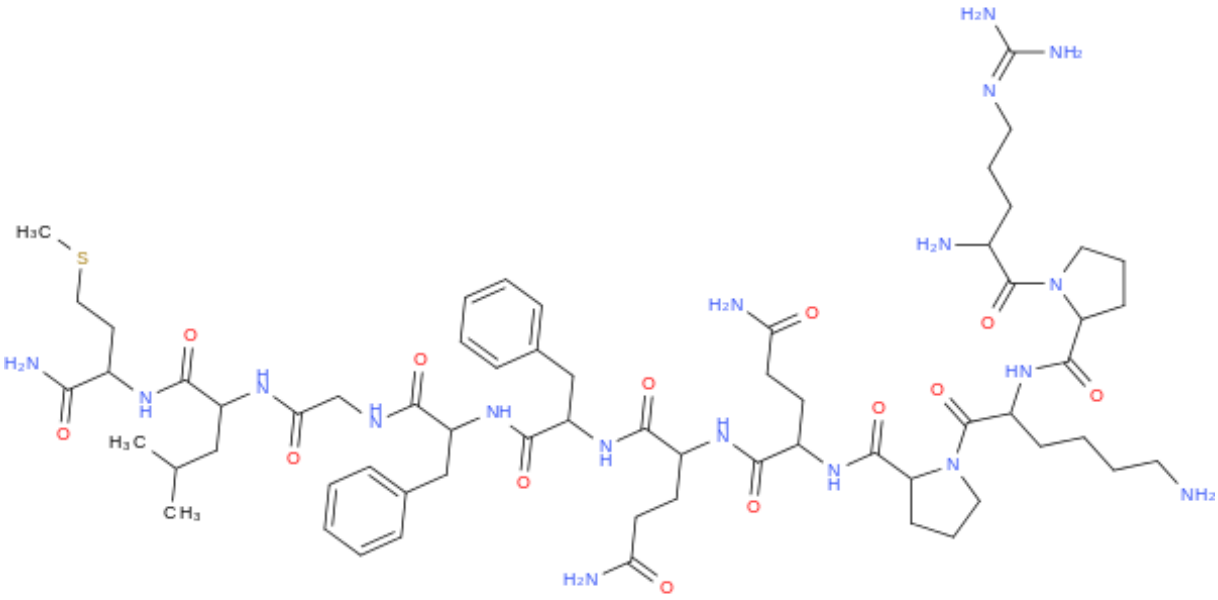


Figure 5.2 Substance P

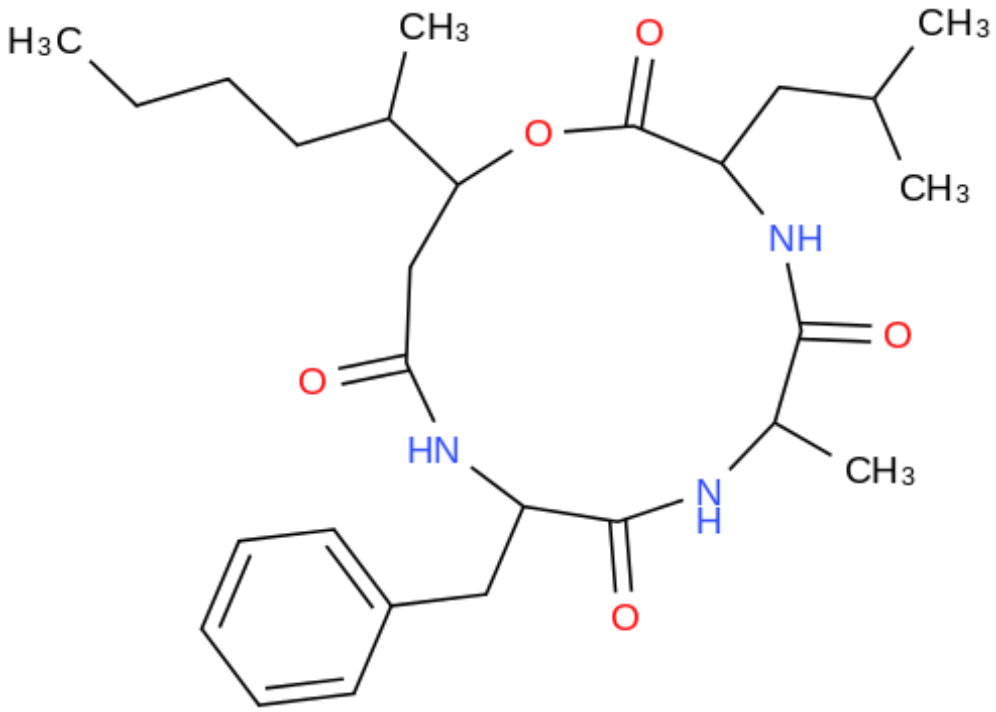


Figure 5.3 Beauverolide I

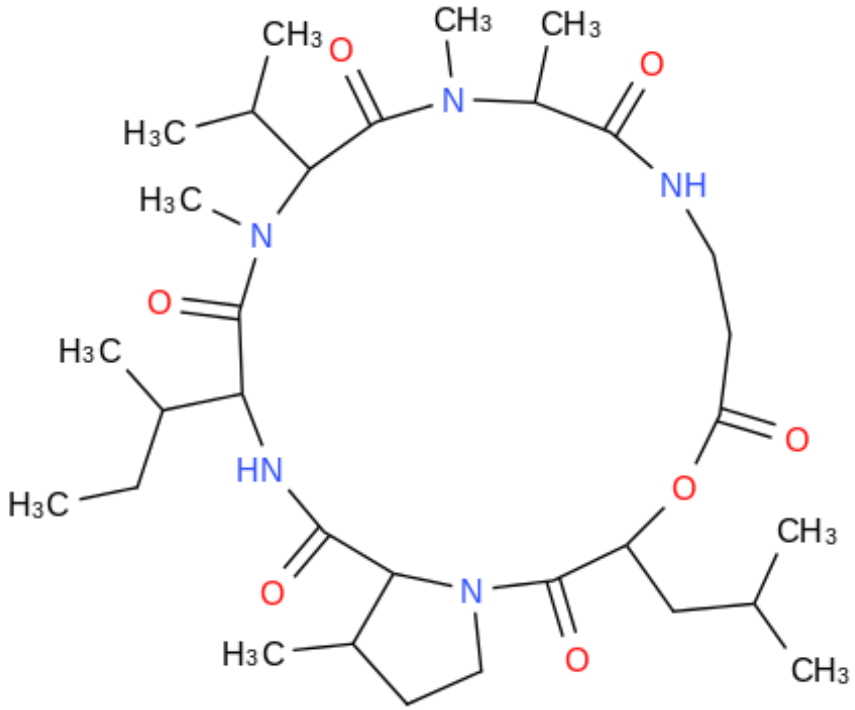


Figure 5.4 Roseotoxin A

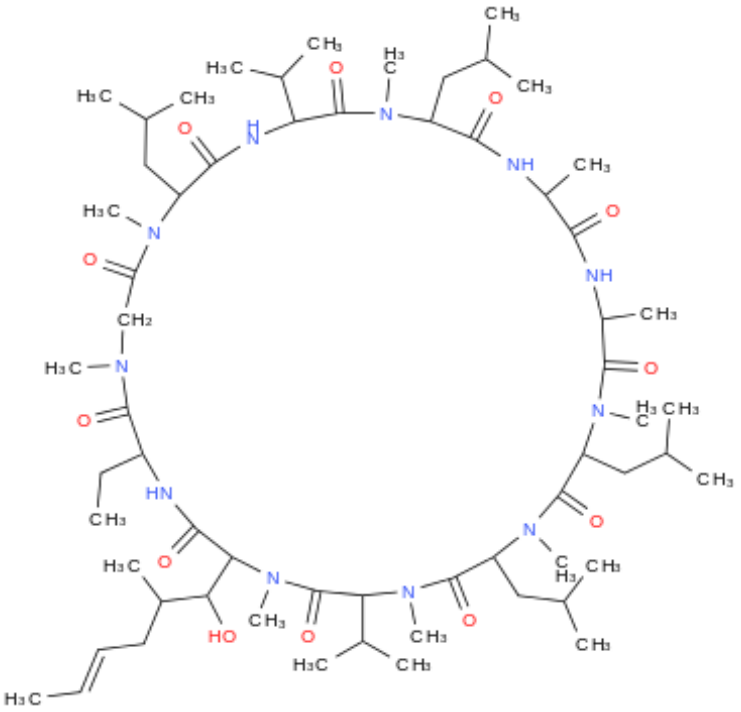


Figure 5.5 Cyclosporin A

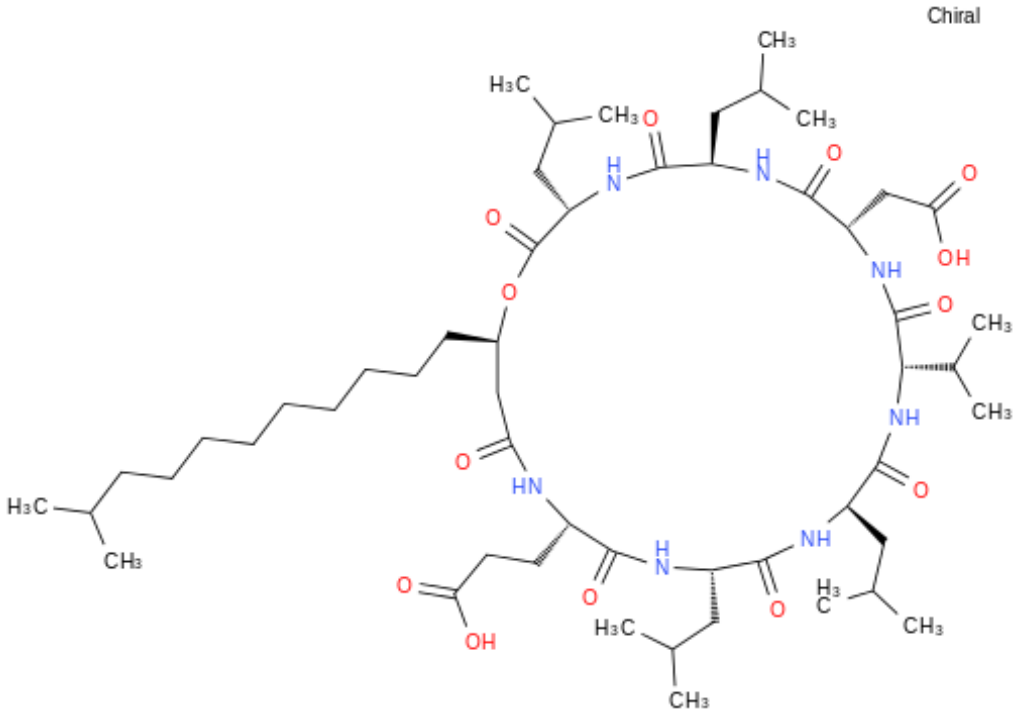


Figure 5.6 Surfactin C

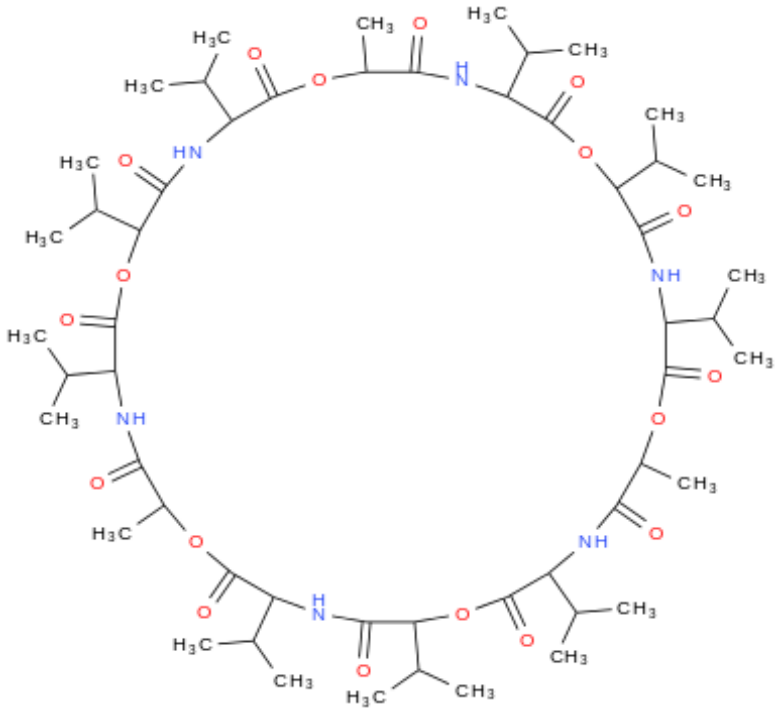


Figure 5.7 Valinomycin

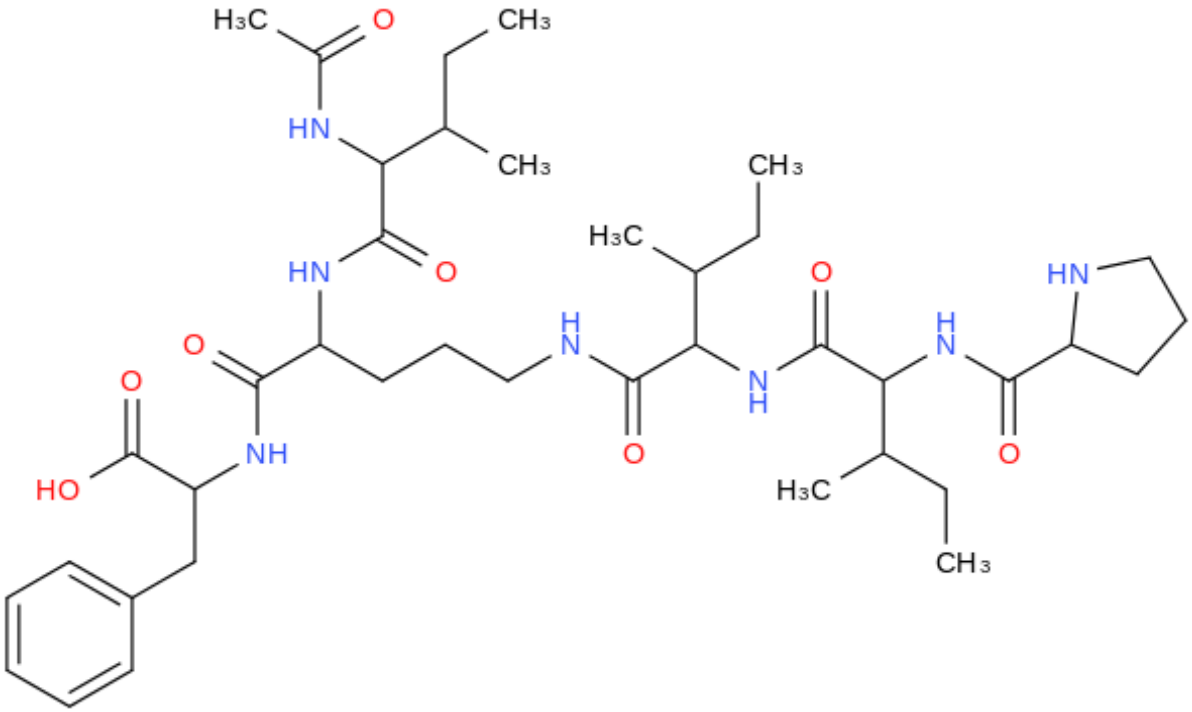


Figure 5.8 Linearized pseudacyclin A

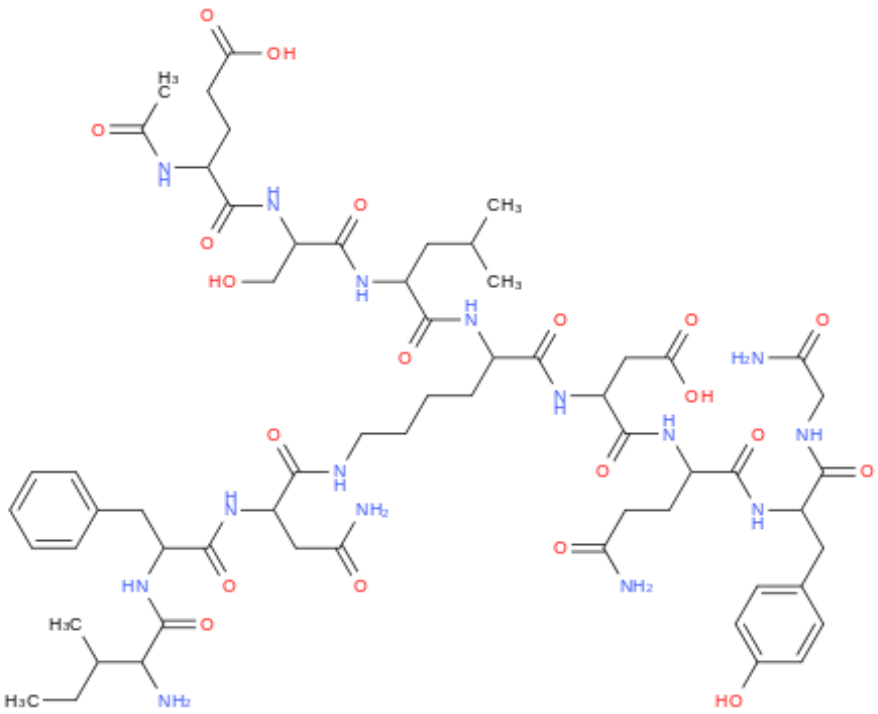


Figure 5.9 N-acetyl-ESL(KNFI)DQYG-NH2

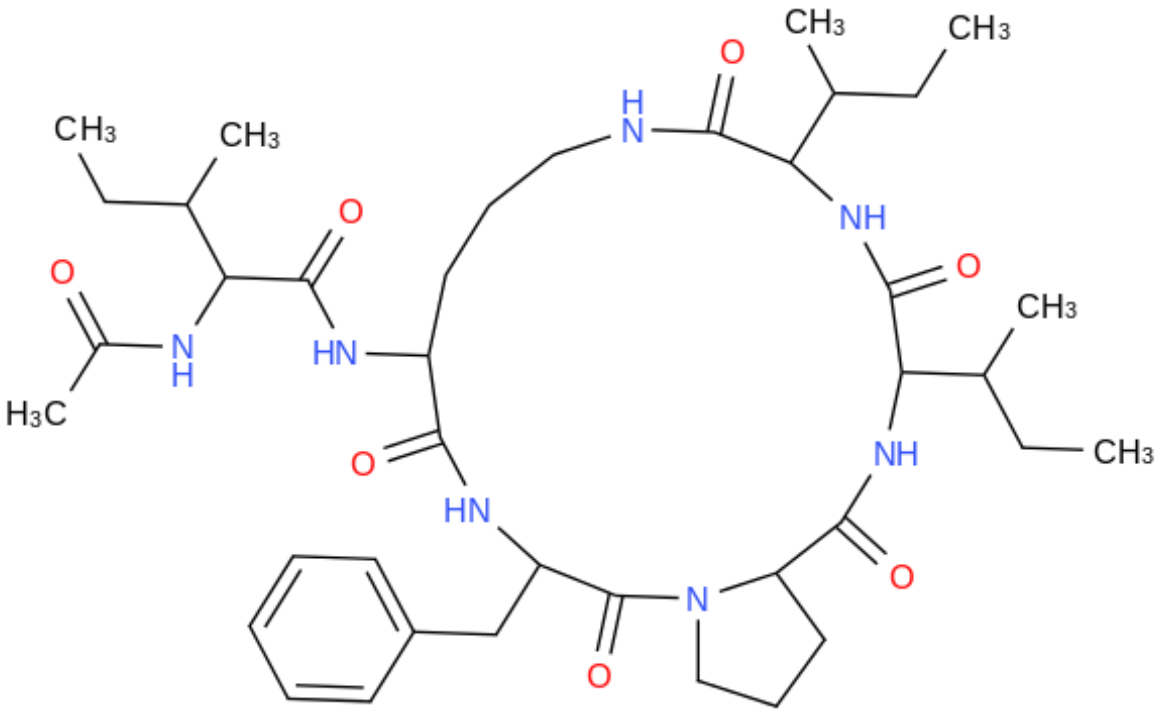


Figure 5.10 Pseudacyclin A

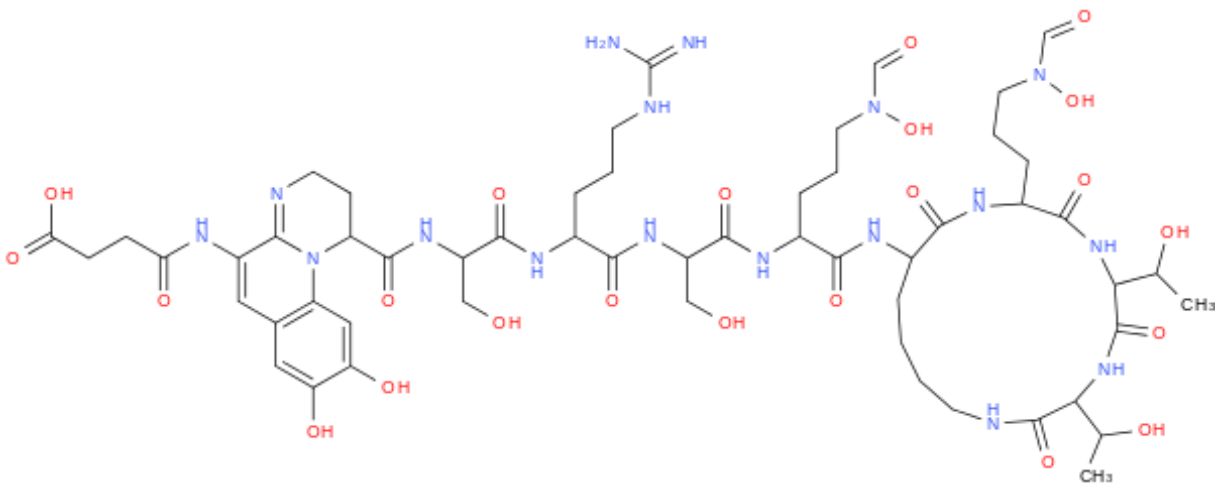


Figure 5.11 Pyoverdin Pa A

The dataset is also available in a public GNPS library.

5.3 Dataset 2: MALDI-MS of Siderophores

Mixture	Spectrum
Ferrichromes	mgf, txt, baf (raw data)
Ferrioxamines	mgf, txt, baf (raw data)
Fusarinines and coprogens	mgf, txt, baf (raw data)
Ornibactins	mgf, txt, baf (raw data)
Pyoverdins	mgf, txt, baf (raw data)
Rhodotulic acid	mgf, txt, baf (raw data)

5.4 Dataset 3: MALDI-MSI and DESI-MSI of Siderophores

Type	Mixture	Spectra
MALDI-MSI (12T SolariX FTICR, Bruker)	01 - ferrichromes diluted 10x; 02 - ferrichromes diluted 100x; 03 - ferrichromes diluted 1000x; 04 - ferrioxamines diluted 10x; 05 - ferrioxamines diluted 100x; 06 - ferrioxamines diluted 1000x; 07 - coprogens/fusarinines diluted 10x; 08 - coprogens/fusarinines diluted 100x; 09 - coprogens/fusarinines diluted 1000x.	imzML + ibd (processed data format; centroided spectra; 64-bit floats; no compression)
DESI-MSI (Synapt G2Si, Waters)	Left - ferrichromes; Middle - ferrioxamines; Right - coprogens/fusarinines.	imzML + ibd (processed data format; profile + centroided spectra; 32-bit floats; no compression)

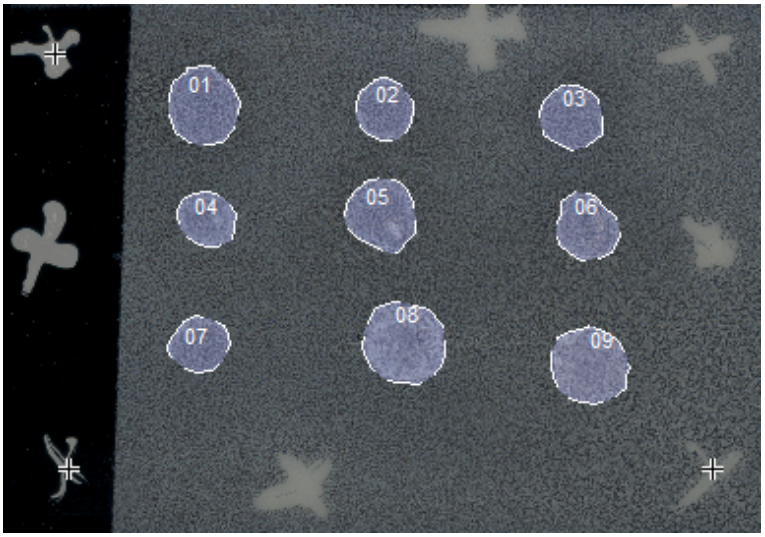


Figure 5.12 MALDI-MSI siderophore dataset - optical image.

Figure 5.13 DESI-MSI siderophore dataset - optical image.

5.5 Dataset 4: LC-MS of Siderophores

Mixture	Spectra
Ferrichromes + ferrioxamines + coprogens and fusarinines	run no. 1: mgf, mzML run no. 2: mgf, mzML

5.6 Dataset 5: MS/MS of Siderophores

Siderophore	Spectrum
Desferri-ferrioxamine B	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-ferrioxamine B	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-ferrioxamine G_1	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-ferrioxamine G_1	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-ferrioxamine E	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-ferrioxamine E	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-ornibactin C_4	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-ornibactin C_6	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-ornibactin C_8	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-coprogen	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-coprogen	mzXML, mzML, mgf, txt, baf (raw data)
Al-coprogen	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-neocoprogen I	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-neocoprogen I	mzXML, mzML, mgf, txt, baf (raw data)
Al-neocoprogen I	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-rhodotulic acid	mzXML, mzML, mgf, txt, baf (raw data)
Al-dimerumic acid	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-TAFC (N',N',N'-triacetylfusarinin C)	mzXML, mzML, mgf, txt, baf (raw data)
Fe-TAFC	mzXML, mzML, mgf, txt, baf (raw data)
Cr-TAFC	mzXML, mzML, mgf, txt, baf (raw data)
Al-TAFC	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-fusarinin C	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-fusarinin C	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-fusarinin B	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-ferrichrome A	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-ferrichrome A	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-ferrirubin	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-ferrirubin	mzXML, mzML, mgf, txt, baf (raw data)
Ferri-ferrichrysin	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-ferricrocin	mzXML, mzML, mgf, txt, baf (raw data)
Desferri-pyoverdin E	mzXML, mzML, mgf, txt, baf (raw data)

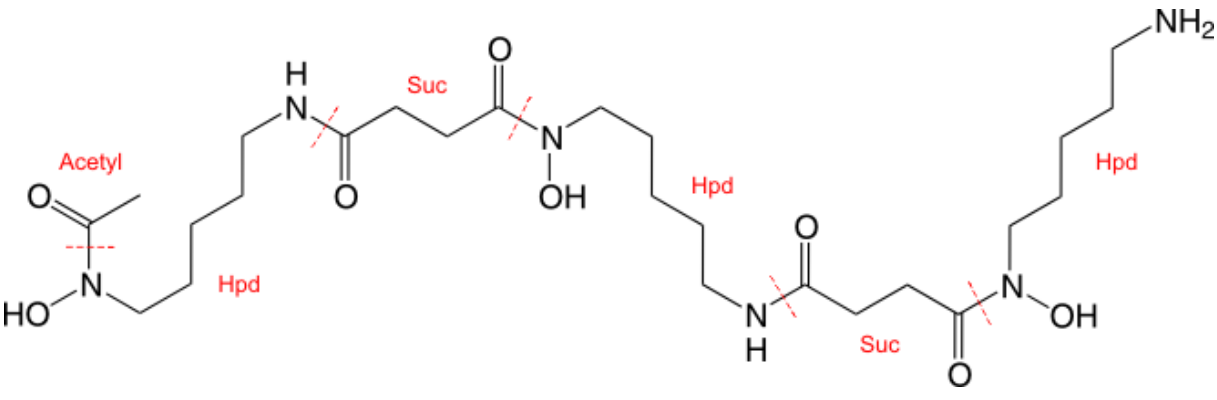


Figure 5.14 Desferri-ferrioxamine B

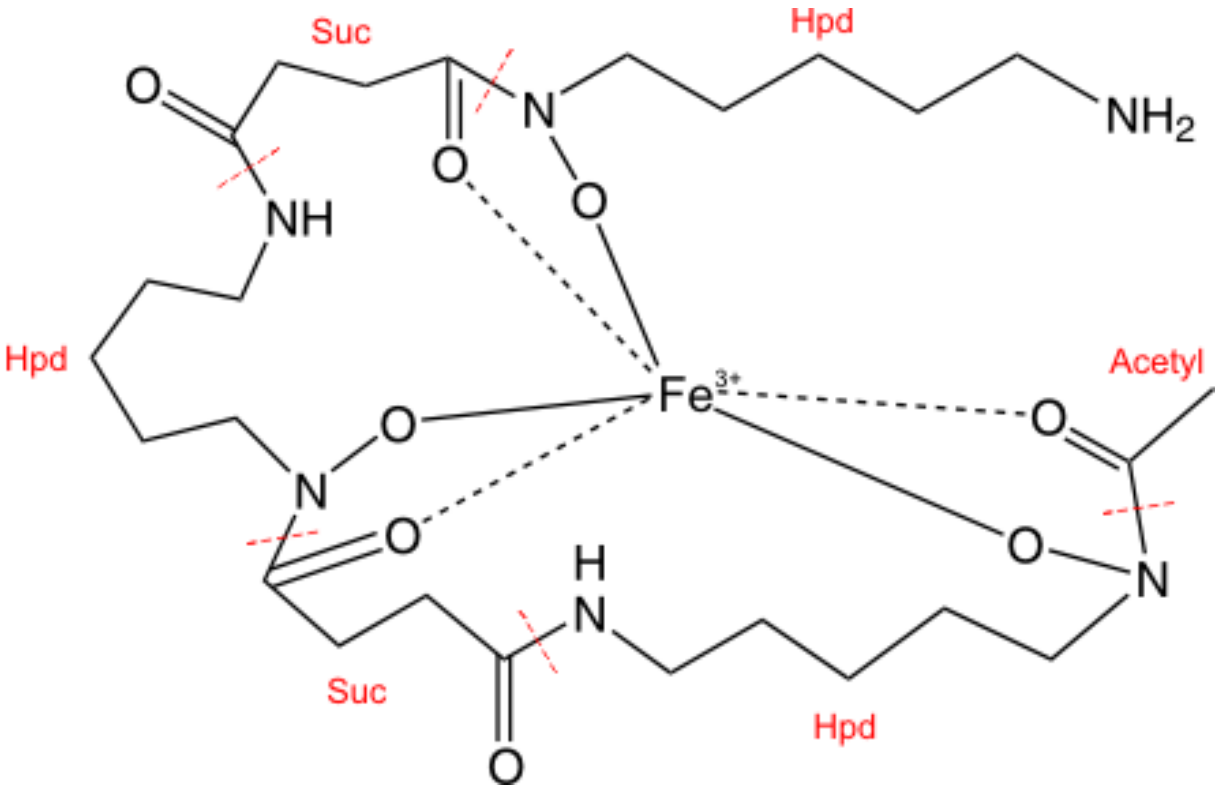


Figure 5.15 Ferri-ferrioxamine B

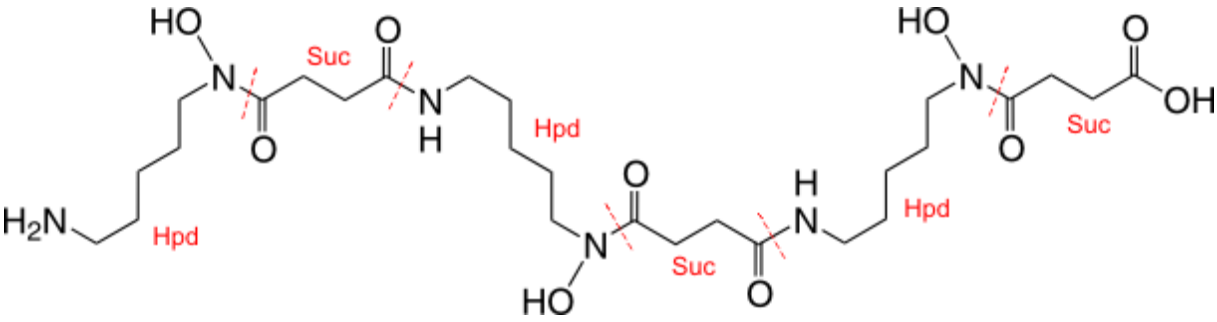


Figure 5.16 Desferri-ferrioxamine G1

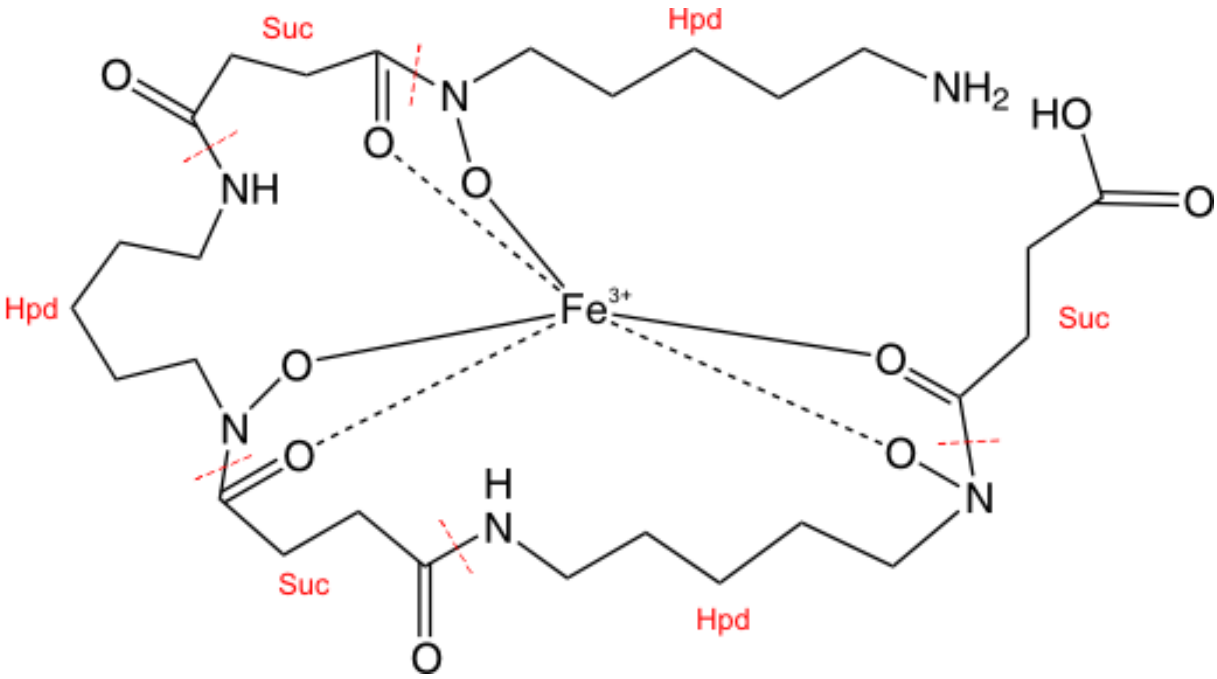


Figure 5.17 Ferri-ferrioxamine G1

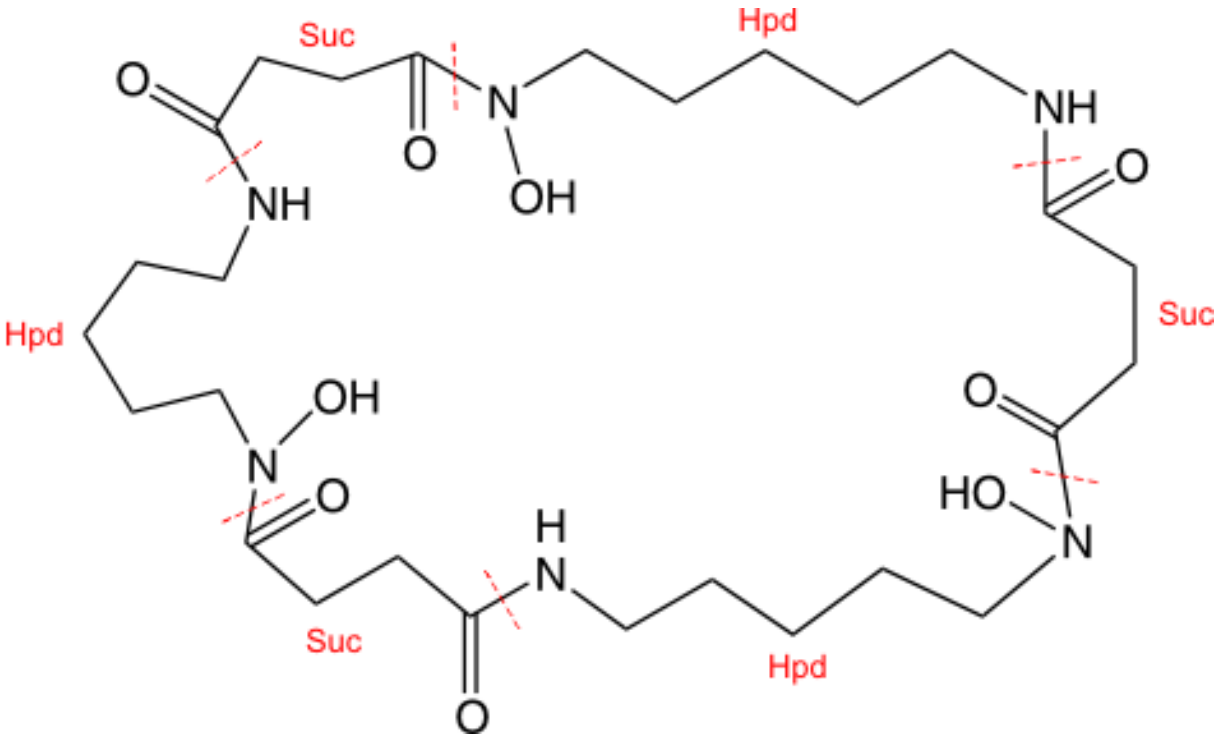


Figure 5.18 Desferri-ferrioxamine E

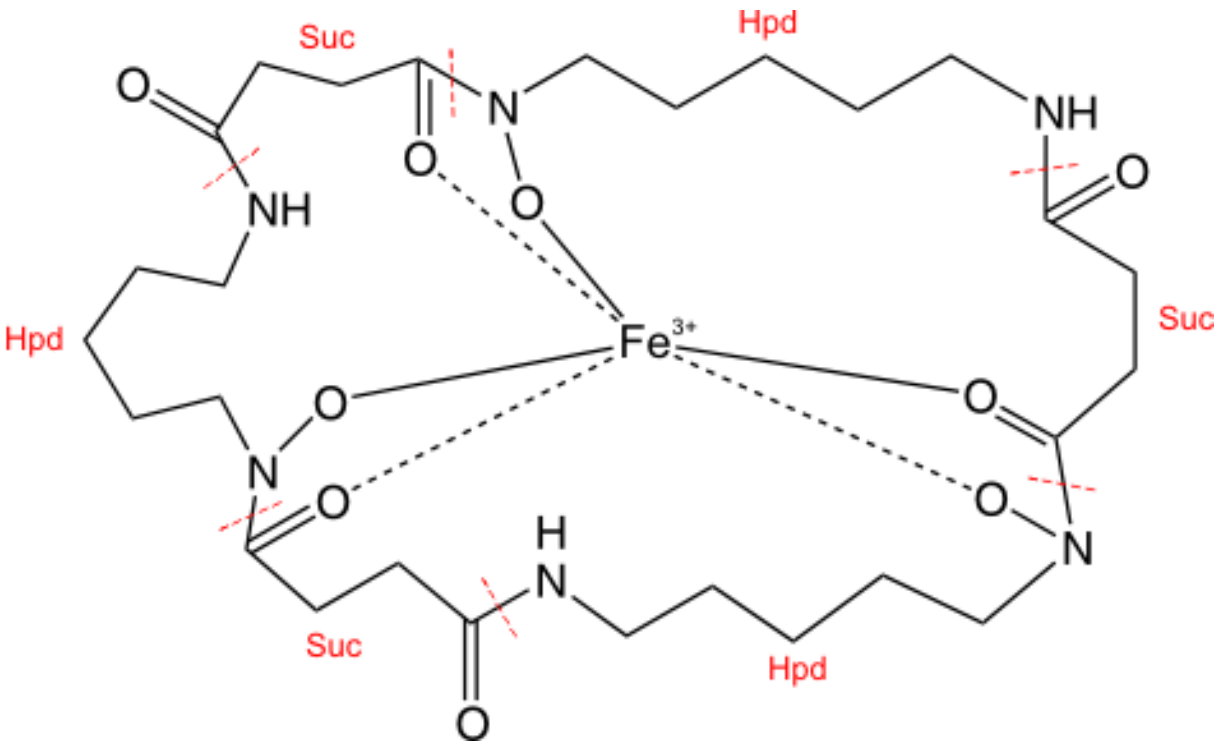


Figure 5.19 Ferri-ferrioxamine E

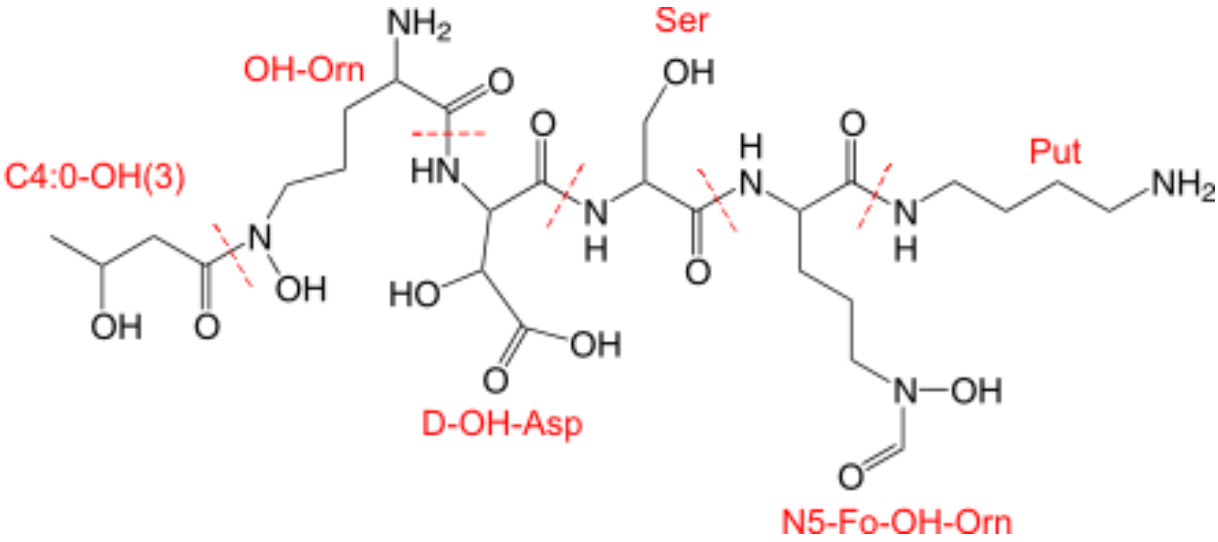


Figure 5.20 Desferri-ornibactin C4

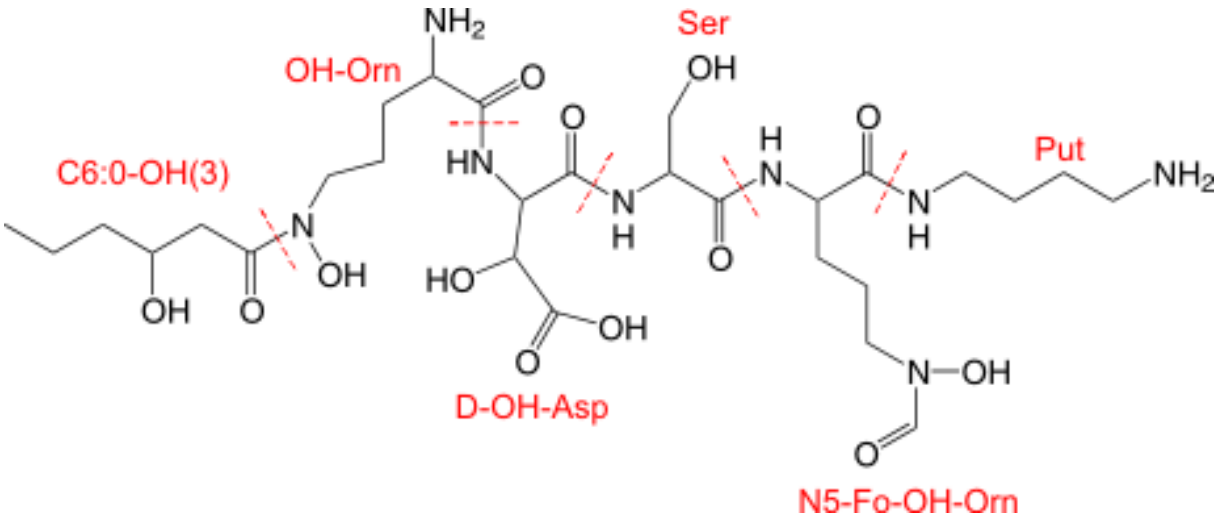


Figure 5.21 Desferri-ornibactin C6

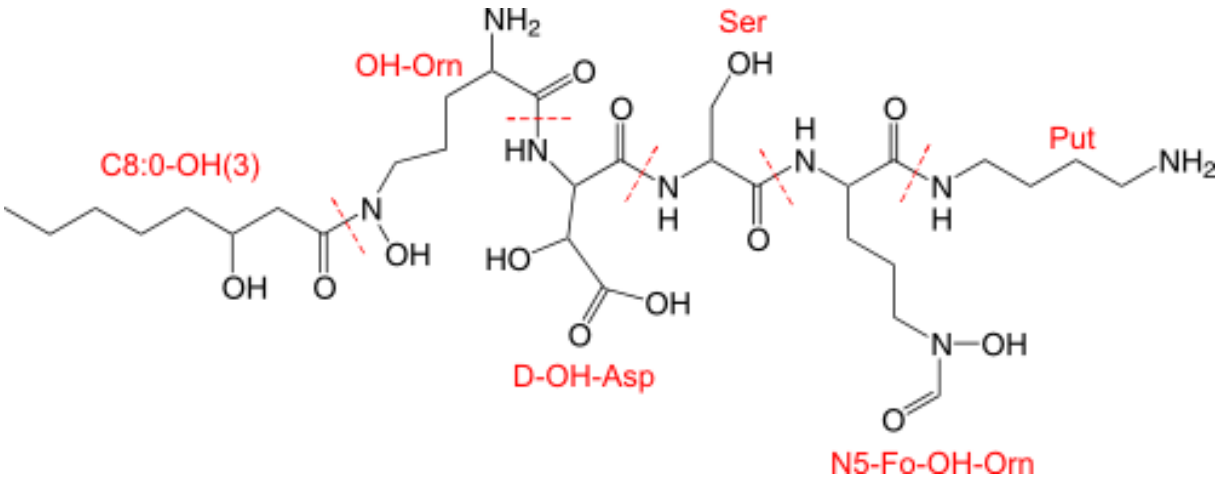


Figure 5.22 Desferri-ornibactin C8

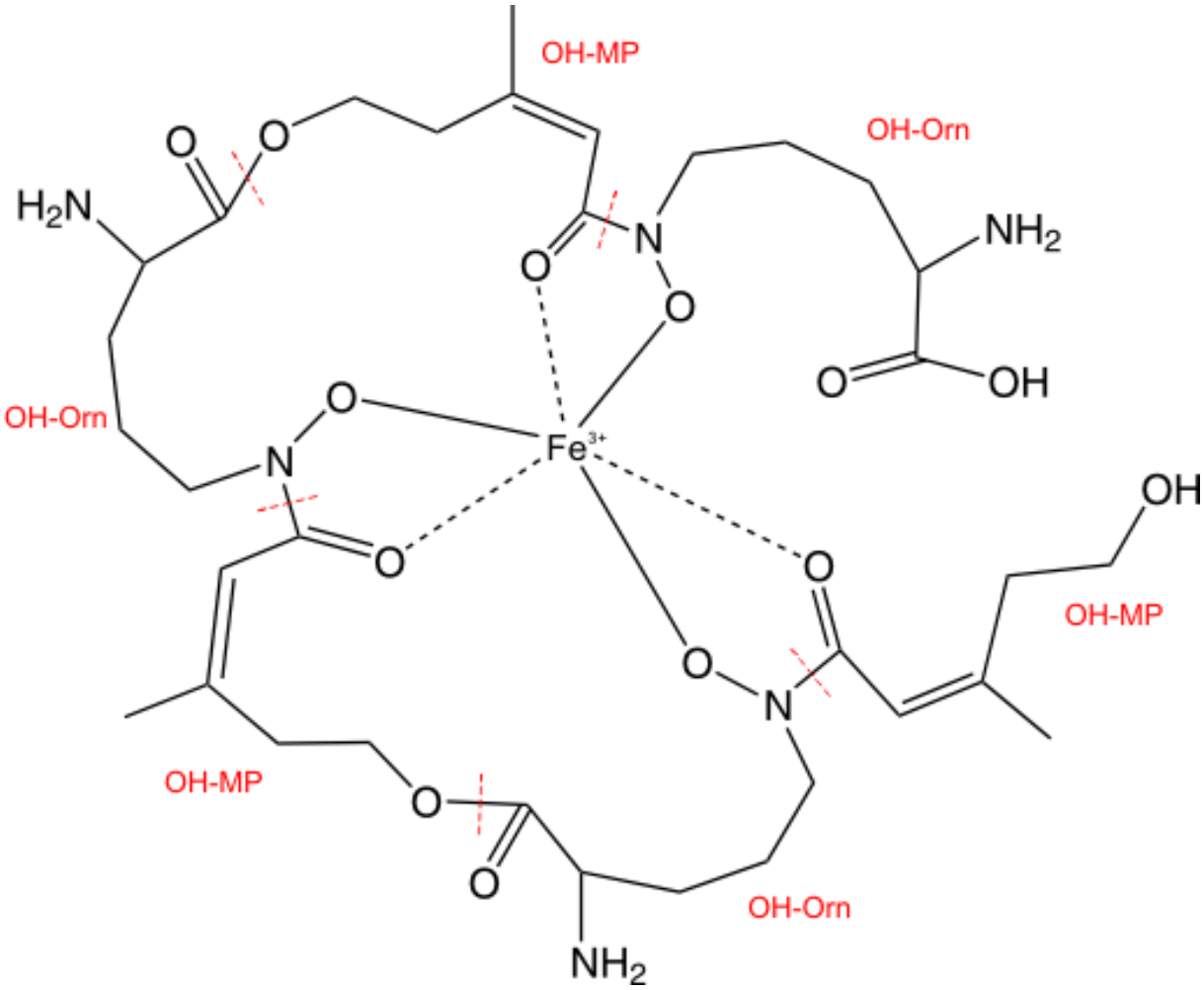


Figure 5.23 Ferri-fusarinin B

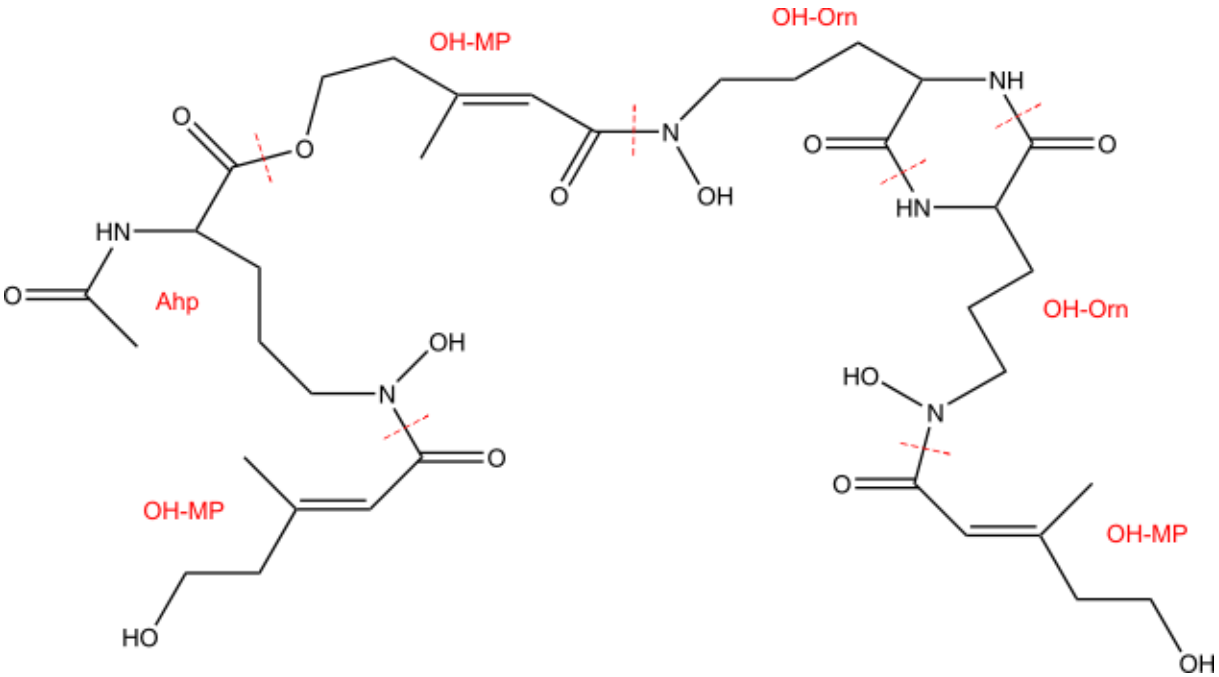


Figure 5.24 Desferri-coprogen

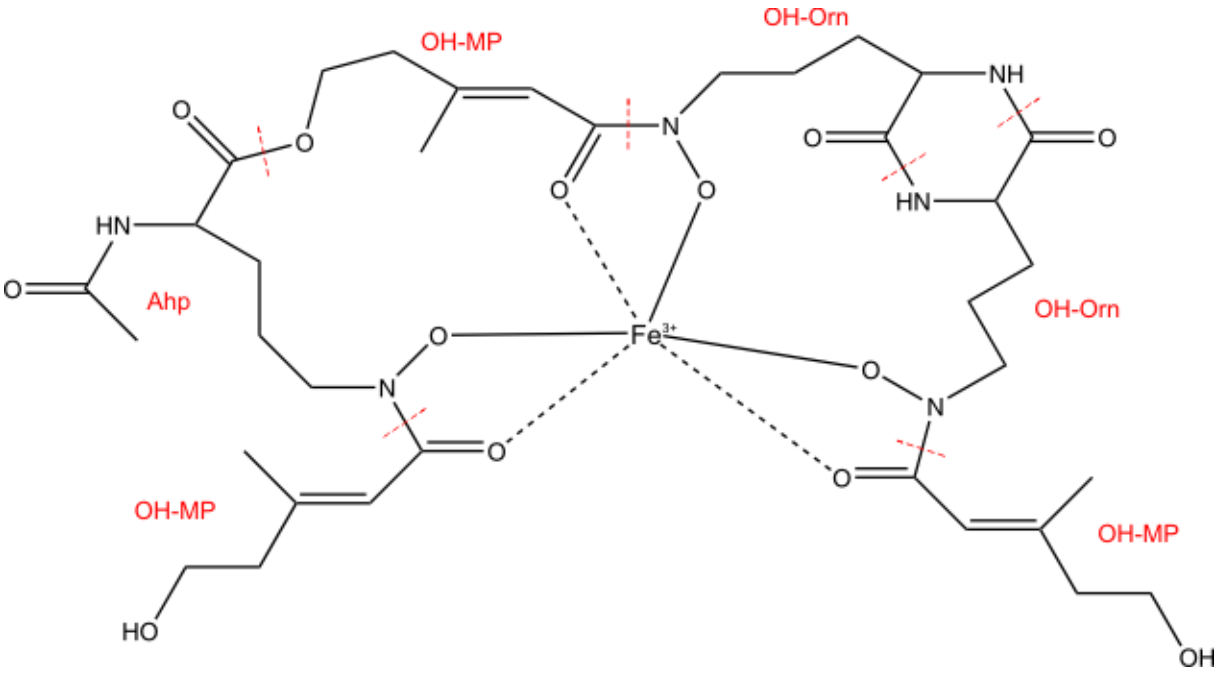


Figure 5.25 Ferri-coprogen

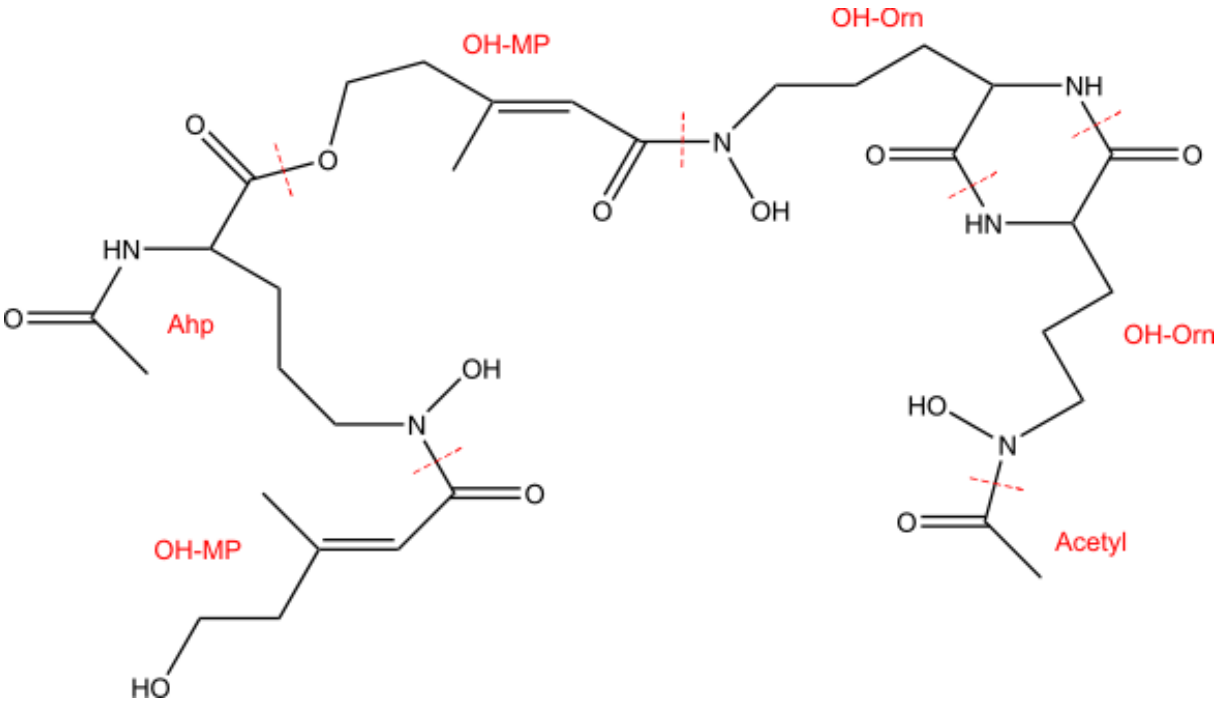


Figure 5.26 Desferri-neocoprogen I

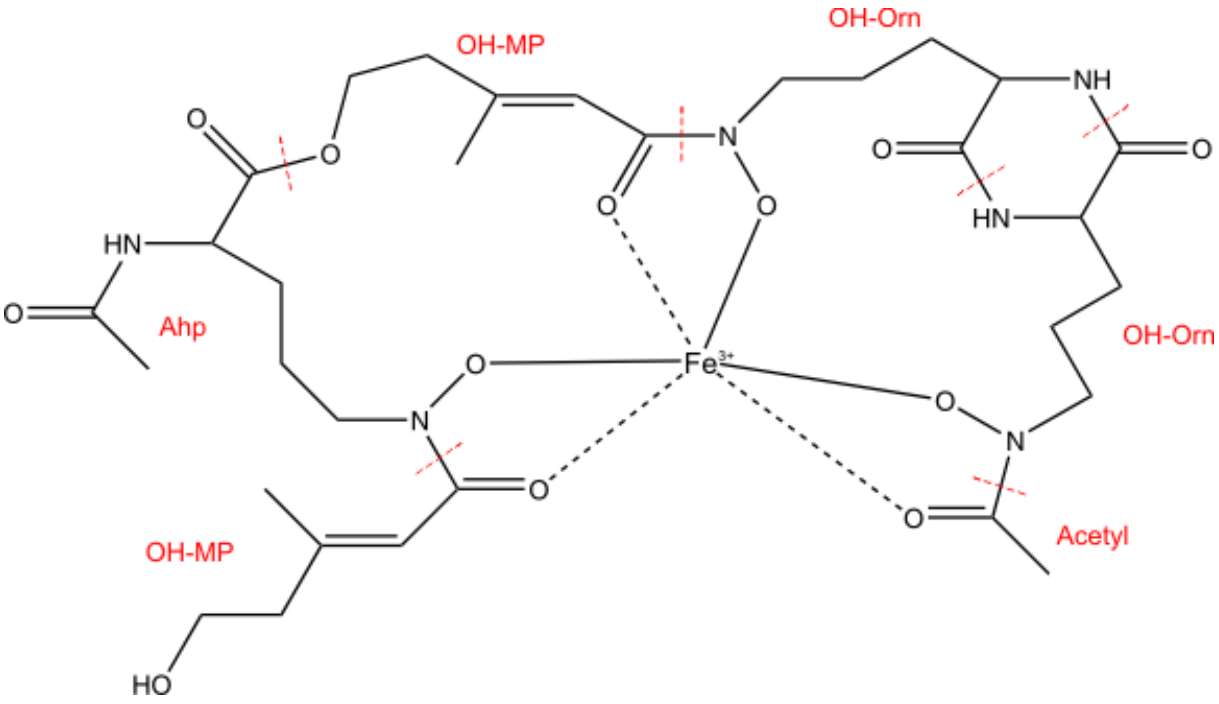


Figure 5.27 Ferri-neocoprogen I

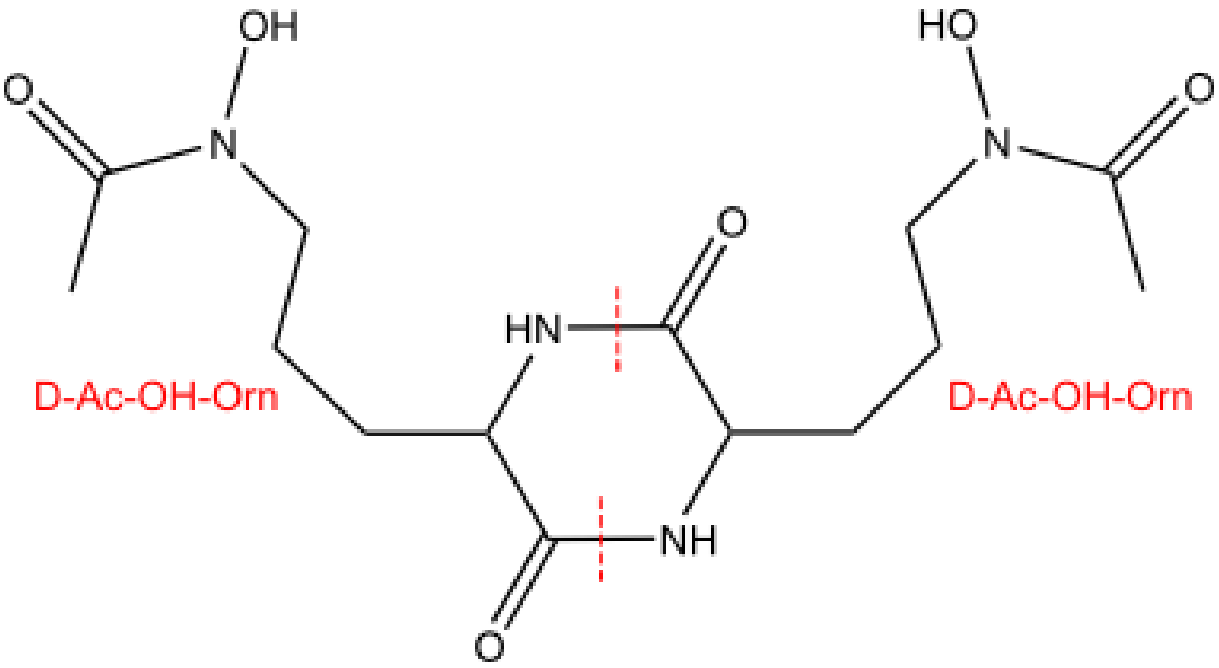


Figure 5.28 Desferri-rhodotulic acid

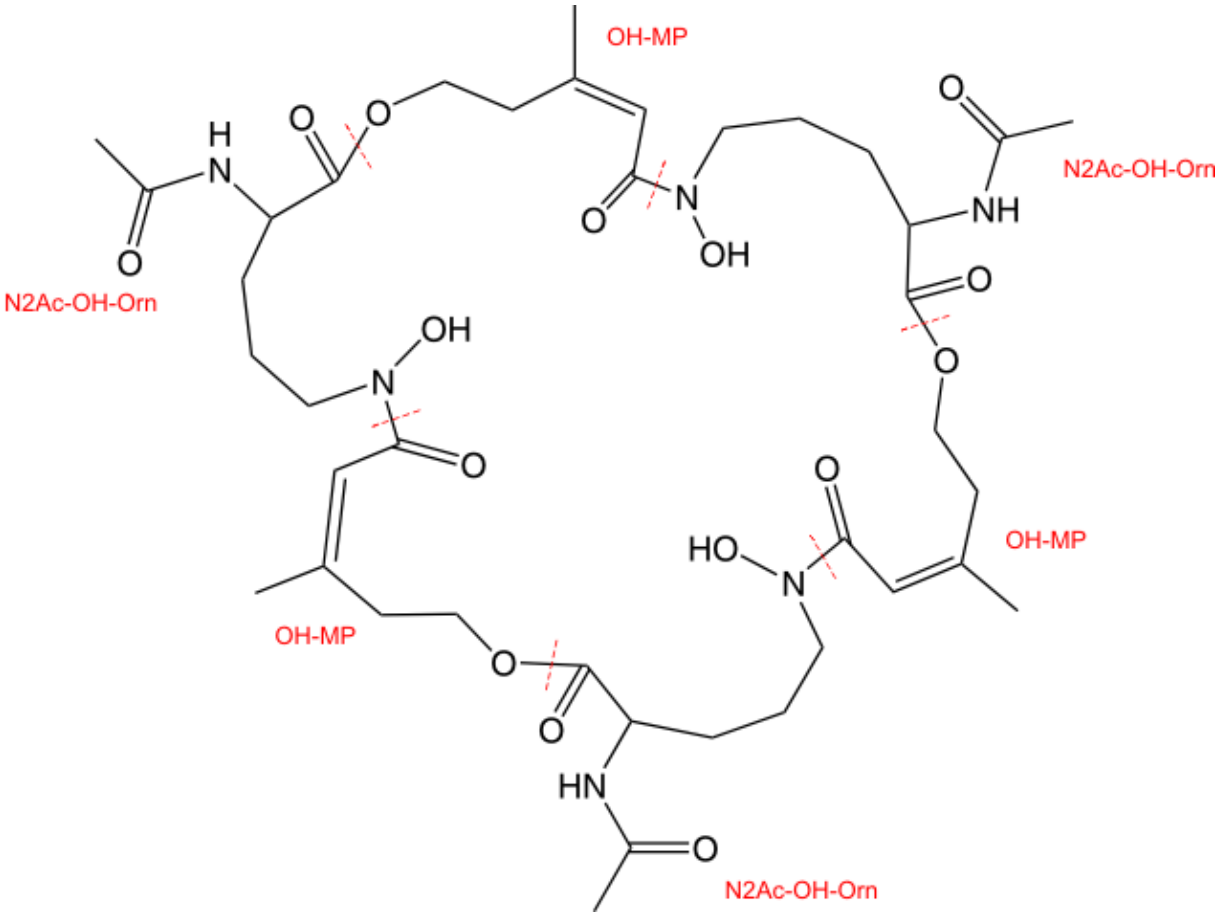


Figure 5.29 Desferri-TAFC

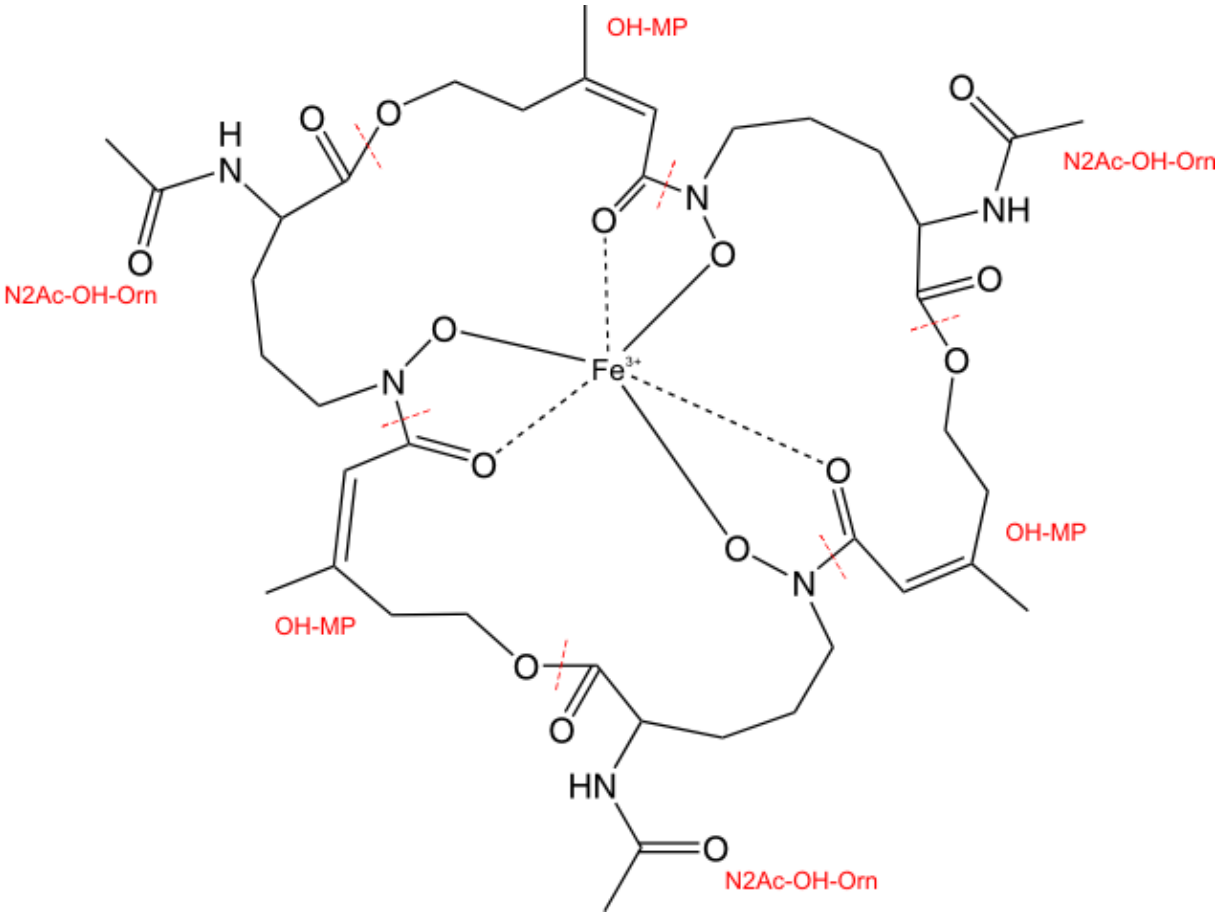


Figure 5.30 Fe-TAFC

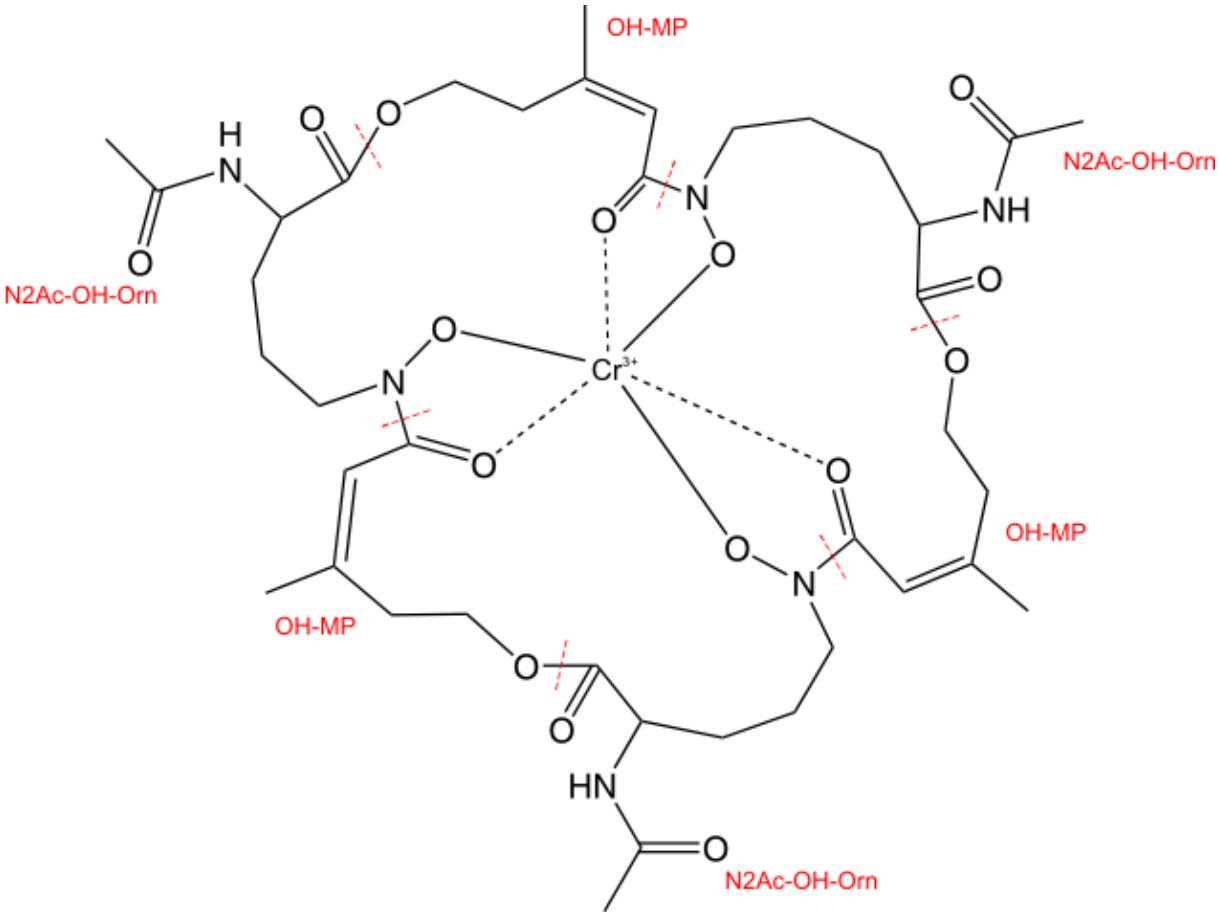


Figure 5.31 Cr-TAFC

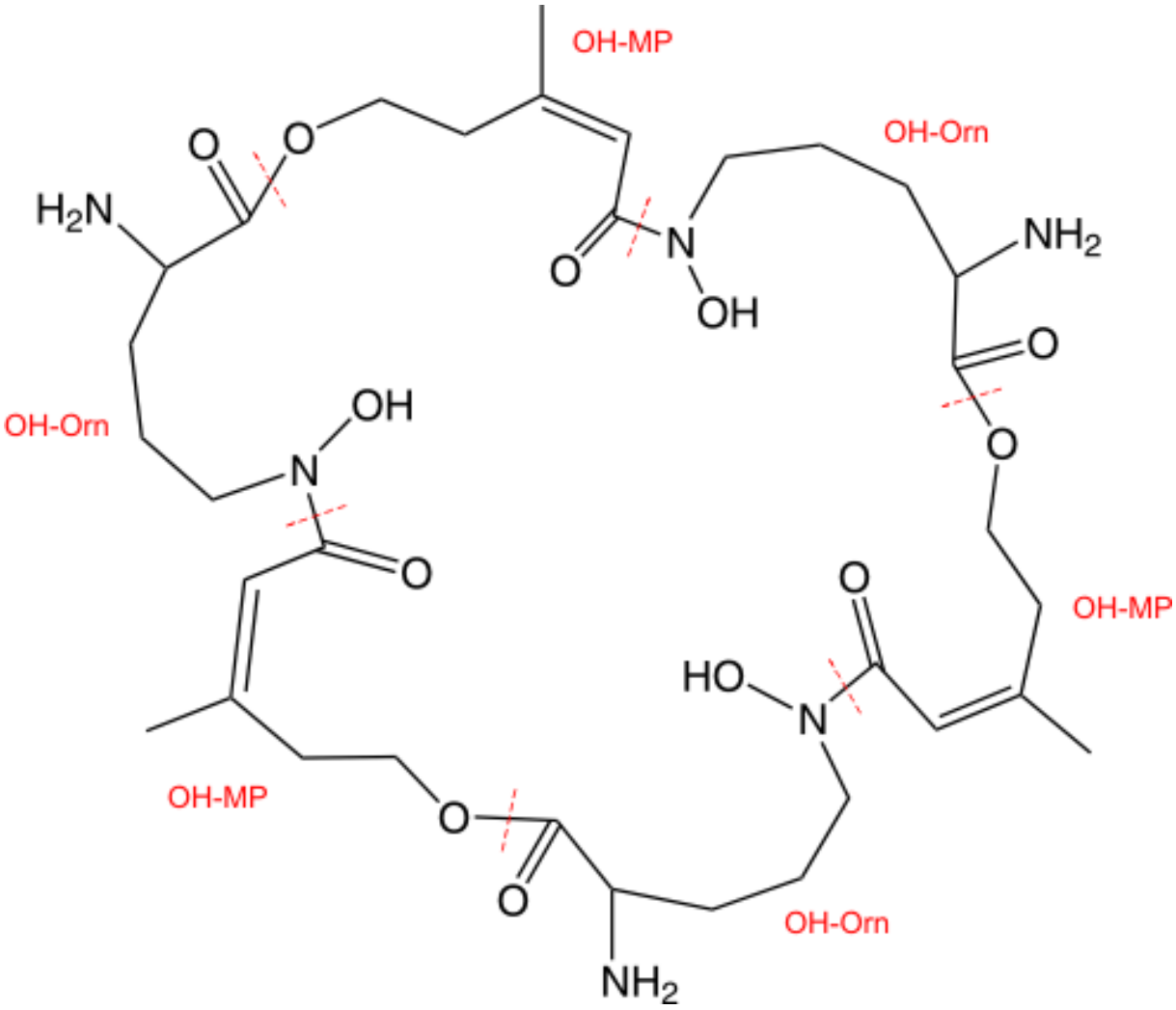


Figure 5.32 Desferri-fusarinin C

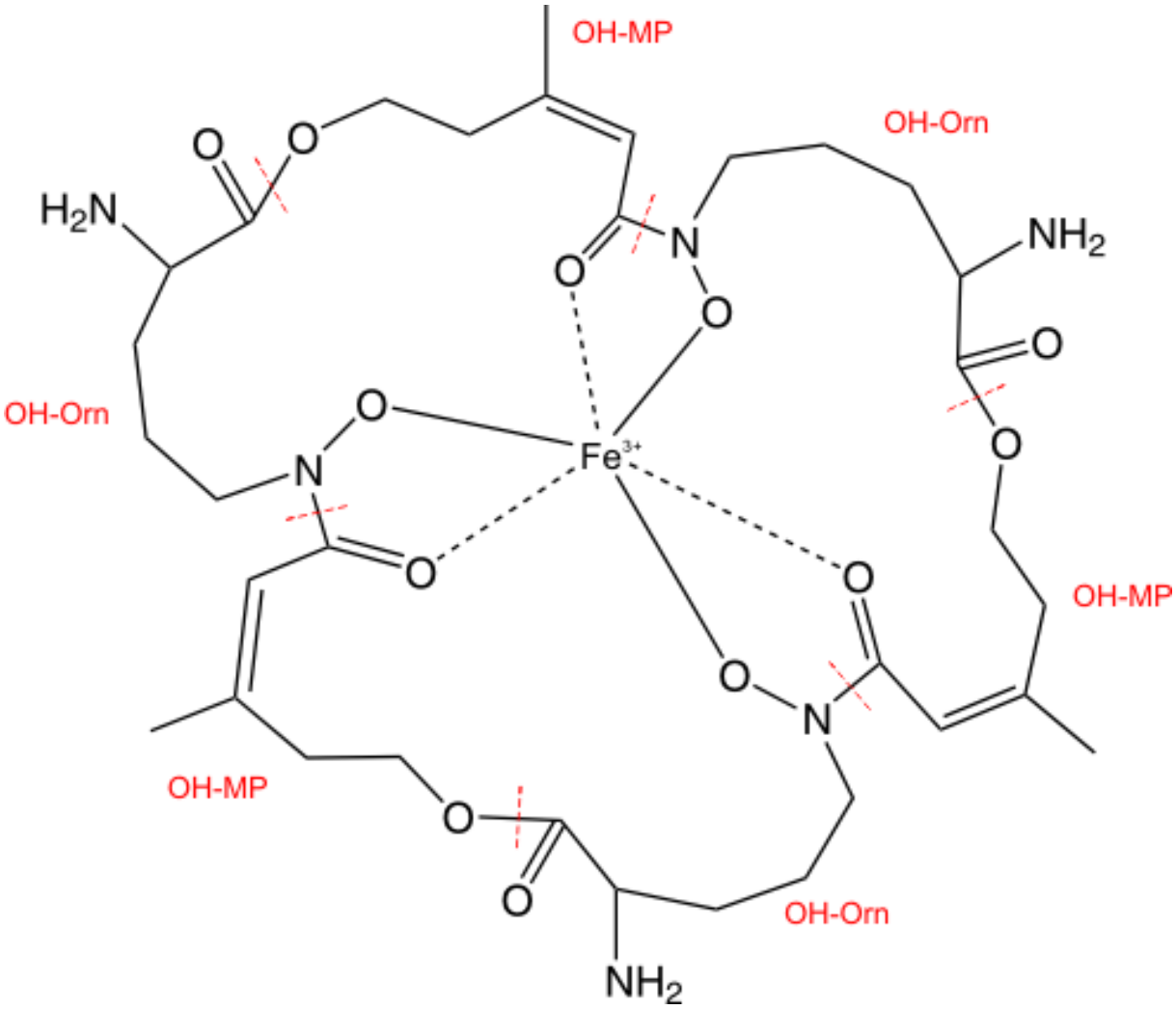


Figure 5.33 Ferri-fusarinin C

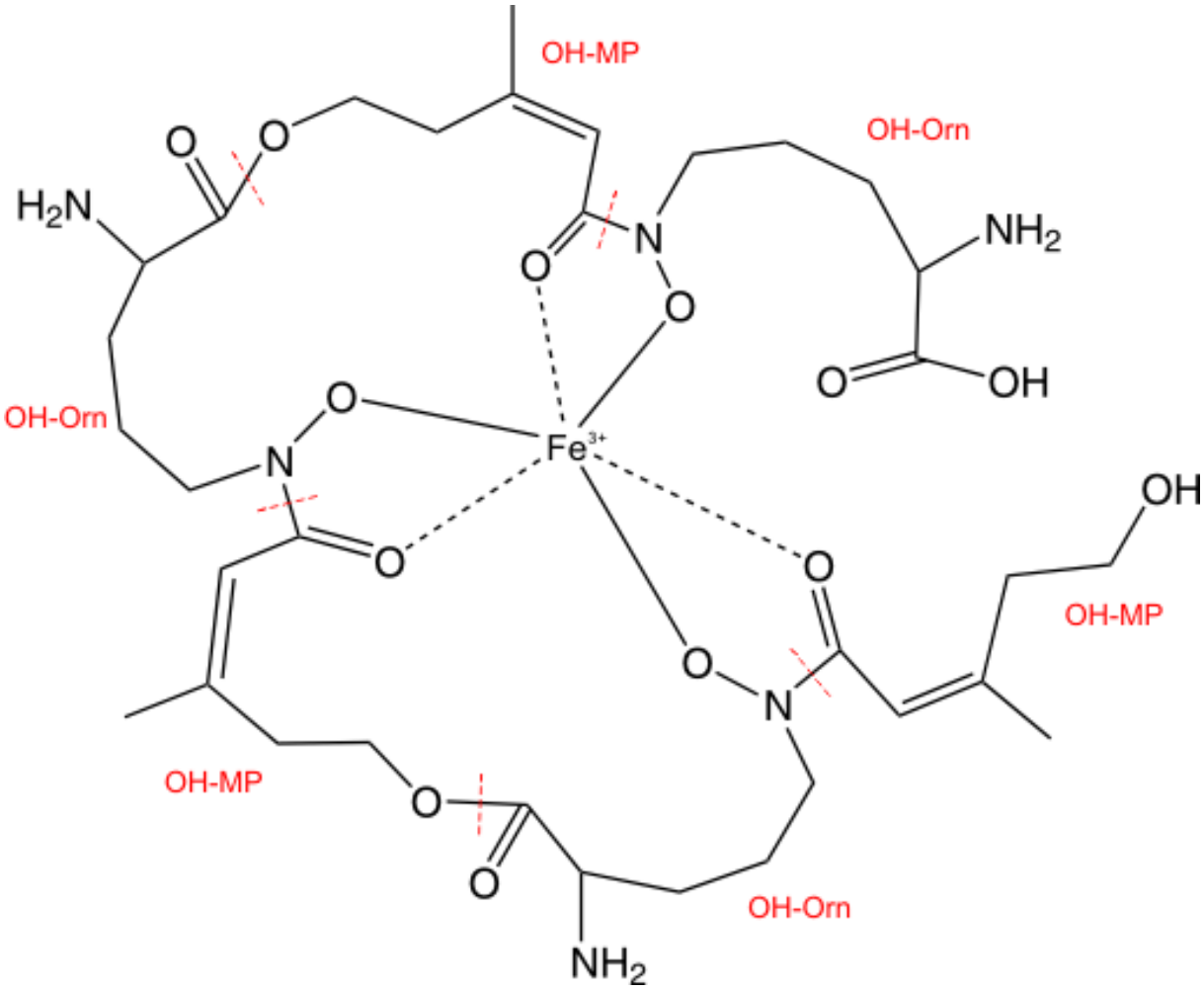


Figure 5.34 Ferri-fusarinin B

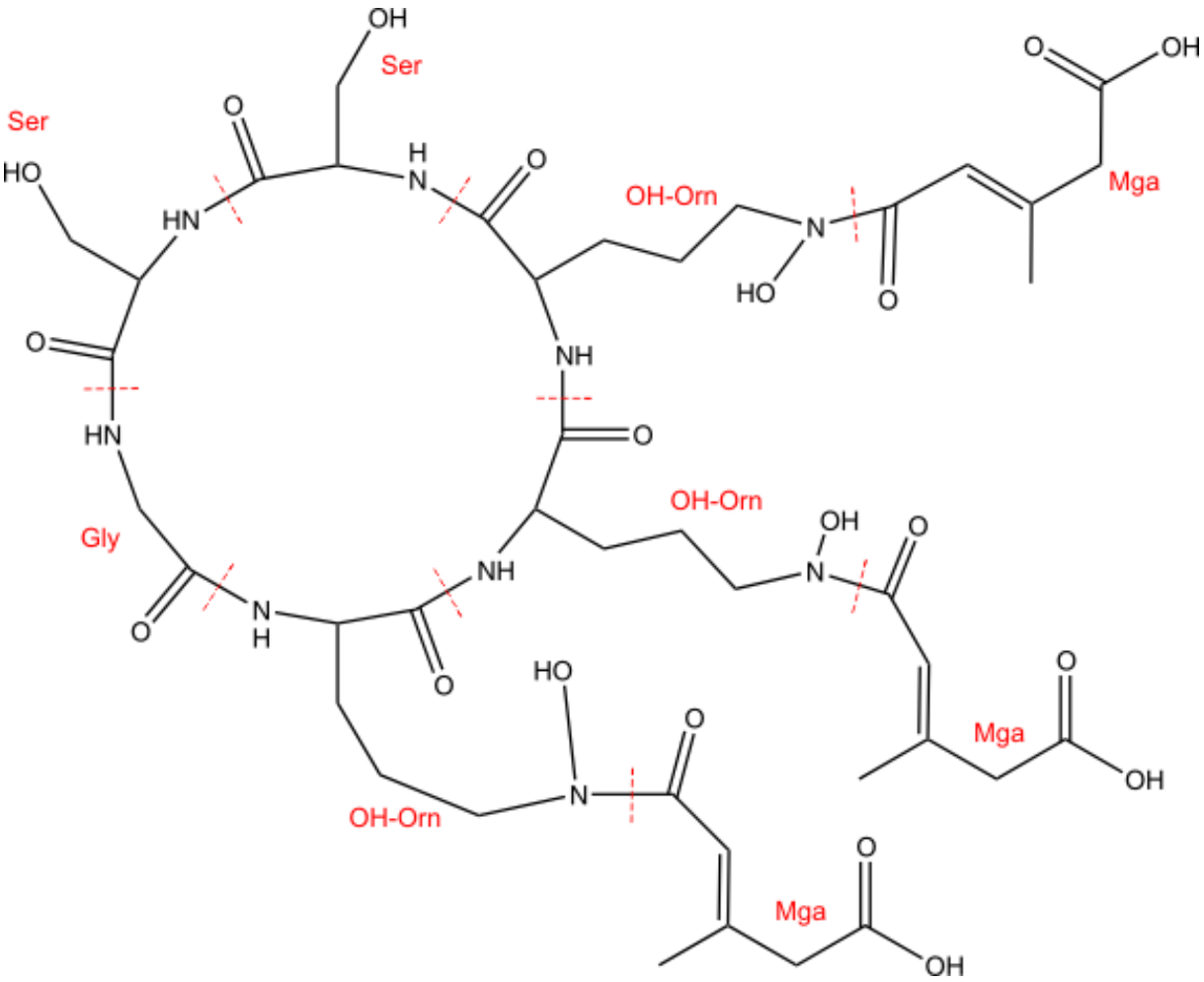


Figure 5.35 Desferri-ferrichrome A

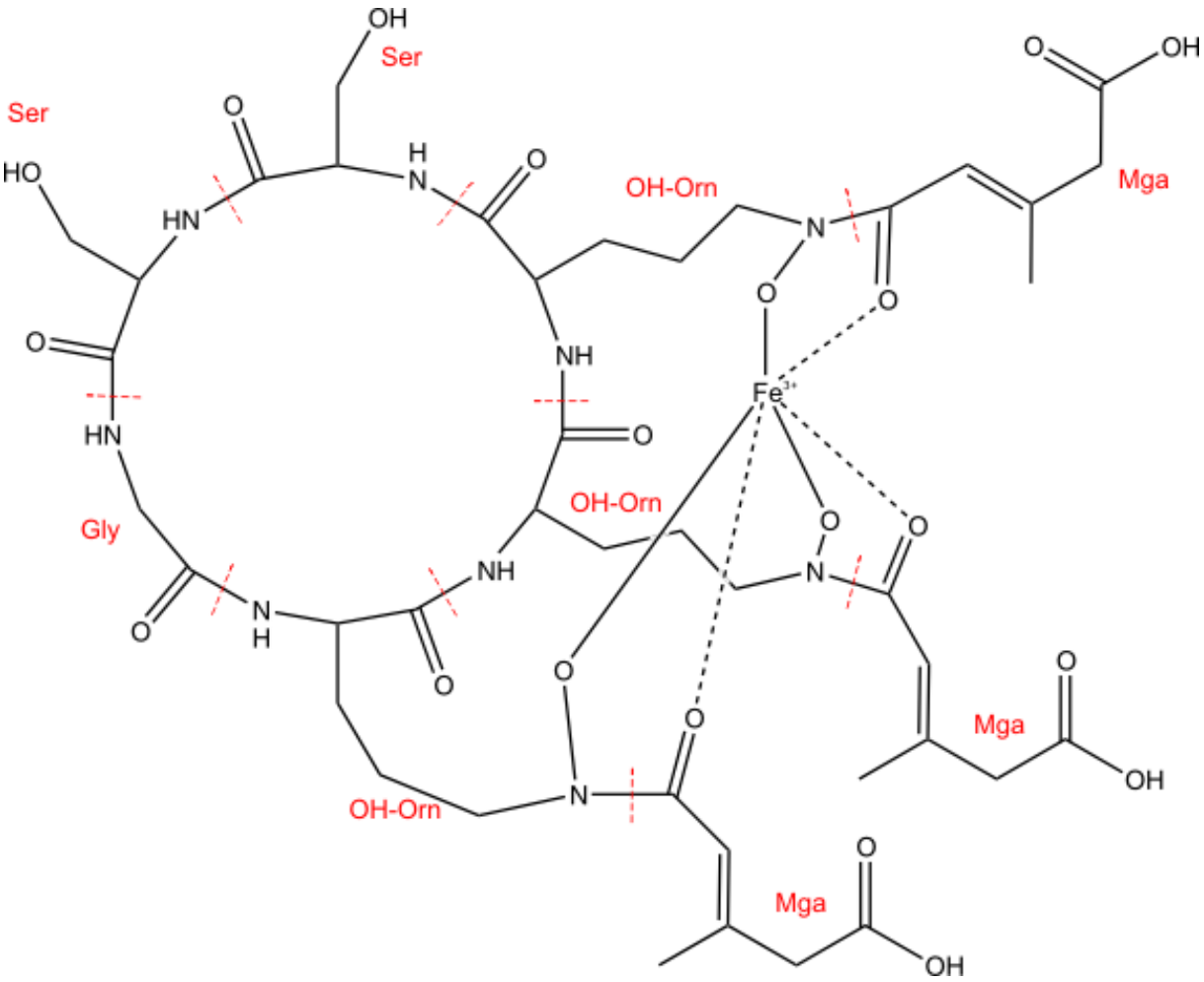


Figure 5.36 Ferri-ferrichrome A

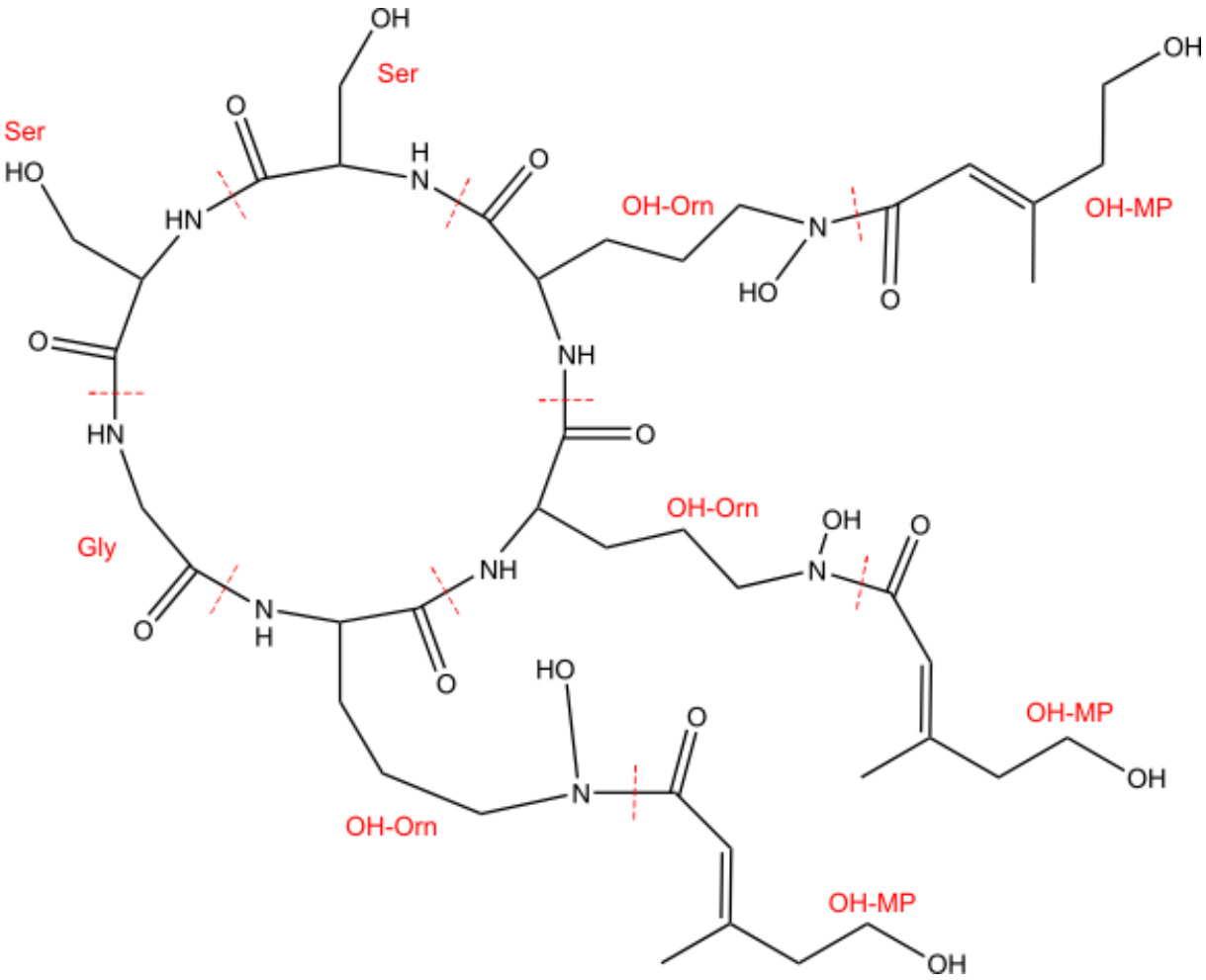


Figure 5.37 Desferri-ferrirubin

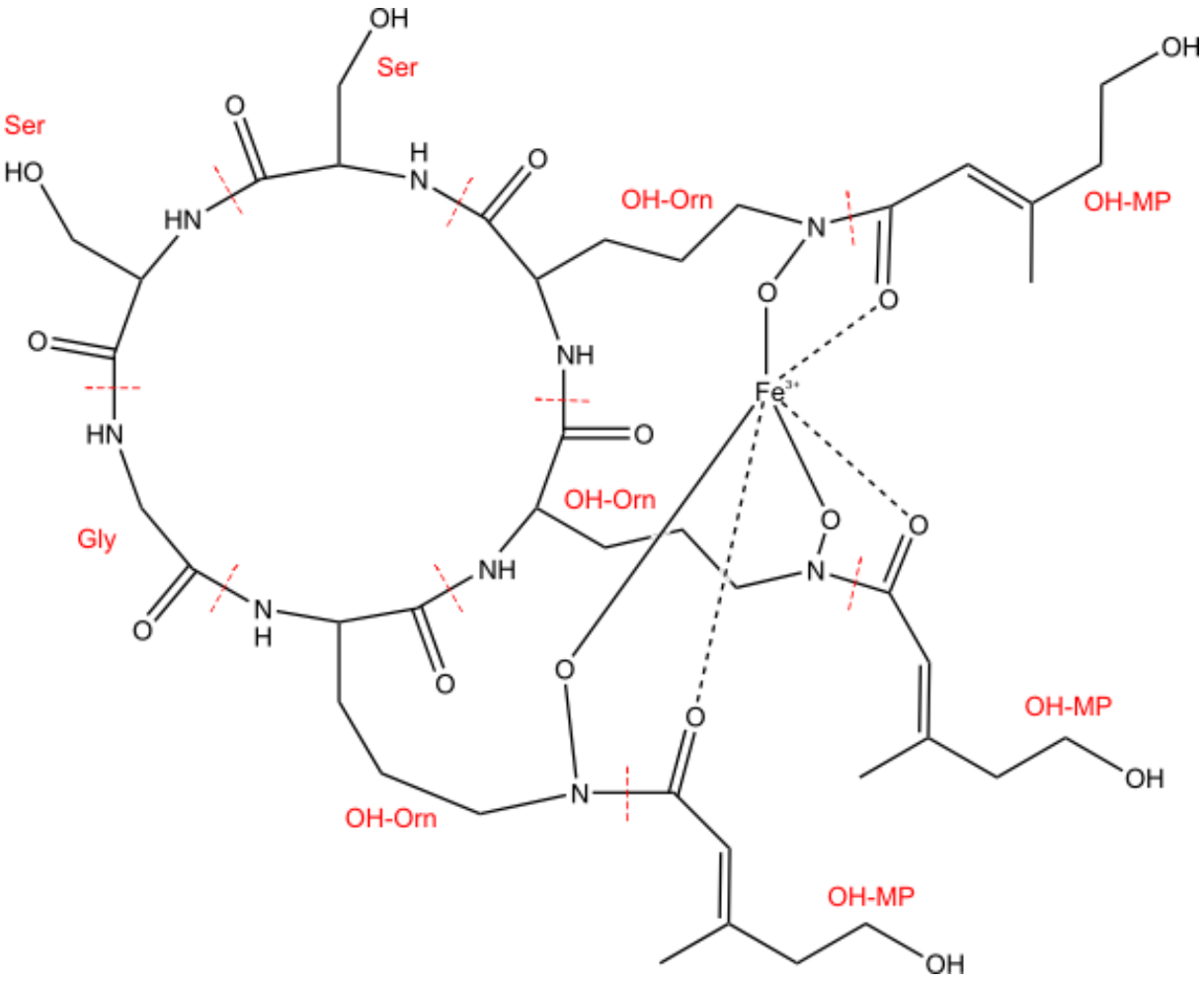


Figure 5.38 Ferri-ferrirubin

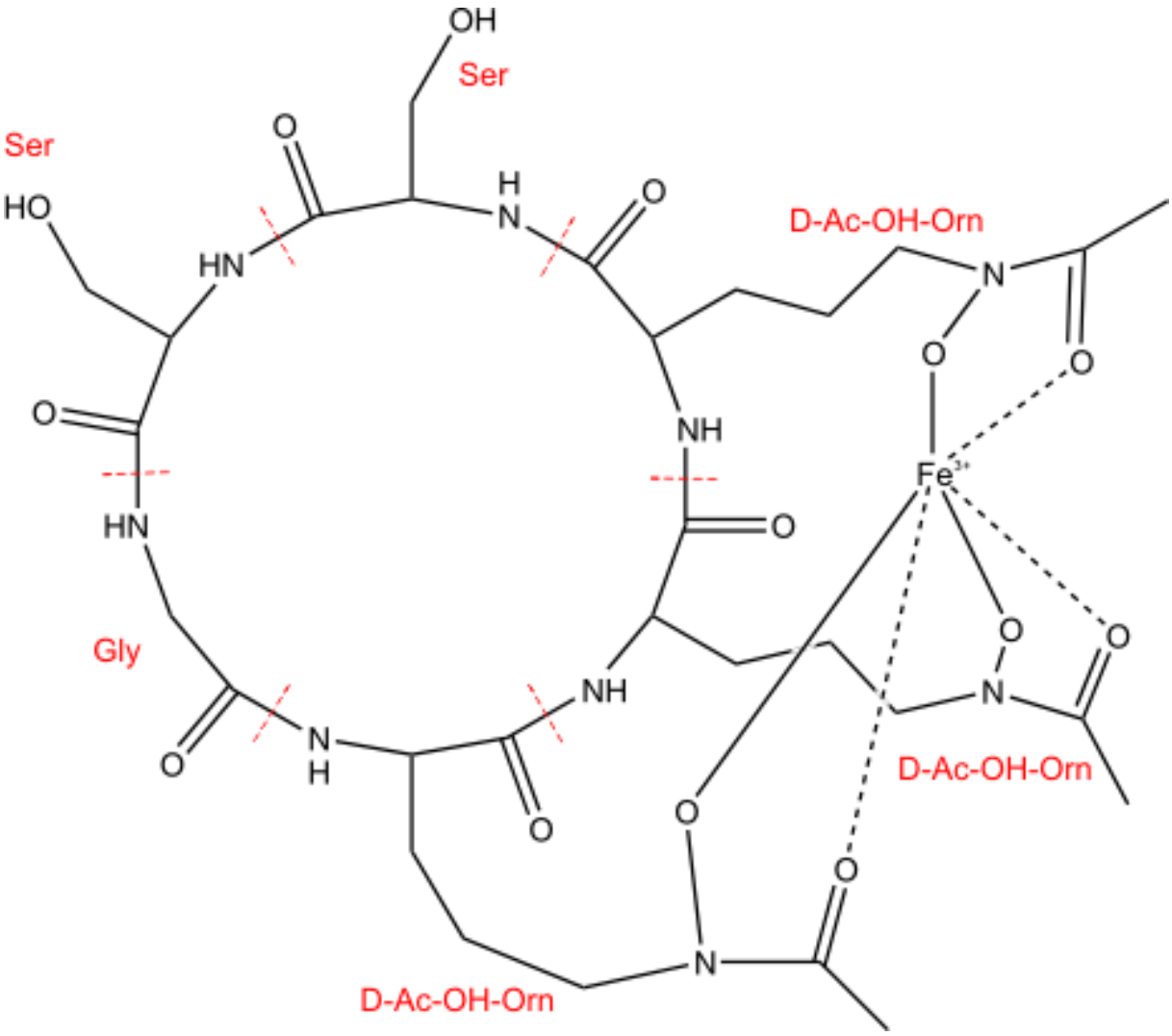


Figure 5.39 Ferri-ferrichrysin

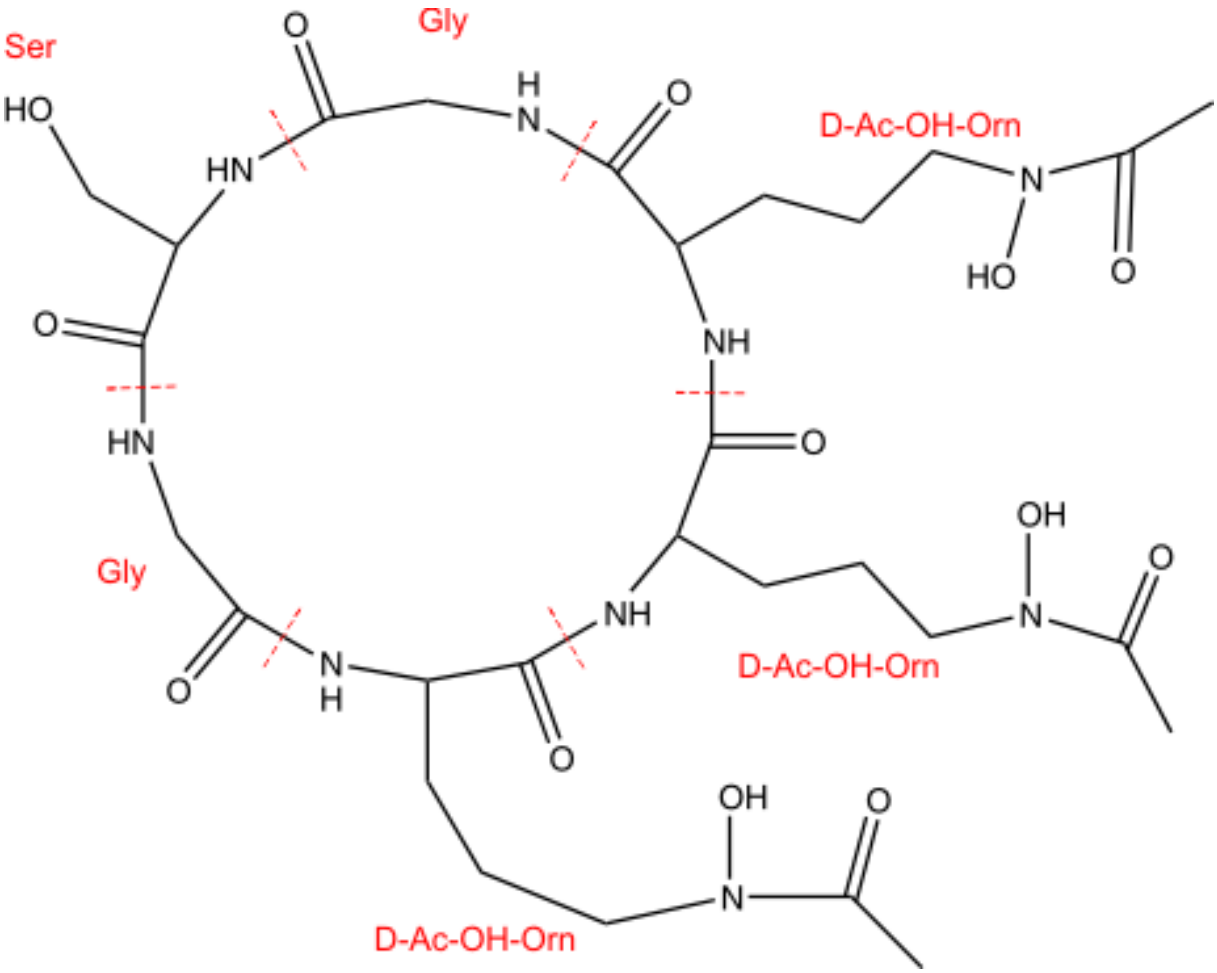


Figure 5.40 Desferri-ferricrocin

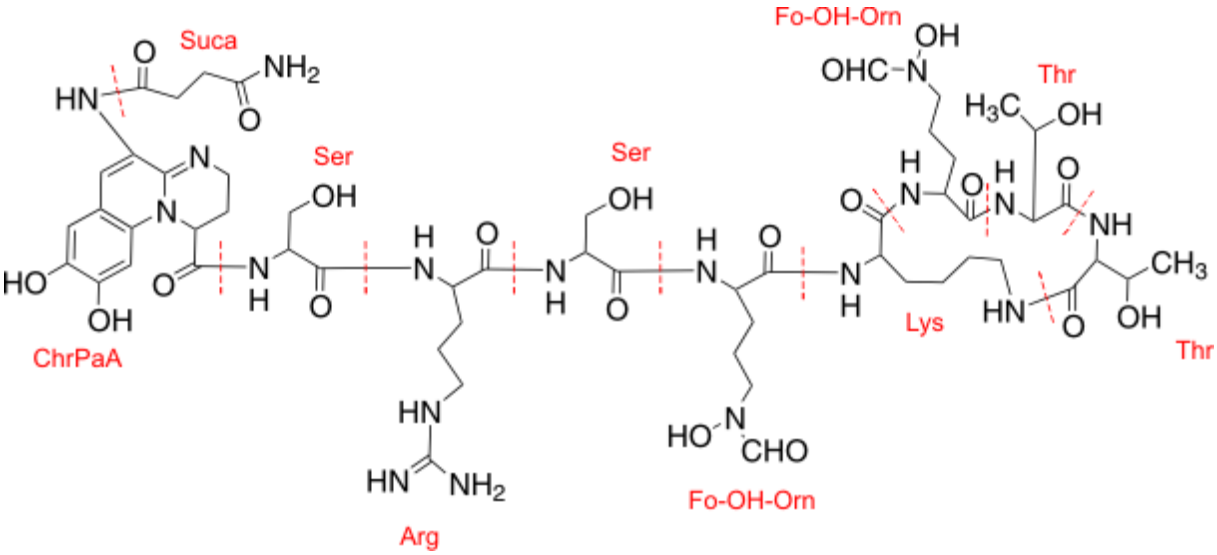


Figure 5.41 Desferri-pyoverdin E

The dataset is also available in a public GNPS library.

5.7 Dataset 6: Correlation of MALDI-MSI with Histology Image

The dataset includes imzML and ibd file (processed data format; centroided spectra; 64-bit floats; no compression), optical scan of ITO MALDI plate and histology image and can be downloaded here.
Reference:
Dominika Luptakova, Tomas Pluhacek, Milos Petrik, Jiri Novak, Andrea Palyzova, Lucie Sokolova, Anton Skriba, Blanka Sediva, Karel Lemr and Vladimir Havlicek.
Non-invasive and Invasive Diagnoses of Aspergillosis in a Rat Model by Mass Spectrometry.
Sci. Rep., vol. 7, pp. 16523, 2017. DOI: **10.1038/s41598-017-16648-z**.

Chapter 6

Publications

6.1 Publication to Cite

Jiri Novak, Anton Skriba and Vladimir Havlicek.
CycloBranch 2: Molecular Formula Annotations Applied to imzML Data Sets in Bimodal Fusion and LC-MS Data Files.
Anal. Chem., vol. 92, no. 10, pp. 6844-6849, 2020. DOI: 10.1021/acs.analchem.0c00170.
Download citation: [ris], [bib]

6.2 Other Publications to Cite

Jiri Novak, Karel Lemr, Kevin A. Schug and Vladimir Havlicek.
CycloBranch: *De Novo* Sequencing of Nonribosomal Peptides from Accurate Product Ion Mass Spectra.
J. Am. Soc. Mass Spectrom., vol. 26, no. 10, pp. 1780-1786, 2015. DOI: 10.1007/s13361-015-1211-1.
Download citation: [ris], [bib]
Jiri Novak, Lucie Sokolova, Karel Lemr, Tomas Pluhacek, Andrea Palyzova and Vladimir Havlicek.
Batch-processing of Imaging or Liquid-chromatography Mass Spectrometry Datasets and *De Novo* Sequencing of Polyketide Siderophores.
BBA - Proteins Proteom., vol. 1865, no. 7, pp. 768-775, 2017. DOI: 10.1016/j.bbapap.2016.12.003.
Download citation: [ris], [bib]
Jiri Novak, Anton Skriba, Jakub Zapal, Marek Kuzma and Vladimir Havlicek.
CycloBranch: An open tool for fine isotope structures in conventional and product ion mass spectra.
J. Mass Spectrom., vol. 53, no. 11, pp. 1097-1103, 2018. DOI: 10.1002/jms.4285.
Download citation: [ris], [bib]
Jiri Novak, Kevin A. Schug and Vladimir Havlicek.
Quantitation of Small Molecules from Liquid Chromatography-Mass Spectrometric Accurate Mass Datasets using CycloBranch.
Eur. J. Mass Spectrom., vol. 29, no. 2, pp. 102-110, 2023. DOI: 10.1177/14690667231164766.
Download citation: [ris], [bib]

6.3 MassSpecBlocks

Jan Privratsky and Jiri Novak.
MassSpecBlocks: a web-based tool to create building blocks and sequences of nonribosomal peptides and polyketides for tandem mass spectra analysis.
J. Cheminformatics, vol. 13, no. 1, pp. 51, 2021. DOI: 10.1186/s13321-021-00530-2.

- Homepage: https://ms.biomed.cas.cz/msb/
- Tutorial: https://www.youtube.com/watch?v=5y8_Ou8luh0

6.4 Other Related Publications

Jiri Novak and Vladimir Havlicek.
Compound Dereplication and *De Novo* Characterization of Small Molecules by Mass Spectrometry / Dereplikace latek a *de novo* charakterizace malych molekul z hmotnostnich spekter.
Chem. Listy, vol. 116, no. 1, pp. 11-19, 2022. DOI: 10.54779/chl20220011 (in Czech only).

Radim Dobias, Anton Skriba, Tomas Pluhacek, Milos Petrik, Andrea Palyzova, Marcela Kanova, Eva Cubova, Jiri Houst, Jiri Novak, David A. Stevens, Goran Mitulovic, Eva Krejci, Petr Hubacek and Vladimir Havlicek.
Noninvasive Combined Diagnosis and Monitoring of *Aspergillus* and *Pseudomonas* Infections: Proof of Concept
J. Fungi, vol. 7, no. 9, pp. 730, 2021. DOI: 10.3390/jof7090730.

Tereza Jurikova, Dominika Luptakova, Olga Kofronova, Anton Skriba, Jiri Novak, Helena Maresova, Andrea Palyzova, Milos Petrik, Vladimir Havlicek and Oldrich Benada.
Bringing SEM and MSI Closer Than Ever Before: Visualizing *Aspergillus* and *Pseudomonas* Infection in the Rat Lungs
J. Fungi, vol. 6, no. 4, pp. 257, 2020. DOI: 10.3390/jof6040257.

Andrea Palyzova, Helena Maresova, Jiri Novak, Jiri Zahradnik and Tomas Rezanka.
Effect of the anti-inflammatory drug diclofenac on lipid composition of bacterial strain *Raoultella* sp. KDF8.
Folia Microbiol., vol. 65, no. 4, pp. 763-773, 2020. DOI: 10.1007/s12223-020-00790-9.

Jiri Novak and Vladimir Havlicek.
Interpretation of peptide product ion mass spectra: Examples / Resene priklady interpretace produktovych spekter peptidu.
Chem. Listy, vol. 114, no. 3, pp. 200-208, 2020. Link (in Czech only).

Ludovit Skultety, Jiri Novak and Vladimir Havlicek.
Fundamentals of the mass spectra interpretation / Zaklady interpretace hmotnostnich spekter.
Chem. Listy, vol. 114, no. 2, pp. 145-154, 2020. Link (in Czech only).

Tomas Pluhacek, Anton Skriba, Jiri Novak, Dominika Luptakova and Vladimir Havlicek.
Analysis of Microbial Siderophores by Mass Spectrometry.
In: Sanjoy K. Bhattacharya (ed.), *Metabolomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1996, pp. 131-153, Springer, 2019. DOI: 10.1007/978-1-4939-9488-5_12.

Andrea Palyzova, Katerina Svobodova, Lucie Sokolova, Jiri Novak and Cenek Novotny.
Metabolic profiling of *Fusarium oxysporum* f. sp. *conglutinans* race 2 in dual cultures with biocontrol agents *Bacillus amyloliquefaciens*, *Pseudomonas aeruginosa*, and *Trichoderma harzianum*.
Folia Microbiol., vol. 64, no. 6, pp. 779-787, 2019. DOI: 10.1007/s12223-019-00690-7.

Dominika Luptakova, Tomas Pluhacek, Milos Petrik, Jiri Novak, Andrea Palyzova, Lucie Sokolova, Anton Skriba, Blanka Sediva, Karel Lemr and Vladimir Havlicek.
Non-invasive and Invasive Diagnoses of Aspergillosis in a Rat Model by Mass Spectrometry.
Sci. Rep., vol. 7, pp. 16523, 2017. DOI: 10.1038/s41598-017-16648-z.

Jiri Novak and Vladimir Havlicek.
Dereplication and Visualization of Fungal Siderophores by CycloBranch.
Mycoses, vol. 60, no. S2, pp. 200, 2017. DOI: 10.1111/myc.12674.

Tomas Pluhacek, Karel Lemr, Dipankar Ghosh, David Milde, Jiri Novak and Vladimir Havlicek.
Characterization of Microbial Siderophores by Mass Spectrometry.
Mass Spectrom. Rev., vol. 35, no. 1, pp. 35-47, 2016. DOI: 10.1002/mas.21461.

Jakub Prichystal, Kevin A. Schug, Karel Lemr, Jiri Novak and Vladimir Havlicek.
Structural Analysis of Natural Products.
Anal. Chem., vol. 88, no. 21, pp. 10338-10346, 2016. DOI: 10.1021/acs.analchem.6b02386.

Chapter 7

Settings

7.1 Settings - Compound Search (MS)

The settings dialog is opened using the command "Search -> Settings... ". Use F1 to open this help.
This page describes only a selection of parameters important for 'Compound Search - MS, LC-MS, MSI' mode. This mode is similar to the dereplication mode with the difference that the compounds are generated from an input list of chemical elements (i.e., an input database of compounds is not required). For other parameters, see the page for dereplication mode.
See also the tutorials Tutorial 3: Identification of unknown compounds in LC-MS data and Tutorial 5: Identification of unknown compou

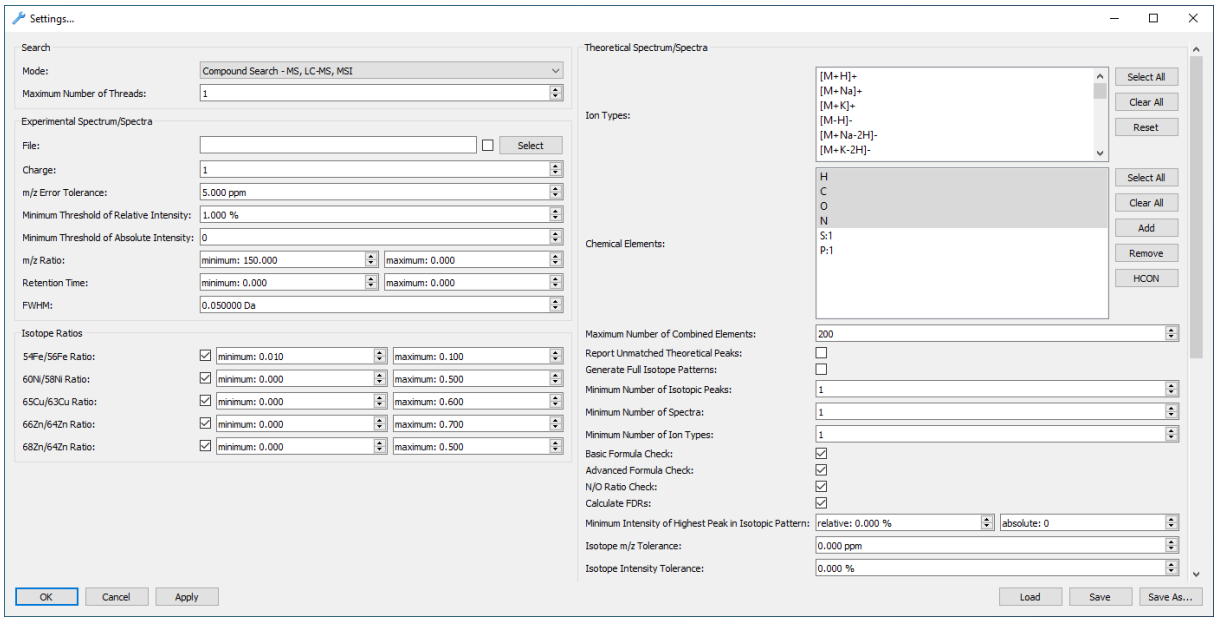


Figure 7.1 Settings dialog - compound search (MS).

7.2 Theoretical Spectrum/Spectra

7.2.1 Chemical Elements

Define and select chemical elements from which the compounds are generated.
Buttons:

- "Select All" - select all elements in the list.
- "Clear All" - unselect all elements in the list.
- "Add" - add a new item at the end of the list - a valid chemical element must be entered.

- "Remove" - remove selected items from the list.
- "HCON" - clear the list and set the default chemical elements H, C, O, N, S, and P.

Use a colon to limit a number of occurrences of an element (e.g., S:2 means that two occurrences of sulfur are allowed).

7.2.2 Maximum Number of Combined Elements

Maximum number of combined chemical elements.

7.2.3 Basic Formula Check

Apply Senior's filtering rules. See https://doi.org/10.1186/1471-2105-8-105 for more details.
In 'Compound Search - MS, LC-MS, MSI' mode, the rules are applied to generated compounds. In other modes, the rules are applied to combinations of neutral losses, see https://doi.org/10.1021/acs.analchem.0c00170 for more details.

7.2.4 Advanced Formula Check

Advanced filtering rules are applied if molecular formulas of compounds are generated. See the tables 1, 2, and 3 in https://doi.org/10.1186/1471-2105-8-105 for more details.

7.2.5 N/O Ratio Check

Check if the number of nitrogen atoms is less or equal to the number of oxygen atoms.

7.2.6 Calculate FDRs

Calculate FDRs for LC-MS and MSI data (experimental feature). It can optionally be disabled to cut the processing time in half.

7.2.7 Minimum Intensity of Highest Peak in Isotopic Pattern

Enter the minimum relative/absolute intensity of the most intense peak in an isotopic pattern. Isotopic patterns with relative/absolute intensities below this value will be kept in the spectrum but not annotated.

7.2.8 Isotope m/z Tolerance

The maximum m/z error of difference between an isotopic peak and the most intense peak in an experimental and theoretical isotopic pattern (0 = disabled) [ppm].
Example, let's have two matched peaks in an isotopic pattern of a compound, an isotopic peak with m/z error 7 ppm and the most intense peak with m/z error 4 ppm:
if m/z Error Tolerance = 10 ppm (i.e. <-10,10>), Isotope m/z Tolerance = 0 ppm (i.e. disabled) => the compound is reported as a correct hit;
if m/z Error Tolerance = 10 ppm (i.e. <-10,10>), Isotope m/z Tolerance = 2 ppm (i.e. <0,2>) => the compound is discarded because 7-4=3 is bigger than 2.

7.2.9 Isotope Intensity Tolerance

The maximum error tolerance of intensities of matched isotopes (0% or 100% = disabled) [in % of relative intensity of the most intense peak].
Example:
Isotope Intensity Tolerance = 10%, Relative Intensity of the Most Intense Peak = 100% => the tolerance of relative intensities of isotopes is 10%;
Isotope Intensity Tolerance = 10%, Relative Intensity of the Most Intense Peak = 50% => the tolerance of relative intensities of isotopes is 5%; etc.

Chapter 8

Settings

8.1 Settings - Database Search (MS/MS)

The settings dialog is opened using the command "Search -> Settings... ". Use F1 to open this help.
This page describes only a selection of parameters important for 'Compare Peaklist with Database - MS/MS' mode.
This mode is similar to de novo sequencing mode with a difference that a database of sequences must be defined.
For other parameters, see the page for dereplication and de novo sequencing modes.
See also the tutorial Tutorial 7: How to search in a database of MS/MS spectra ?

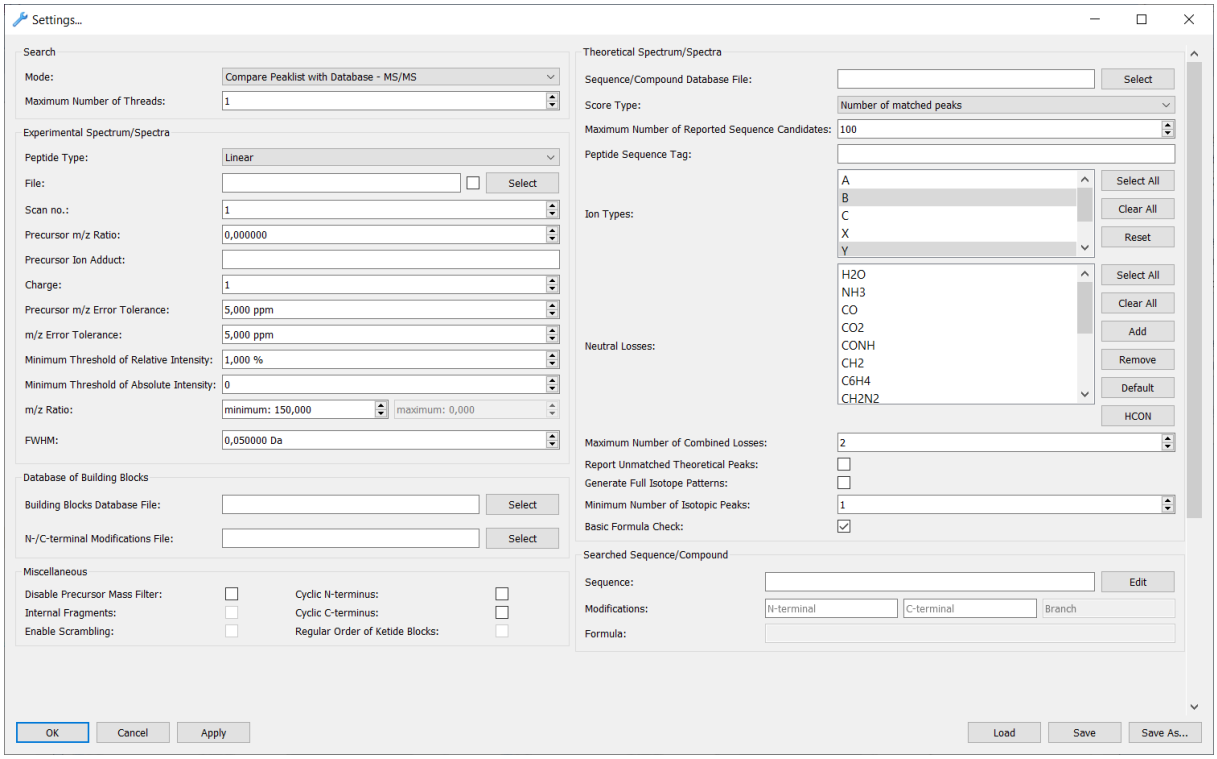


Figure 8.1 Settings dialog - database search (MS/MS).

8.2 Miscellaneous

8.2.1 Disable Precursor Mass Filter

Disable the filtering of compounds by precursor mass.

8.3 Theoretical Spectrum/Spectra

8.3.1 Sequence/Compound Database File (CTRL + T)

A text file containing a database of sequences/compounds. See Sequence/Compound Database Editor.

Chapter 9

Settings

9.1 Settings - De Novo Sequencing (MS/MS)

The settings dialog is opened using the command "Search -> Settings... ". Use F1 to open this help.
This page describes only a selection of parameters important for 'De Novo Search Engine - MS/MS' mode. For other parameters, see the page for dereplication mode.
See also the Tutorials 1-5 (CycloBranch 1.x).

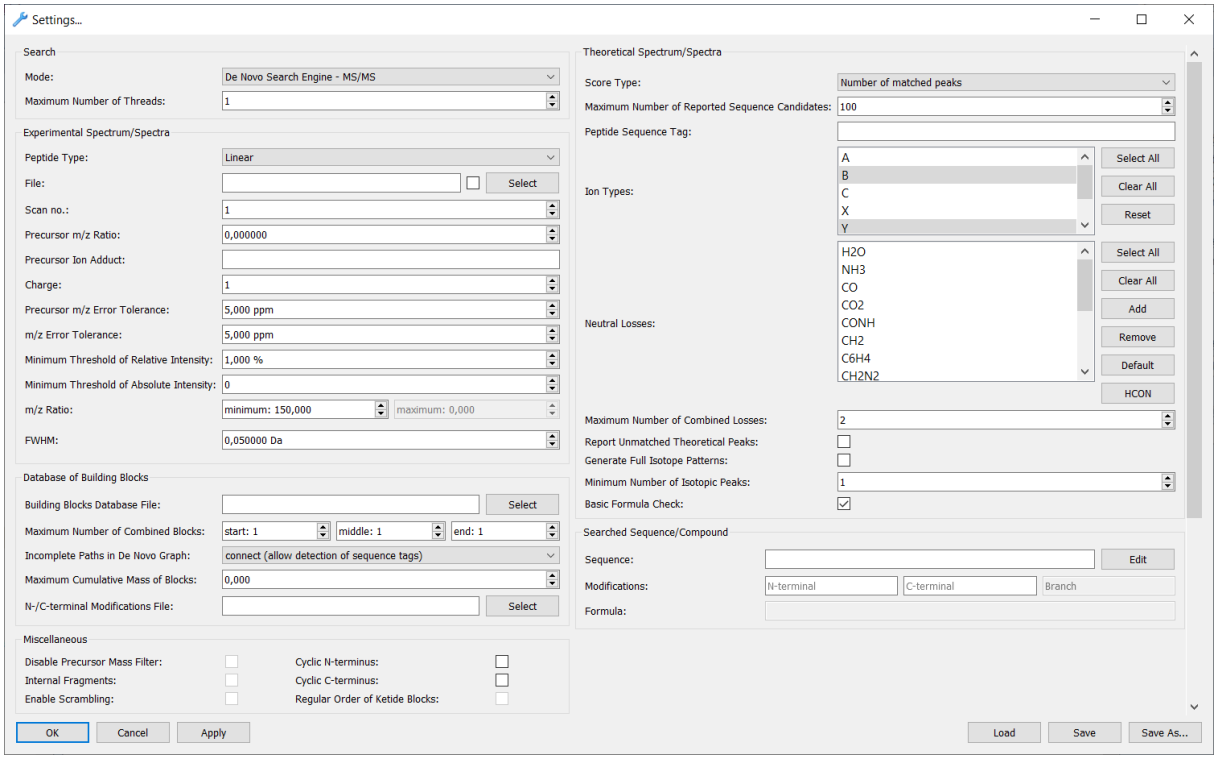


Figure 9.1 Settings dialog - de novo sequencing (MS/MS).

9.2 Experimental Spectrum/Spectra

9.2.1 Peptide Type

Select the type of peptide or polyketide.

- Linear
- Cyclic

- Branched
- Branch-cyclic
- Linear polyketide
- Cyclic polyketide
- Other

The type 'Other' can be used for fragmentation metabolites. See more details here

9.2.2 Scan no.

Number of a spectrum to be processed in a LC-MS/MS data file. The number may be different from an internal scan number stored in the input data file.

9.2.3 Precursor m/z Ratio

Enter the precursor mass-to-charge (m/z) ratio. The value is decharged automatically. The precursor m/z ratio in the input peaklist file is ignored.

9.2.4 Precursor Ion Adduct

Enter the formula of a precursor ion adduct (e.g., Na, K, Li). H is used by default if the field is empty. m/z ratios of theoretical peaks are shifted by mass of adduct. If a metal is used (e.g., FeH-2) and the option "Generate Full Isotope Patterns" is disabled, the peaks of isotopes having natural abundances greater than 3% are generated automatically in theoretical peaklists next to the peaks of most abundant isotopes.
Examples of precursor ion adducts:

Precursor Ion	Precursor Adduct
$[M+H]^+$	H or empty
$[M+Na]^+$	Na
$[M+K]^+$	K
$[M-H]^-$	H or empty
$[M+Fe-2H]^+$	FeH-2
$[M+Fe-3H+Na]^+$	FeH-3Na
$[M+Fe-4H]^-$	FeH-4
$[M+Al-2H]^+$	AlH-2
$[M+Si-3H]^+$	SiH-3
$[M+Zn-H]^+$	ZnH-1
$[M+Zn-2H+Na]^+$	ZnH-2Na
$[M+Zn-3H]^-$	ZnH-3

9.2.5 Charge

Enter the charge of precursor ion. Negative values are allowed. The charge of precursor ion in the input peaklist file is ignored.

9.2.6 Precursor m/z Error Tolerance

Enter the precursor m/z error tolerance in ppm. The value in the input peaklist file is ignored.

9.2.7 m/z Error Tolerance

Enter the fragment m/z error tolerance [ppm].

9.3 Database of Building Blocks

9.3.1 Building Blocks Database File (CTRL + B)

A text file containing a database of building blocks. See Format of Building Blocks Databases.

9.3.2 Maximum Number of Combined Blocks (start / middle / end)

Maximum number of combined building blocks to skip a gap in a de novo graph. A small value speeds up the search and vice versa. Depending on the position of a gap between peaks in a spectrum, the three different values are set up:

- **Start:** Maximum number of combined building blocks to skip a gap leading **from a start point** in a de novo graph.
- **Middle:** Maximum number of combined building blocks to skip a gap in the middle of the graph. **Except gaps leading from a start point or to an end point.**
- **End:** Maximum number of combined building blocks to skip a gap leading **to an end point** in the de novo graph.

In the following scheme, beauverolide I is detected. Since the peak between m/z 1 and 304 is missing, the minimum values start/middle/end are 2/1/1. Otherwise, beauverolide I is not detected when "Incomplete Paths in De Novo Graph" is set up to "keep" or "remove".

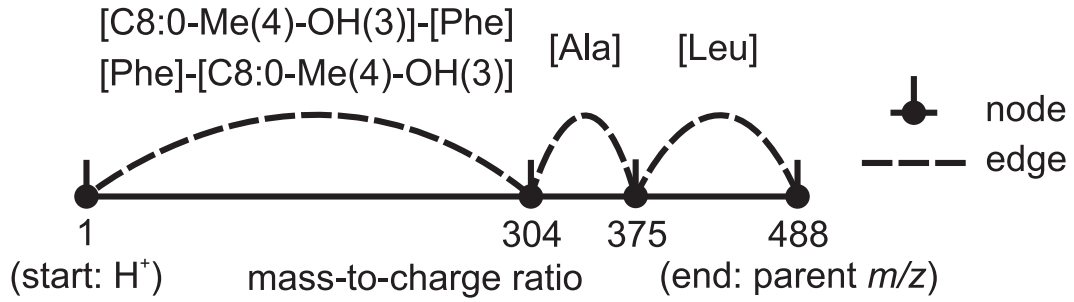


Figure 9.2 De novo graph created from mass spectrum of beauverolide I.

Tip: When an unknown molecule is analyzed, check peptide sequence tags first. Use the following settings:

- **Set up start/middle/end to 1/1/1.**
- **Set up the "Peptide Type" to "linear" or "cyclic".**
- **Set up the "Incomplete Paths in De Novo Graph" to "connect".**

When a branched peptide is analyzed, the minimum values are 1/2/1 (the branch is in the middle), 1/1/3 (the branch is at the end) or 3/1/1 (the branch is at the beginning). If smaller values are used, the branch cannot be detected. See also Branched Sequence Detection.
When a branch-cyclic peptide is analyzed, the minimum values are 1/2/1, 1/1/2 or 2/1/1. See also Branch-cyclic Sequence Detection.

9.3.3 Incomplete Paths in De Novo Graph

The operation to be performed with edges forming incomplete paths (i.e., the paths which do not lead from the start node or do not lead to the end node). The following options can be selected:

- keep (edges are kept - you can see a complete de novo graph in 'View -> Graph')
- remove (edges are removed - speeds up the search)
- connect (edges are connected - allow detection of sequence tags)

9.3.4 Maximum Cumulative Mass of Blocks

Enter the maximum cumulative mass of combined blocks (0 = the maximum mass is unlimited). A small value speeds up the search and vice versa.

9.3.5 N-/C-terminal Modifications File (CTRL + M)

A text file containing a list of N-terminal and C-terminal modifications. See Format of Modification Databases.

9.4 Miscellaneous

9.4.1 Disable Precursor Mass Filter

Disable the filtering of compounds by precursor mass.

9.4.2 Internal Fragments

Generate internal fragment ions in theoretical spectra. This feature is currently available only for branch-cyclic peptides.

9.4.3 Enable Scrambling

Generate scrambled fragment ions of cyclic peptides in theoretical spectra.

9.4.4 Cyclic N-terminus

The water molecule is subtracted from all theoretical N-terminal fragment ions and the theoretical precursor mass. This feature is useful if a linear peptide includes a small cycle close to the N-terminus. If the linear polyketide is selected as the peptide type, the water molecule is subtracted only from the precursor ion.

9.4.5 Cyclic C-terminus

The water molecule is subtracted from all theoretical C-terminal fragment ions and the theoretical precursor mass. This feature is useful if a linear peptide includes a small cycle close to the C-terminus. If the linear polyketide is selected as the peptide type, the water molecule is subtracted only from the precursor ion.

9.4.6 Regular Order of Ketide Blocks

Keep only polyketide sequence candidates whose ketide building blocks are in the regular order [water eliminating block]-[2H eliminating block]-[water eliminating block]-[2H eliminating block], etc.

9.5 Theoretical Spectrum/Spectra

9.5.1 Score Type

A score for peptide-spectrum matches. The following scores are predefined:

- Number of matched peaks
- Sum of relative intensities of matched peaks
- Number of b-ions
- Number of y-ions
- Number of b-ions + y-ions
- Weighted ratio of matched peaks (a sum of relative intensities of matched peaks / a sum of relative intensities of matched and unmatched peaks)
- Cosine Similarity

9.5.2 Maximum Number of Sequence Candidates Reported

A maximum length of an output report with peptide sequence candidates.

9.5.3 Peptide Sequence Tag

Each peptide sequence candidate generated from a de novo graph must fulfil the peptide sequence tag. Otherwise, its theoretical spectrum is not generated and the peptide sequence candidate is excluded from the search. A name of a building block must be enclosed in '[' and ']'. Enclosed building blocks must be separated by '-'. A branch of a branched or a branch-cyclic peptide must be enclosed in "\(" and "\)". Additional backslashes must be used because of a special meaning of '(' and ')' in regular expressions. The tag can be a regular expression in ECMAScript syntax except symbols '[' and ']' which have a special meaning and enclose a building block. Examples of peptide sequence tags:

- `[Val]-[Lac]-[Val]-[Hiv]` (a peptide sequence candidate must contain the following subsequence: valine, lactic acid, valine, 2-hydroxyisovaleric acid)
- `([Val]-[Lac]-[Val]-[Hiv]-*){3}` (the tag must be repeated exactly 3x in a peptide sequence candidate, i.e., the tag corresponds to valinomycin; the symbol '*' means that the previous symbol '-' may or may not be present)
- `[Pro]-[Ile]-[Ile]\([Orn]-[N-Ac-Ile]\)[Phe]` (a tag corresponding to the branched peptide linearized pseudacyclin A)

Note that a peptide sequence candidate is kept if the tag is detected in any linearized sequence of a cyclic, branched or branch-cyclic peptide sequence. See also Format of Sequence Databases.

9.5.4 Ion Types

The following types of fragment ions are defined for MS/MS spectra:

- A (a-ion)
- B (b-ion)
- C (c-ion)
- X (x-ion)
- Y (y-ion)
- Z (z-ion)

Note that y-ions are not supported when cyclic peptides are searched. Fragment ions series of linear and cyclic polyketides are described on extra pages. See Nomenclature of Linear Polyketide Series and Nomenclature of Cyclic Polyketide Series.
Buttons:

- "Select All" - select all fragment ions in the list.
- "Clear All" - unselect all fragment ions in the list.
- "Reset" - b-ion is selected automatically if a cyclic peptide is choosen as "Peptide Type". B-ion and y-ion are selected if a linear, branched or branch-cyclic peptide is choosen.

9.5.5 Neutral Losses

Define and select the types of neutral losses or chemical elements which will be used when generating theoretical spectra.
Buttons:

- "Select All" - select all neutral losses in the list.
- "Clear All" - unselect all neutral losses in the list.

- "Add" - add a new item at the end of the list - a valid molecular formula must be entered.
- "Remove" - remove selected items from the list.
- "Default" - clear the list and define a list of default neutral losses.
- "HCON" - clear the list and set the default chemical elements H, C, O, N, S, and P.

The default neutral losses are defined as follows:

Formula	Description
H2O	water
NH3	ammonia
CO	formyl group
CO2	carboxyl group
CONH	carbamyl group
CH2	shortened carbon chain
C6H4	benzene ring removal
CH2N2	part of Arg side chain
CH2O	Ser side chain
CH2S	Cys side chain
C4H9N	Lys side chain
C4H4N2	His side chain
C4H9N3	Arg side chain
C9H7N	Trp side chain

The molecular formulas of proteinogenic amino acids side chains can be optionally combined from the default values as follows:

Amino acid	Side chain		Amino acid	Side chain
Gly	-		Asp	C2H2O2 (CH2+CO2)
Ala	CH2		Gln	C3H5NO (CH2+CH2+CONH)
Ser	CH2O		Lys	C4H9N
Pro	C3H4 (not defined)		Glu	C3H4O2 (CH2+CH2+CO2)
Val	C3H6 (CH2+CH2+CH2)		Met	C3H6S (CH2+CH2+CH2S)
Thr	C2H4O (CH2+CH2O)		His	C4H4N2
Cys	CH2S		Phe	C7H6 (CH2+C6H4)
Leu	C4H8 (CH2+CH2+CH2+CH2)		Arg	C4H9N3
Ile	C4H8 (CH2+CH2+CH2+CH2)		Tyr	C7H6O (CH2O+C6H4)
Asn	C2H3NO (CH2+CONH)		Trp	C9H7N

In some cases, the list of neutral losses can be replaced by a list of chemical elements (e.g. H, C, O, N, S, and P). This is useful if an experimental spectrum contains peaks corresponding to ions with unknown neutral losses, if a metabolite (i.e. a non-peptidic compound) is fragmented or if the mode 'Compound Search - MS, LC-MS, MSI' is used.

9.5.6 Maximum Number of Combined Losses

Maximum number of combined neutral losses.

9.6 Searched Sequence/Compound

9.6.1 Sequence

A peptide sequence which you are searching for or a peptide sequence tag. A peptide sequence must be entered if "Mode" is set up to "Compare Peaklist(s) with Spectrum of Searched Sequence". Otherwise, the option is similar

to "Peptide Sequence Tag" with a difference that a peptide sequence candidate is not removed from the search but it is just highlighted in the output list of peptide sequence candidates. If "Compare Peaklist(s) with Spectrum of Searched Sequence" mode is used, the only possible format of the expression is a peptide sequence (e.g., `[Val]-[Lac]-[Val]-[Hiv]-[Val]-[Lac]-[Val]-[Hiv]-[Val]-[Lac]-[Val]-[Hiv]` or `[Pro]-[Ile]-[Ile]\([Orn]-[N-Ac-Ile]\)[Phe]`). Regular expressions like `([Val]-[Lac]-[Val]-[Hiv]-*){3}` are not supported in this mode. See also Format of Sequence Databases.
Button 'Edit (CTRL + E)' - Edit the peptide sequence using the Draw Peptide Tool. The sequence is imported into the draw peptide tool and visualized if its syntax is correct (see also Format of Sequence Databases).

9.6.2 N-terminal Modification

A name of an N-terminal modification which belongs to the searched peptide. The name must be defined in N-/C-terminal Modifications File.

9.6.3 C-terminal Modification

A name of a C-terminal modification which belongs to the searched peptide. The name must be defined in N-/C-terminal Modifications File.

9.6.4 Branch Modification

A name of an N-terminal or C-terminal modification which belongs to a branch of a searched peptide (branched and branch-cyclic peptides only). The name must be defined in N-/C-terminal Modifications File.

Chapter 10

Settings

10.1 Settings - Direct Comparison of Experimental and Theoretical Spectrum (MS/MS)

The settings dialog is opened using the command "Search -> Settings... ". Use F1 to open this help.
This page describes only a selection of parameters important for 'Compare Peaklist(s) with Spectrum of Searched Sequence - MS/MS' mode. For other parameters, see the page for dereplication and de novo sequencing modes.
See also the tutorials Tutorial 6: How to compare a theoretical spectrum with a peaklist ?, Tutorial 6: Annotate MS/MS spectrum of a r and Tutorial 9: Draw Peptide Tool.

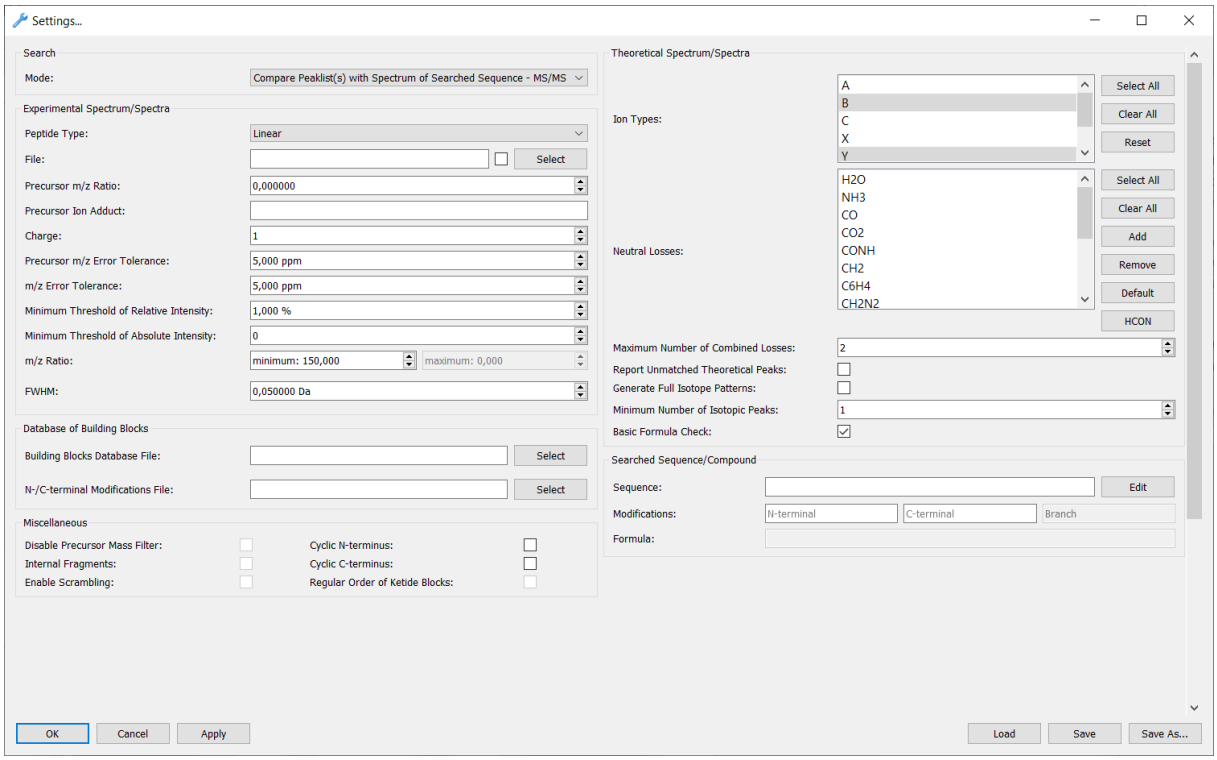


Figure 10.1 Settings dialog - direct comparison of experimental and theoretical spectrum (MS/MS).

10.2 Searched Sequence/Compound

10.2.1 Sequence

A peptide sequence which you are searching for or a peptide sequence tag. A peptide sequence must be entered if "Mode" is set up to "Compare Peaklist(s) with Spectrum of Searched Sequence". Otherwise, the option is similar to "Peptide Sequence Tag" with a difference that a peptide sequence candidate is not removed from the search

but it is just highlighted in the output list of peptide sequence candidates. If "Compare Peaklist(s) with Spectrum of Searched Sequence" mode is used, the only possible format of the expression is a peptide sequence (e.g., `[Val]-[Lac]-[Val]-[Hiv]-[Val]-[Lac]-[Val]-[Hiv]-[Val]-[Lac]-[Val]-[Hiv]` or `[Pro]-[Ile]-[Ile]\([Orn]-[N-Ac-Ile]\)[Phe]`). Regular expressions like `([Val]-[Lac]-[Val]-[Hiv]-*){3}` are not supported in this mode. See also Format of Sequence Databases.

10.2.2 N-terminal Modification

A name of an N-terminal modification which belongs to the searched peptide. The name must be defined in N-/C-terminal Modifications File.

10.2.3 C-terminal Modification

A name of a C-terminal modification which belongs to the searched peptide. The name must be defined in N-/C-terminal Modifications File.

10.2.4 Branch Modification

A name of an N-terminal or C-terminal modification which belongs to a branch of a searched peptide (branched and branch-cyclic peptides only). The name must be defined in N-/C-terminal Modifications File.

10.2.5 Formula

If 'Peptide Type' is set to 'Other', we can define a formula of a neutral molecule instead of a 'Sequence'. This feature is useful for fragmentation of secondary metabolites with structures which cannot be represented as sequences of building blocks. See the Tutorial 6: Annotate MS/MS spectrum of a metabolite.

Chapter 11

Tutorials (CycloBranch 1.x)

11.1 List of Tutorials

- Tutorial 1: Does a peptide have a cycle ?
- Tutorial 2: How to determine a tag ?
- Tutorial 3: How to determine a complete sequence ?
- Tutorial 4: How to determine a branched sequence ?
- Tutorial 5: How to determine terminal modifications ?
- Tutorial 6: How to compare a theoretical spectrum with a peaklist ?
- Tutorial 7: How to search in a database of MS/MS spectra ?
- Tutorial 8: How to search MS spectrum against a database of compounds ?
- Tutorial 9: Draw Peptide Tool
- Tutorial 10: Identification of Branch-Cyclic Peptide with a Long-sized Branch
- Tutorial 11: Identification of Peptide Family using Similarity Search
- Tutorial 12: Identification of Siderophores in MS/MS data
- Tutorial 13: Identification of Siderophores in MALDI-MS and LC-MS data
- Tutorial 14: Identification of Siderophores in Imaging Mass Spectra

A PDF guide through tutorials 1-11 (for CycloBranch 1.0.1216) and through the tutorials 12-14 (for CycloBranch 1.1.570) is available **here**.

11.2 Tutorial 1: Does a peptide have a cycle ?

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.3 Tutorial 2: How to determine a tag ?

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.4 Tutorial 3: How to determine a complete sequence ?

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.5 Tutorial 4: How to determine a branched sequence ?

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.6 Tutorial 5: How to determine terminal modifications ?

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.7 Tutorial 6: How to compare a theoretical spectrum with a peaklist ?

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.8 Tutorial 7: How to search in a database of MS/MS spectra ?

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.9 Tutorial 8: How to search MS spectrum against a database of compounds ?

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.10 Tutorial 9: Draw Peptide Tool

Download mp4 (CycloBranch 1.0.1216)

YouTube (CycloBranch 1.0.1216)

11.11 Tutorial 10: Identification of Branch-Cyclic Peptide with a Long-sized Branch

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.12 Tutorial 11: Identification of Peptide Family using Similarity Search

Download mp4 (CycloBranch 1.0.1216)
YouTube (CycloBranch 1.0.1216)

11.13 Tutorial 12: Identification of Siderophores in MS/MS data

Download mp4 (CycloBranch 1.1.570)
YouTube (CycloBranch 1.1.570)

11.14 Tutorial 13: Identification of Siderophores in MALDI-MS and LC-MS data

Download mp4 (CycloBranch 1.1.570)
YouTube (CycloBranch 1.1.570)

11.15 Tutorial 14: Identification of Siderophores in Imaging Mass Spectra

Download mp4 (CycloBranch 1.1.570)
YouTube (CycloBranch 1.1.570)

Chapter 12

Tutorials (CycloBranch 2.x)

12.1 List of Tutorials

- Tutorial 1: Dereplication in MS1 data
- Tutorial 2: Dereplication in LC-MS data
- Tutorial 3: Identification of unknown compounds in LC-MS data
- Tutorial 4: Dereplication in MSI data and fusion with histology image
- Tutorial 5: Identification of unknown compounds in MSI data
- Tutorial 6: Annotate MS/MS spectrum of a metabolite

12.2 Tutorial 1: Dereplication in MS1 data

The dereplication is the process of annotation of already known compounds (i.e. the database search). To perform the dereplication, click on Search->Settings in the main window of the application and configure the tool.

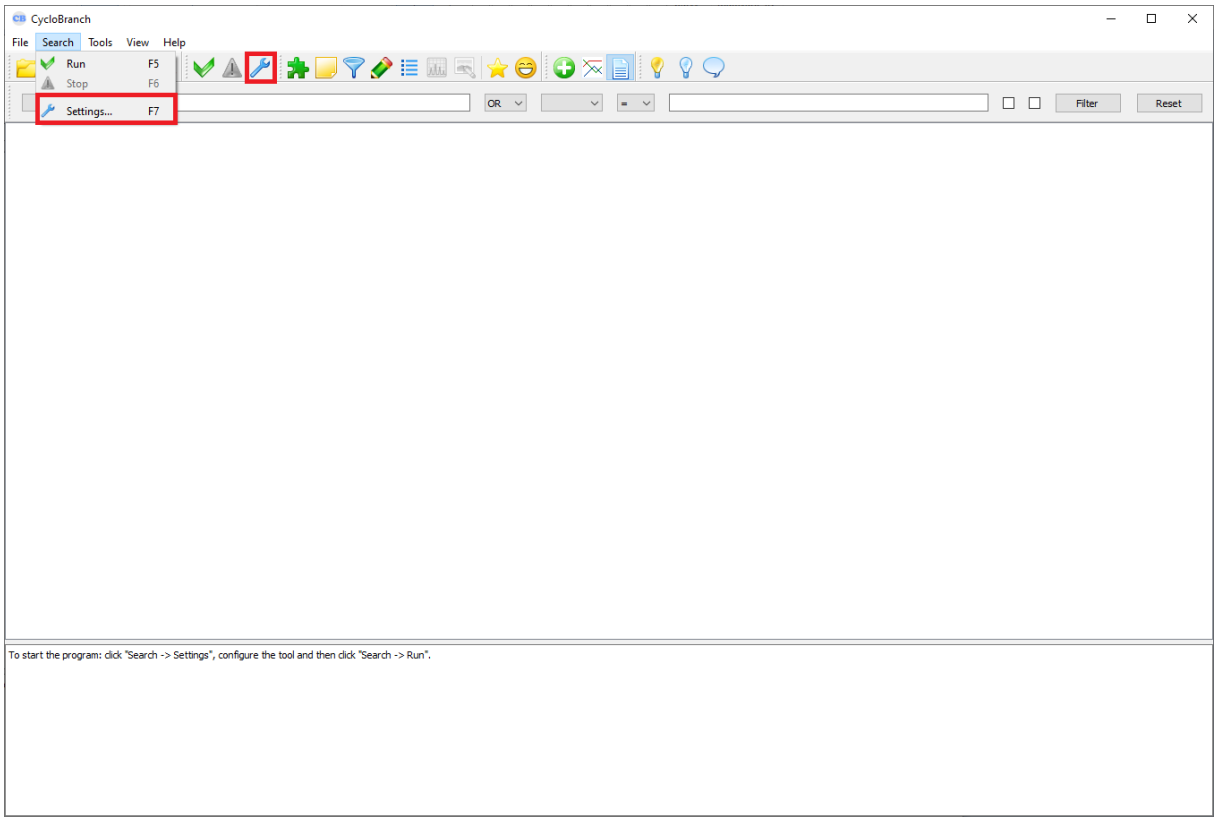


Figure 12.1 Tutorial 1: Main window of the engine.

The most important parameters are:
Mode - select Compare Peaklist(s) with Database - MS, LC-MS, MSI.
File - select an input file with mass spectra. The list of supported file formats is available here (you may need a third-party software for some file formats). Enable the checkbox, if you would like to extract profile mass spectra from the file for later visualization (note: the input file must include profile spectra; the extraction may be time-consuming; and the visualization must be enabled in the spectrum detail window).
Charge - the maximum charge of generated theoretical peaks. If the value is equal to 2, then the theoretical peaks of singly and doubly charged ions are generated. Use -1 for negative mode.
FWHM - see the profile peaks in your spectra and estimate the full width at half maximum (i.e. the width of a representative peak at half of his height; the value corresponds to m/z ratio of a peak divided by its resolution). The theoretical peaks are split or merged using FWHM value prior they are matched with experimental peaks.
Sequence/Compound Database File - select a text file serving as a database. The proper text file can be prepared using Sequence/Compound Database Editor or created using a `template in Excel`. Several in-house prepared databases are also available. In this mode, it is sufficient if you define only a name and a molecular formula (neutral) for every compound. The type of compound can be defined as 'other' and the monoisotopic mass is calculated from the molecular formula automatically. The text file has items separated by tabulator on each row and can also be easily prepared using a script (please, remember that you have to keep the correct number of columns/tabulators to make the file readable by the search engine).
Ion Types - select the types of ions to be searched. Remember that for negative ions (e.g. [M-H]-), the value of charge must be negative (i.e. -1).
Generate Full Isotope Patterns - enable the checkbox to generate the full isotopic patterns of compounds (the parameter FWHM must be estimated correctly). Optionally, you can disable this feature to generate only the theoretical peaks of monoisotopic ions. It may be advantageous for a fast check of a dataset.
Minimum Number of Isotopic Peaks - define the minimum number of peaks which must be matched in an isotopic pattern to report a given compound. The feature **Generate Full Isotope Patterns** must be enabled.

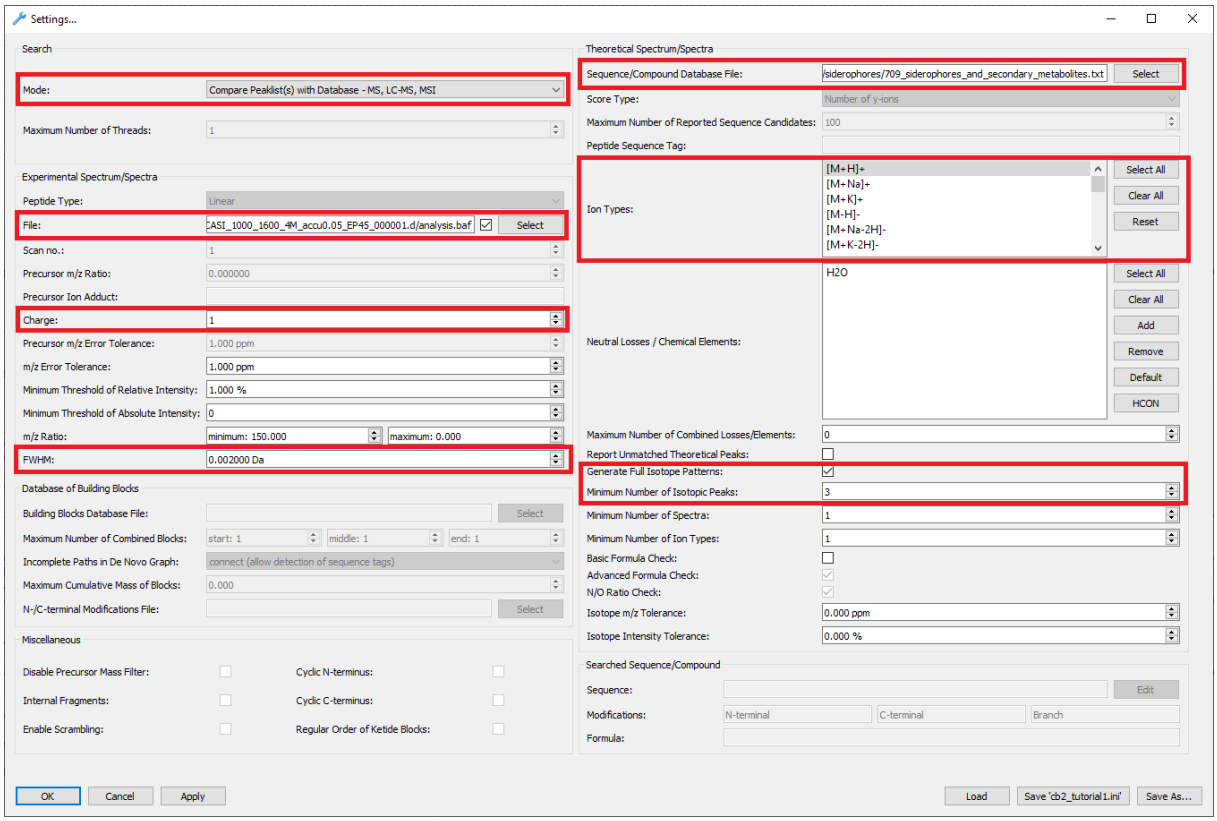


Figure 12.2 Tutorial 1: Configuration for dereplication in MS1 data.

We recommend to save the configuration into an '.ini' file.
Click **OK**. In the main window select **Search->Run** to process the dataset.

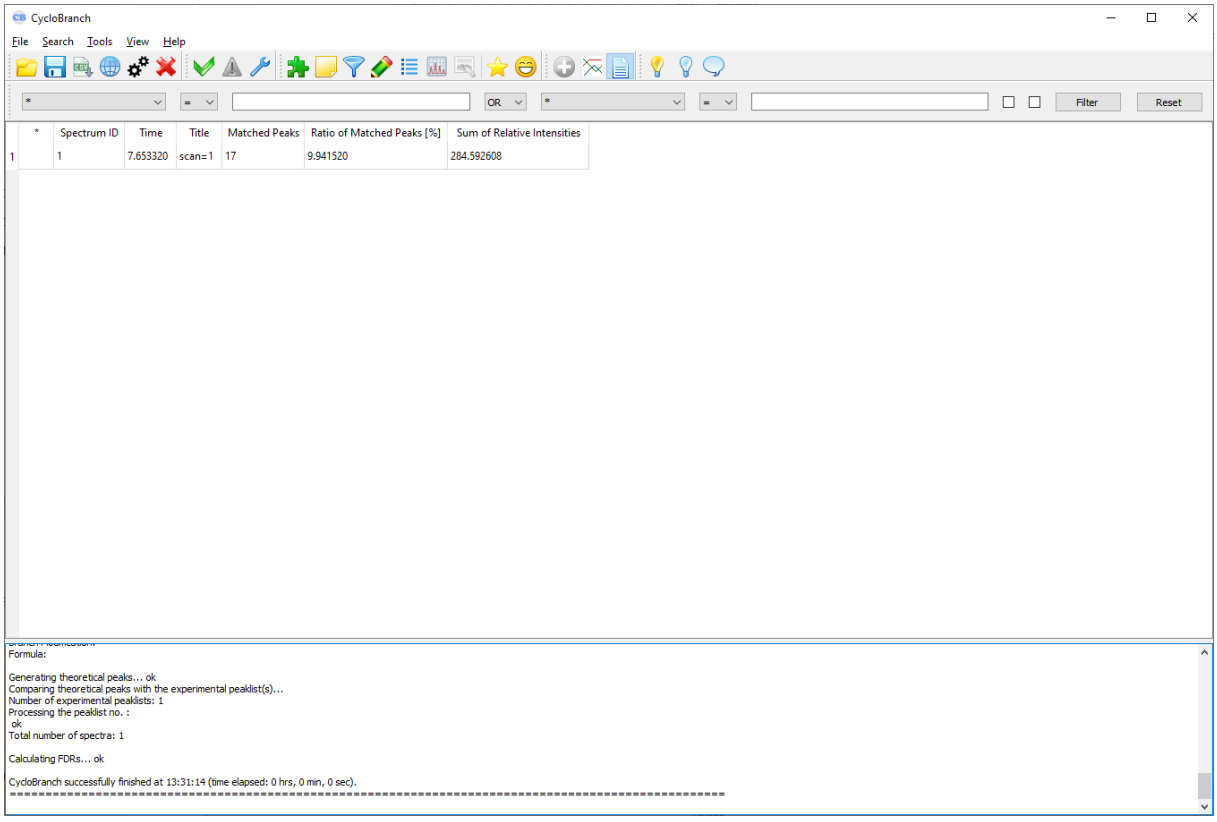
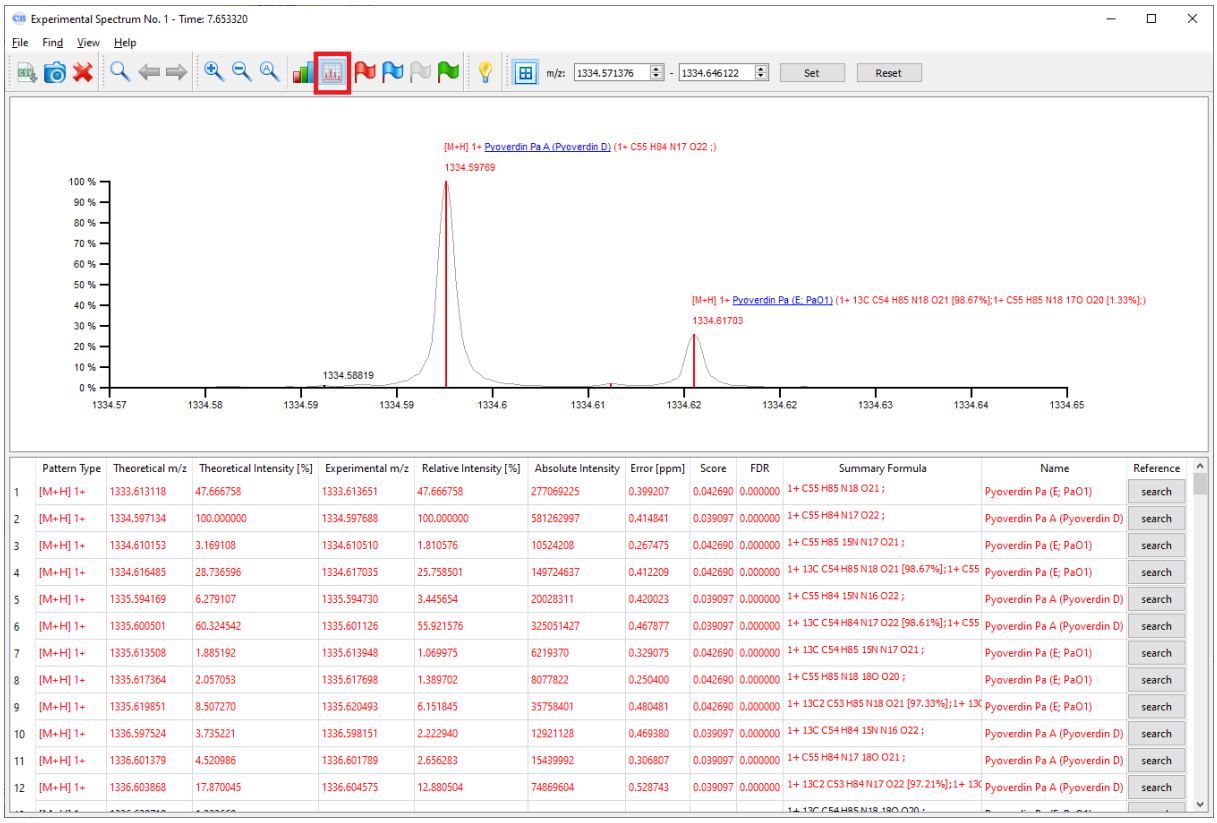


Figure 12.3 Tutorial 1: Search process completed.

Double-click on the row to display the details. Click on the highlighted icon to display the profile spectrum (if the option was enabled in the settings dialog).



	Pattern Type	Theoretical m/z	Theoretical Intensity [%]	Experimental m/z	Relative Intensity [%]	Absolute Intensity	Error [ppm]	Score	FDR	Summary Formula	Name	Reference
1	[M+H] 1+	1333.613118	47.666758	1333.613651	47.666758	277069225	0.399207	0.042690	0.000000	1+ C55 H85 N18 O21 ;	Pyoverdin Pa (E; PaO1)	search
2	[M+H] 1+	1334.597134	100.000000	1334.597688	100.000000	581262997	0.414841	0.039097	0.000000	1+ C55 H84 N17 O22 ;	Pyoverdin Pa A (Pyoverdin D)	search
3	[M+H] 1+	1334.610153	3.169108	1334.610510	1.810576	10524208	0.267475	0.042690	0.000000	1+ C55 H85 15N N17 O21 ;	Pyoverdin Pa (E; PaO1)	search
4	[M+H] 1+	1334.616485	28.736596	1334.617035	25.758501	149724637	0.412209	0.042690	0.000000	1+ 13C C54 H85 N18 O21 [98.67%];1+ C55	Pyoverdin Pa (E; PaO1)	search
5	[M+H] 1+	1335.594169	6.279107	1335.594730	3.445654	20028311	0.420023	0.039097	0.000000	1+ C55 H84 15N N16 O22 ;	Pyoverdin Pa A (Pyoverdin D)	search
6	[M+H] 1+	1335.600501	60.324542	1335.601126	55.921576	325051427	0.467877	0.039097	0.000000	1+ 13C C54 H84 N17 O22 [98.61%];1+ C55	Pyoverdin Pa A (Pyoverdin D)	search
7	[M+H] 1+	1335.613508	1.885192	1335.613948	1.069975	6219370	0.329075	0.042690	0.000000	1+ 13C C54 H85 15N N17 O21 ;	Pyoverdin Pa (E; PaO1)	search
8	[M+H] 1+	1335.617364	2.057053	1335.617698	1.389702	8077822	0.250400	0.042690	0.000000	1+ C55 H85 N18 18O O20 ;	Pyoverdin Pa (E; PaO1)	search
9	[M+H] 1+	1335.619851	8.507270	1335.620493	6.151845	35758401	0.480481	0.042690	0.000000	1+ 13C2 C53 H85 N18 O21 [97.33%];1+ 13C	Pyoverdin Pa (E; PaO1)	search
10	[M+H] 1+	1336.597524	3.735221	1336.598151	2.222940	12921128	0.469380	0.039097	0.000000	1+ 13C C54 H84 15N N16 O22 ;	Pyoverdin Pa A (Pyoverdin D)	search
11	[M+H] 1+	1336.601379	4.520986	1336.601789	2.656283	15439992	0.306807	0.039097	0.000000	1+ C55 H84 N17 18O O21 ;	Pyoverdin Pa A (Pyoverdin D)	search
12	[M+H] 1+	1336.603868	17.870045	1336.604575	12.880504	74869604	0.528743	0.039097	0.000000	1+ 13C2 C53 H84 N17 O22 [97.21%];1+ 13C	Pyoverdin Pa A (Pyoverdin D)	search

Figure 12.4 Tutorial 1: Detail of a mass spectrum. The red peaks were matched.

12.3 Tutorial 2: Dereplication in LC-MS data

Follow the instructions in Tutorial 1 and use the following extra parameter:
Minimum Number of Spectra - the minimum number of consecutive scans in which a compound must be matched to be reported.

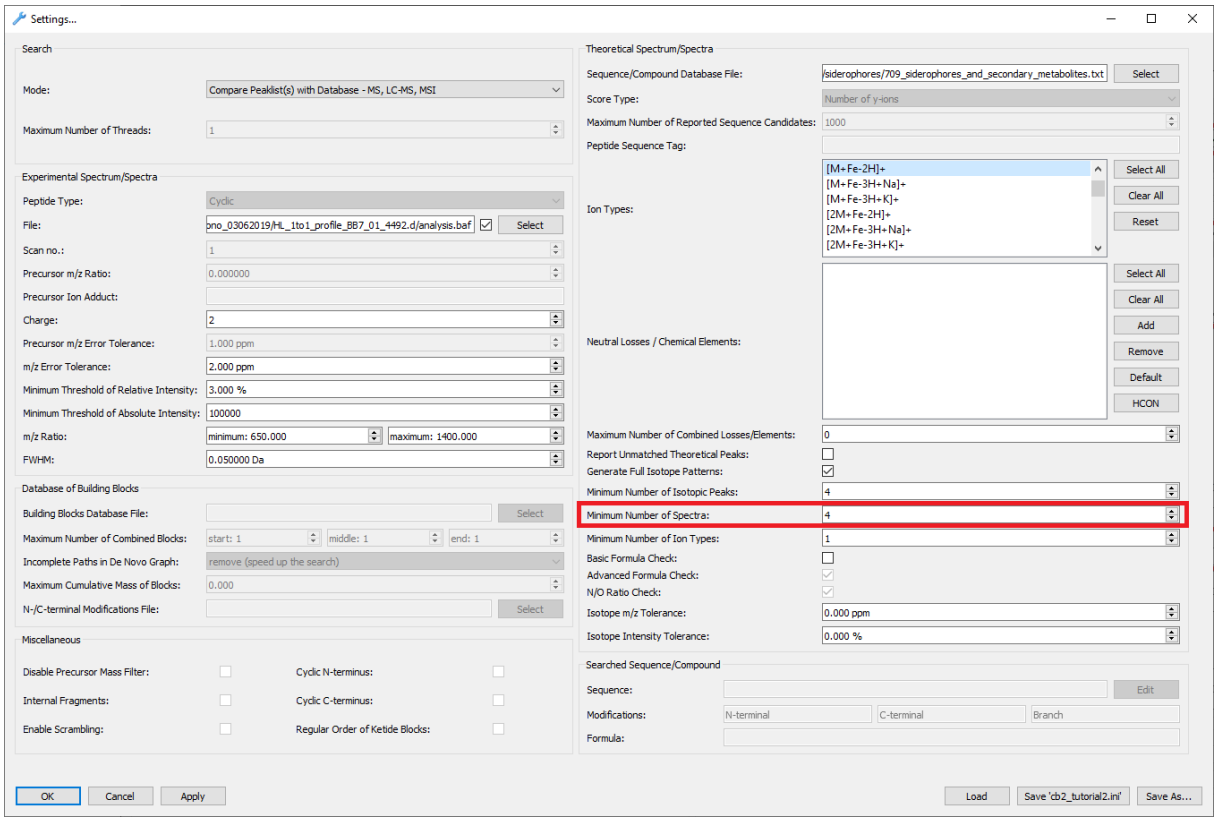


Figure 12.5 Tutorial 2: Configuration for dereplication in LC-MS data.

All MS1 scans included in the input file are listed in the output report. Double-click on any row to display the spectrum from the corresponding scan.

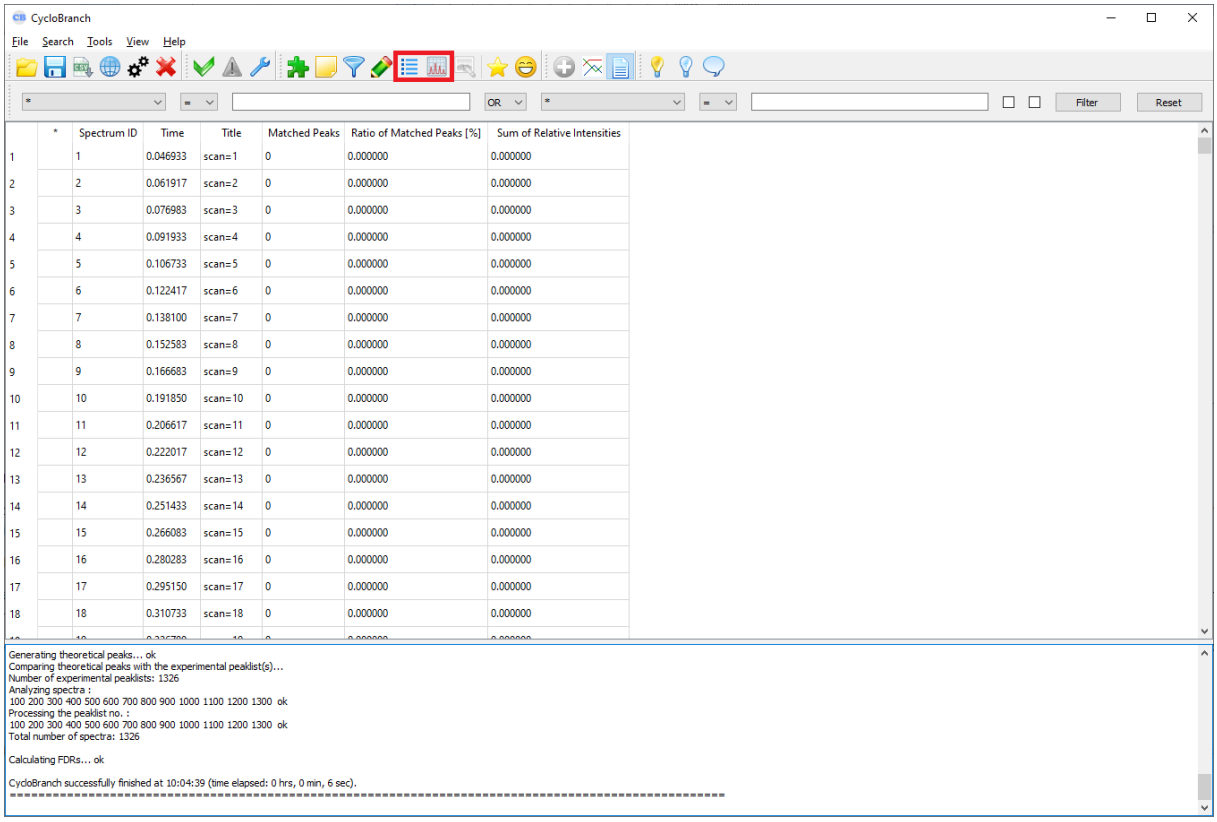


Figure 12.6 Tutorial 2: Search process completed.

In the main window, click on Tools->Chromatogram to see the chromatographic peaks of detected compounds and their retention times. Double-click on any chromatographic peak to open the corresponding mass spectrum.

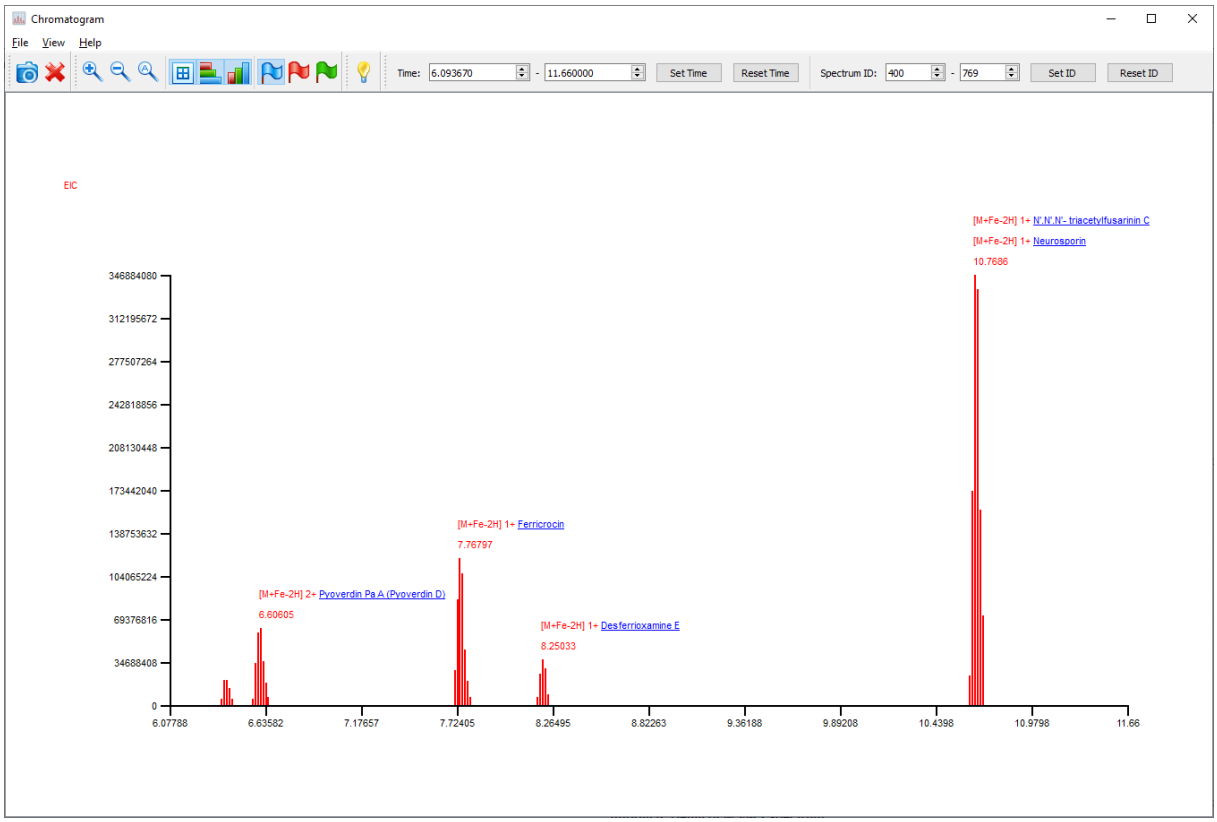


Figure 12.7 Tutorial 2: Chromatogram.

In the main window, click on Tools->Summary Table of Matched Peaks to see the list of all matched peaks in all the scans. Double-click on any row to open the corresponding mass spectrum. The table can be filtered using any column e.g. **Name** or **Theoretical m/z**. Start typing into the text field and the names of identified compounds are automatically suggested. Use the arrows to browse the identified compounds individually.
The table is interactively-connected with the chromatogram. What you see in the table is automatically sent to Chromatogram window. So, you can easily browse extracted ion chromatograms of all identified compounds.

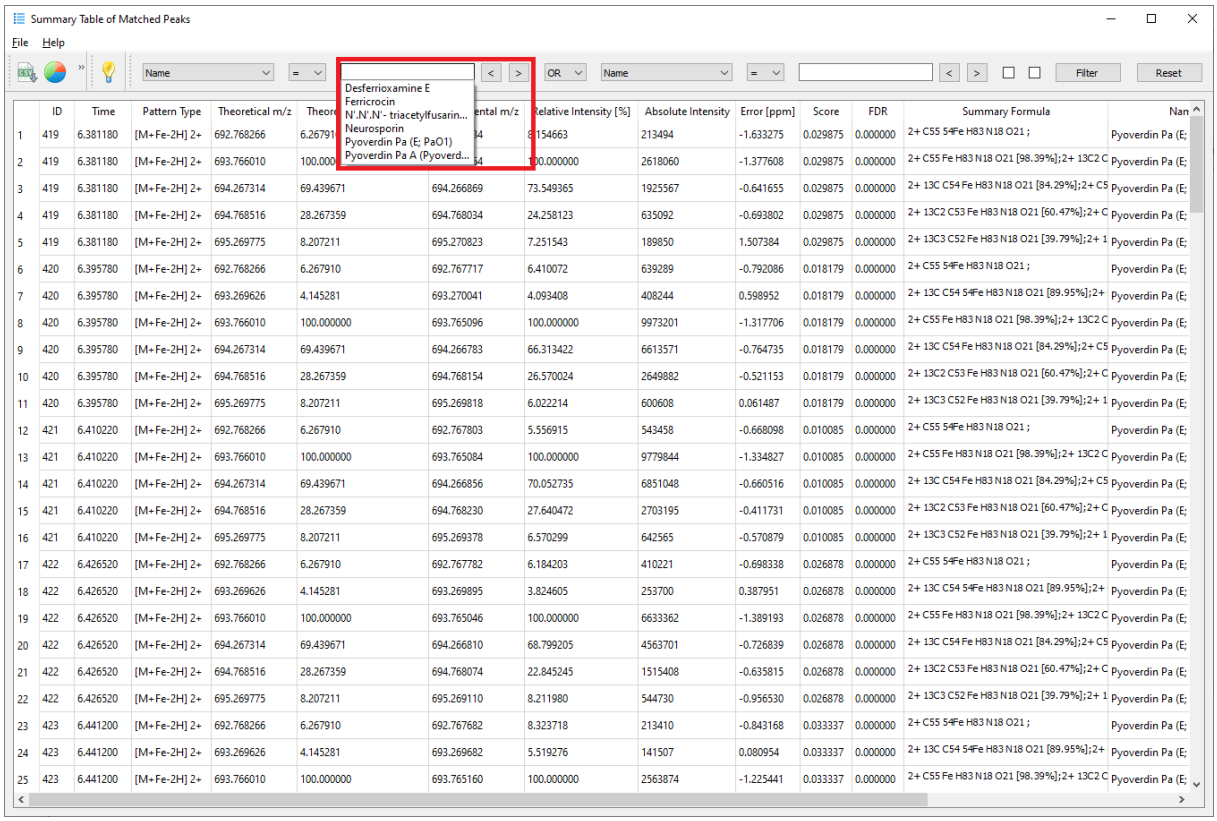


Figure 12.8 Tutorial 2: Summary Table of Matched Peaks.

12.4 Tutorial 3: Identification of unknown compounds in LC-MS data

See the Tutorial 1 and Tutorial 2. In this example, any database of compounds is not used but the molecular formulas of compounds are generated from an input list of chemical elements e.g. H, C, O, N, S, and P. Because the search space grows exponentially with increasing m/z value and the number of input elements, carefully configure the following parameters to limit the number of false-positive formulas reported, to speed-up the search process and to save the computer's main memory:

Mode - select Compound Search - MS, LC-MS, MSI.

m/z Error Tolerance - define m/z error tolerance e.g. 2 ppm. You can use bigger value e.g. 5 or 10 ppm but in this case define also the parameter **Isotope m/z Tolerance**.

Minimum Threshold of Relative Intensity [%] - use e.g. 3 or 5% for an initial check of a dataset. Check the results and optionally reduce to 1% or lower value. Use 0 to disable this feature. The value is used together with **Minimum Threshold of Absolute Intensity**.

Minimum Threshold of Absolute Intensity - use bigger value (e.g. 100000) for an initial check of a dataset and then optionally reduce to lower value. Use 0 to disable this feature. The value is used together with **Minimum Threshold of Relative Intensity**.

m/z Ratio - define the range of m/z values in which the compounds should be detected. If a narrow interval is used (e.g. 100 or 200 Da), the search process is faster. If a wide interval (e.g. 1000 Da) is used, the searching is slower.

FWHM - see the profile peaks in your spectra and estimate the full width at half maximum (i.e. the width of a representative peak at half of his height; the value corresponds to m/z ratio of a peak divided by its resolution). The theoretical peaks are split or merged using FWHM value prior they are matched with experimental peaks.

Ion Types - select the types of ions to be searched. Remember that for negative ions (e.g. [M-H]-), the value of charge must be negative (i.e. -1). For an initial check of a dataset, use only one or a very small number of ions.

Neutral Losses / Chemical Elements - Define an input list of chemical elements. Even if any custom element can be added, use the smallest number of elements as possible. Use the colon to limit the maximum number of occurrences of elements like sulfur or phosphorus (e.g. S:2 or P:1; double-click on an element to edit this value). If the ions like [M+Na]+, [M+K]+ and [M+Fe-2H]+ are searched, do not add the elements Na, K, and Fe, respectively.

They are used automatically. Use the **HCON** button to quickly define the basic elements H, C, O, N, S, and P.
Maximum Number of Combined Losses / Elements - define the maximum number of elements in a molecule (e.g. 150 or 200). Example, the ion [C39H60N6O15+H]+ has 39+60+6+15+1=121 elements.
Generate Full Isotope Patterns - enable the checkbox to generate the full isotopic patterns of compounds (the parameter FWHM must be estimated correctly).
Minimum Number of Isotopic Peaks - define the minimum number of peaks which must be matched in an isotopic pattern to report a given compound. The feature **Generate Full Isotope Patterns** must be enabled. Use e.g. 2 if a big value of **Minimum Threshold of Relative Intensity [%]** is used.
Minimum Number of Spectra - the minimum number of consecutive scans in which a compound must be matched to be reported.
Basic Formula Check - apply Senior's filtering rules to remove unlikely molecular formulas. See https://doi.org/10.1186/1471-2105-8-105 for more details.
Advanced Formula Check - advanced filtering rules are applied if molecular formulas of compounds are generated. See the tables 1, 2, and 3 in https://doi.org/10.1186/1471-2105-8-105 for more details.
N/O Ratio Check - check if the number of nitrogen atoms is less or equal to the number of oxygen atoms.
Isotope m/z Tolerance - the maximum m/z error of difference between an isotopic peak and the most intense peak in an experimental and theoretical isotopic pattern (0 = disabled); see the definition of Isotope m/z Tolerance.
Isotope Intensity Tolerance - the maximum error tolerance of intensities of matched isotopes (e.g. 10%); see the definition of Isotope Intensity Tolerance.

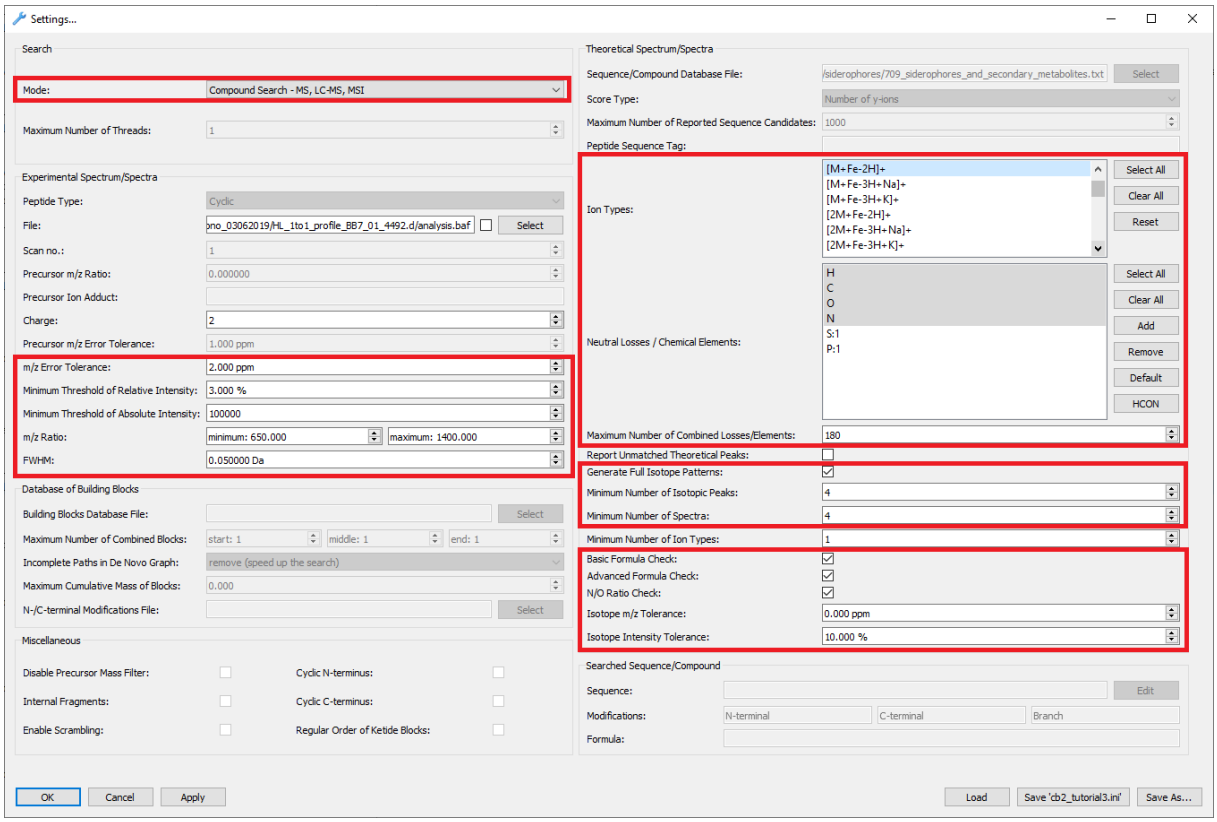
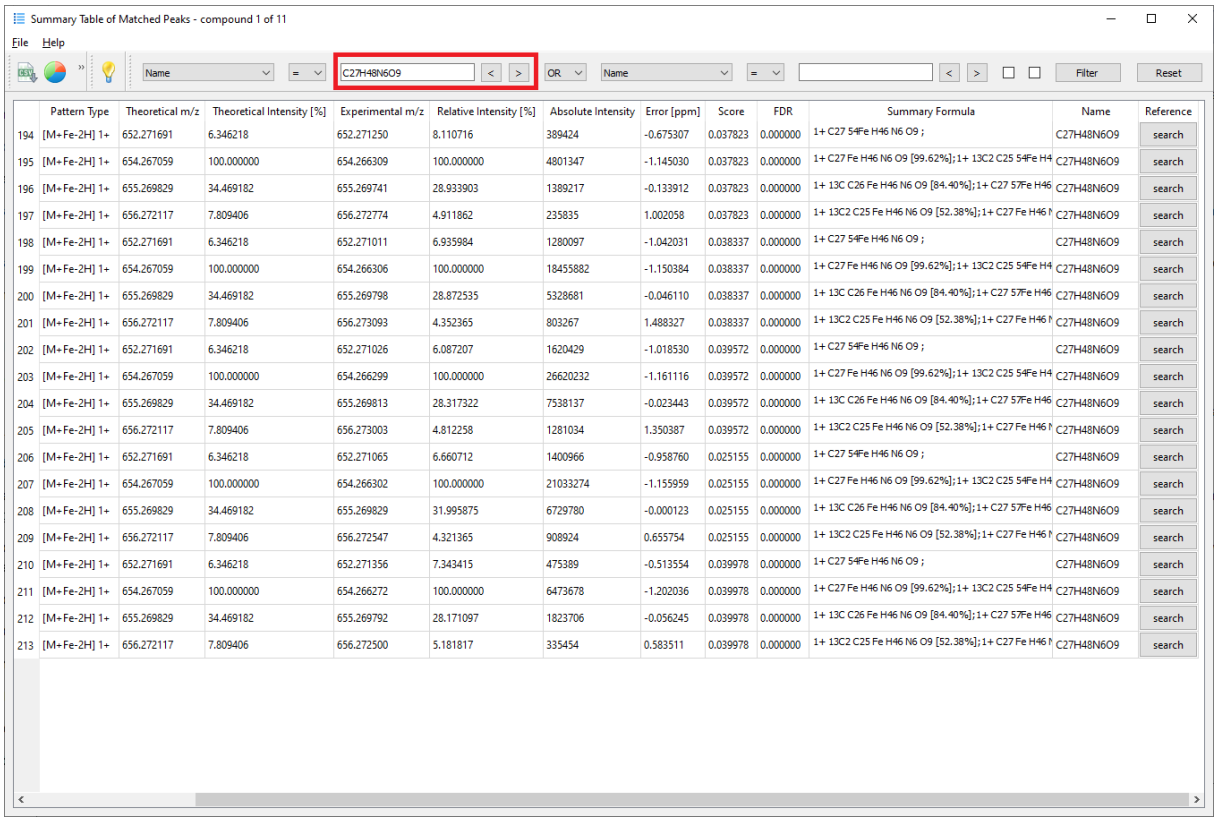


Figure 12.9 Tutorial 3: Configuration for identification of unknown compounds in LC-MS data.

Similarly to the Tutorial 2, the compounds can be browsed in the Summary Table of Matched Peaks while their extracted ion chromatograms can be directly seen in the Chromatogram window. The difference is that the formulas of compounds are automatically suggested instead of the names of compounds.



	Pattern Type	Theoretical m/z	Theoretical Intensity [%]	Experimental m/z	Relative Intensity [%]	Absolute Intensity	Error [ppm]	Score	FDR	Summary Formula	Name	Reference
194	[M+Fe-2H] 1+	652.271691	6.346218	652.271250	8.110716	389424	-0.675307	0.037823	0.000000	1+ C27 54Fe H46 N6 O9 ;	C27H48N6O9	search
195	[M+Fe-2H] 1+	654.267059	100.000000	654.266309	100.000000	4801347	-1.145030	0.037823	0.000000	1+ C27 Fe H46 N6 O9 [99.62%];1+ 13C2 C25 54Fe H4	C27H48N6O9	search
196	[M+Fe-2H] 1+	655.269829	34.469182	655.269741	28.933903	1389217	-0.133912	0.037823	0.000000	1+ 13C C26 Fe H46 N6 O9 [84.40%];1+ C27 57Fe H46	C27H48N6O9	search
197	[M+Fe-2H] 1+	656.272117	7.809406	656.272774	4.911862	235835	1.002058	0.037823	0.000000	1+ 13C2 C25 Fe H46 N6 O9 [52.38%];1+ C27 Fe H46 I	C27H48N6O9	search
198	[M+Fe-2H] 1+	652.271691	6.346218	652.271011	6.935984	1280097	-1.042031	0.038337	0.000000	1+ C27 54Fe H46 N6 O9 ;	C27H48N6O9	search
199	[M+Fe-2H] 1+	654.267059	100.000000	654.266306	100.000000	18455882	-1.150384	0.038337	0.000000	1+ C27 Fe H46 N6 O9 [99.62%];1+ 13C2 C25 54Fe H4	C27H48N6O9	search
200	[M+Fe-2H] 1+	655.269829	34.469182	655.269798	28.872535	5328681	-0.046110	0.038337	0.000000	1+ 13C C26 Fe H46 N6 O9 [84.40%];1+ C27 57Fe H46	C27H48N6O9	search
201	[M+Fe-2H] 1+	656.272117	7.809406	656.273093	4.352365	803267	1.488327	0.038337	0.000000	1+ 13C2 C25 Fe H46 N6 O9 [52.38%];1+ C27 Fe H46 I	C27H48N6O9	search
202	[M+Fe-2H] 1+	652.271691	6.346218	652.271026	6.087207	1620429	-1.018530	0.039572	0.000000	1+ C27 54Fe H46 N6 O9 ;	C27H48N6O9	search
203	[M+Fe-2H] 1+	654.267059	100.000000	654.266299	100.000000	26620232	-1.161116	0.039572	0.000000	1+ C27 Fe H46 N6 O9 [99.62%];1+ 13C2 C25 54Fe H4	C27H48N6O9	search
204	[M+Fe-2H] 1+	655.269829	34.469182	655.269813	28.317322	7538137	-0.023443	0.039572	0.000000	1+ 13C C26 Fe H46 N6 O9 [84.40%];1+ C27 57Fe H46	C27H48N6O9	search
205	[M+Fe-2H] 1+	656.272117	7.809406	656.273003	4.812258	1281034	1.350387	0.039572	0.000000	1+ 13C2 C25 Fe H46 N6 O9 [52.38%];1+ C27 Fe H46 I	C27H48N6O9	search
206	[M+Fe-2H] 1+	652.271691	6.346218	652.271065	6.660712	1400966	-0.958760	0.025155	0.000000	1+ C27 54Fe H46 N6 O9 ;	C27H48N6O9	search
207	[M+Fe-2H] 1+	654.267059	100.000000	654.266302	100.000000	21033274	-1.155959	0.025155	0.000000	1+ C27 Fe H46 N6 O9 [99.62%];1+ 13C2 C25 54Fe H4	C27H48N6O9	search
208	[M+Fe-2H] 1+	655.269829	34.469182	655.269829	31.995875	6729780	-0.000123	0.025155	0.000000	1+ 13C C26 Fe H46 N6 O9 [84.40%];1+ C27 57Fe H46	C27H48N6O9	search
209	[M+Fe-2H] 1+	656.272117	7.809406	656.272547	4.321365	908924	0.655754	0.025155	0.000000	1+ 13C2 C25 Fe H46 N6 O9 [52.38%];1+ C27 Fe H46 I	C27H48N6O9	search
210	[M+Fe-2H] 1+	652.271691	6.346218	652.271356	7.343415	475389	-0.513554	0.039978	0.000000	1+ C27 54Fe H46 N6 O9 ;	C27H48N6O9	search
211	[M+Fe-2H] 1+	654.267059	100.000000	654.266272	100.000000	6473678	-1.202036	0.039978	0.000000	1+ C27 Fe H46 N6 O9 [99.62%];1+ 13C2 C25 54Fe H4	C27H48N6O9	search
212	[M+Fe-2H] 1+	655.269829	34.469182	655.269792	28.171097	1823706	-0.056245	0.039978	0.000000	1+ 13C C26 Fe H46 N6 O9 [84.40%];1+ C27 57Fe H46	C27H48N6O9	search
213	[M+Fe-2H] 1+	656.272117	7.809406	656.272500	5.181817	335454	0.583511	0.039978	0.000000	1+ 13C2 C25 Fe H46 N6 O9 [52.38%];1+ C27 Fe H46 I	C27H48N6O9	search

Figure 12.10 Tutorial 3: Summary Table of Matched Peaks.

12.5 Tutorial 4: Dereplication in MSI data and fusion with histology image

Follow the instructions in Tutorial 1 and Tutorial 2. Modify the following parameters:
File - select an input imzML file.
Minimum Number of Spectra - the minimum number of single-pixel mass spectra in which a compound must be matched to be reported.

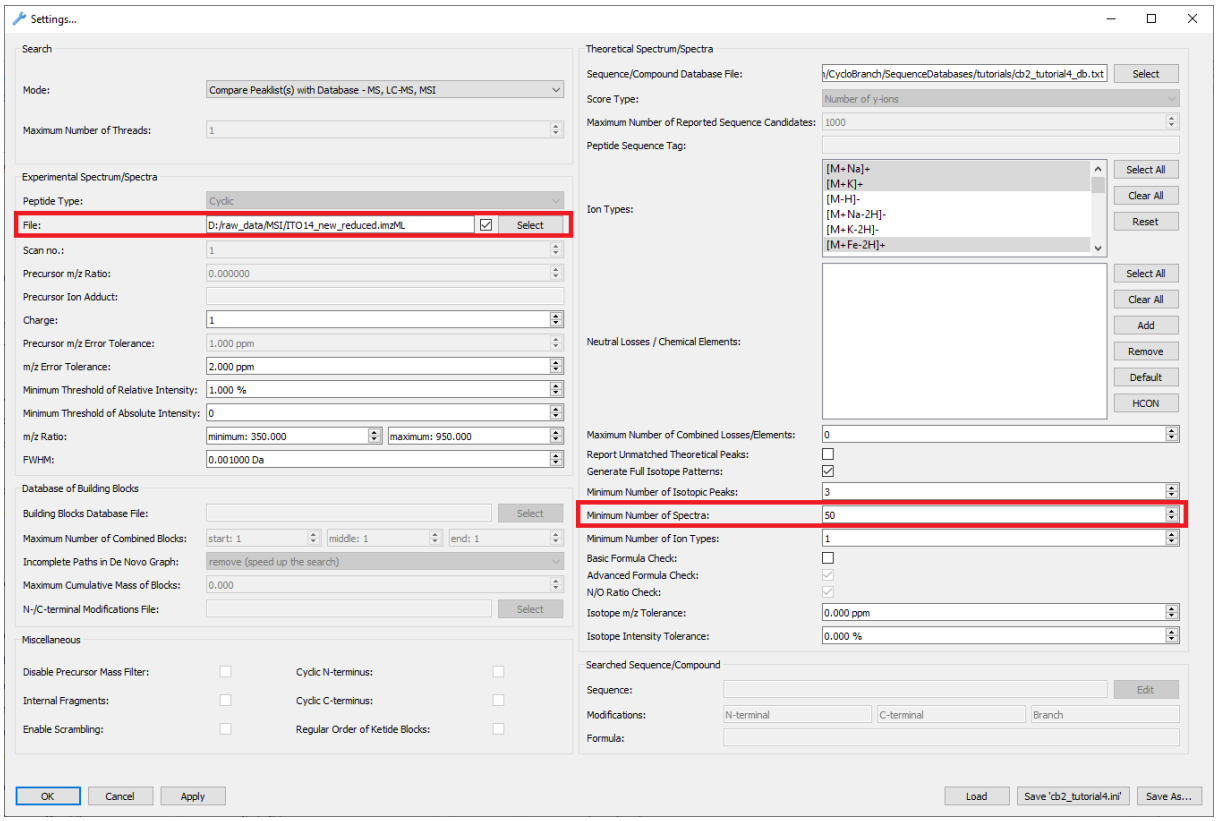
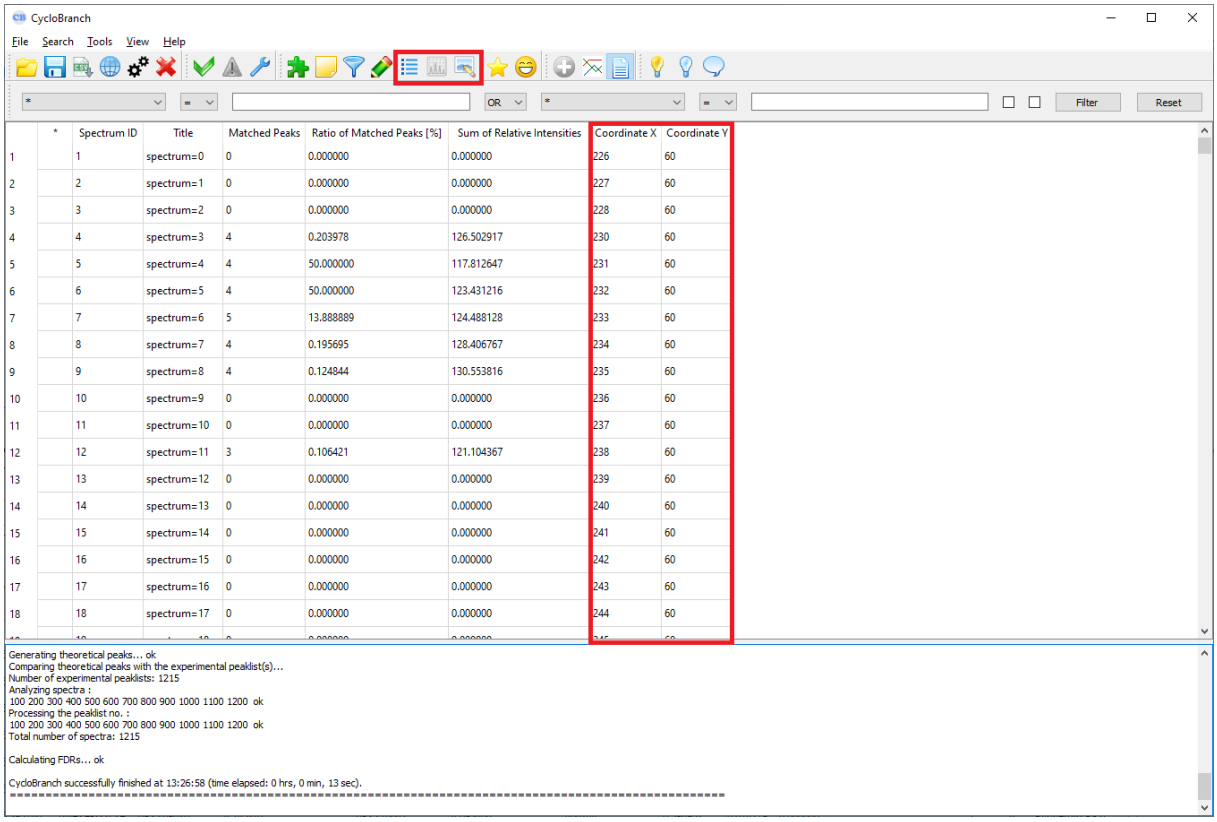


Figure 12.11 Tutorial 4: Configuration for dereplication in MSI data.

Every row in the output report includes coordinates [x, y] and corresponds to a single-pixel mass spectrum. Double-click on any row to display the spectrum from the corresponding pixel.



	*	Spectrum ID	Title	Matched Peaks	Ratio of Matched Peaks [%]	Sum of Relative Intensities	Coordinate X	Coordinate Y
1		1	spectrum=0	0	0.000000	0.000000	226	60
2		2	spectrum=1	0	0.000000	0.000000	227	60
3		3	spectrum=2	0	0.000000	0.000000	228	60
4		4	spectrum=3	4	0.203978	126.502917	230	60
5		5	spectrum=4	4	50.000000	117.812647	231	60
6		6	spectrum=5	4	50.000000	123.431216	232	60
7		7	spectrum=6	5	13.888889	124.488128	233	60
8		8	spectrum=7	4	0.195695	128.406767	234	60
9		9	spectrum=8	4	0.124844	130.553816	235	60
10		10	spectrum=9	0	0.000000	0.000000	236	60
11		11	spectrum=10	0	0.000000	0.000000	237	60
12		12	spectrum=11	3	0.106421	121.104367	238	60
13		13	spectrum=12	0	0.000000	0.000000	239	60
14		14	spectrum=13	0	0.000000	0.000000	240	60
15		15	spectrum=14	0	0.000000	0.000000	241	60
16		16	spectrum=15	0	0.000000	0.000000	242	60
17		17	spectrum=16	0	0.000000	0.000000	243	60
18		18	spectrum=17	0	0.000000	0.000000	244	60

Figure 12.12 Tutorial 4: Search process completed.

In the main window, click on Tools->Image Fusion. In the Image Fusion tool, click on **File->Open**, select **Optical Image**, and click **Ok**. Then select the image file which was used during the acquisition of your data. The optical image is automatically correlated with pixels (or squares) corresponding to the individual mass spectra. In some cases, the correlation of data may be vendor specific, see the details here.

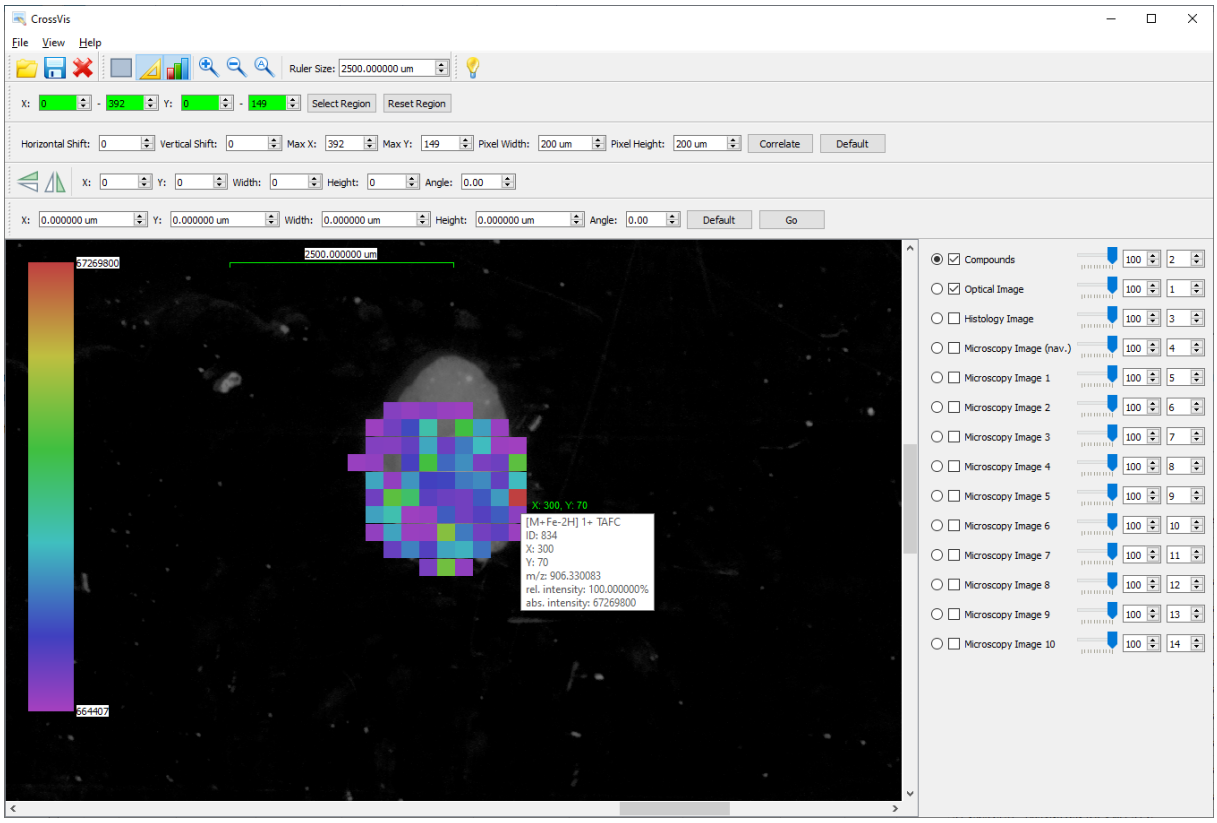
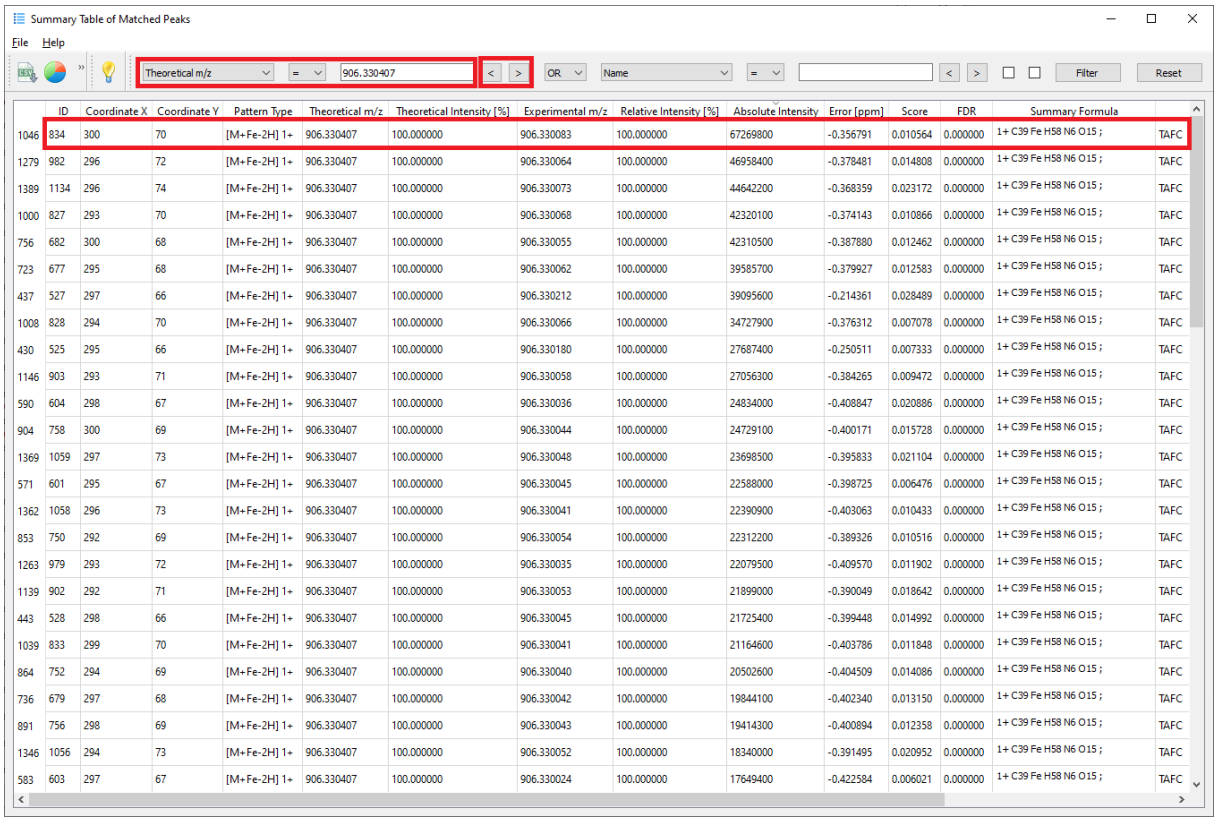


Figure 12.13 Tutorial 4: Fusion of optical image with identified compounds.

The color of every square represents the sum of absolute/relative intensities of all matched peaks in the correspoding mass spectrum by default. However, the Image Fusion tool is interactively-connected with Summary Table of Matched Peaks. To display a single compound or to browse spatial distributions of all identified compounds, open the Tools->Summary Table of Matched Peaks and filter the list of all matched peaks by the name of a compound or by theoretical m/z ratio. Start typing into the text field and the names of identified compounds are automatically suggested. Use the arrows to browse the identified compounds individually.



	ID	Coordinate X	Coordinate Y	Pattern Type	Theoretical m/z	Theoretical Intensity [%]	Experimental m/z	Relative Intensity [%]	Absolute Intensity	Error [ppm]	Score	FDR	Summary Formula	
1046	834	300	70	[M+Fe-2H] 1+	906.330407	100.000000	906.330083	100.000000	67269800	-0.356791	0.010564	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1279	982	296	72	[M+Fe-2H] 1+	906.330407	100.000000	906.330064	100.000000	46958400	-0.378481	0.014808	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1389	1134	296	74	[M+Fe-2H] 1+	906.330407	100.000000	906.330073	100.000000	44642200	-0.368359	0.023172	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1000	827	293	70	[M+Fe-2H] 1+	906.330407	100.000000	906.330068	100.000000	42320100	-0.374143	0.010866	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
756	682	300	68	[M+Fe-2H] 1+	906.330407	100.000000	906.330055	100.000000	42310500	-0.387880	0.012462	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
723	677	295	68	[M+Fe-2H] 1+	906.330407	100.000000	906.330062	100.000000	39585700	-0.379927	0.012583	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
437	527	297	66	[M+Fe-2H] 1+	906.330407	100.000000	906.330212	100.000000	39095600	-0.214361	0.028489	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1008	828	294	70	[M+Fe-2H] 1+	906.330407	100.000000	906.330066	100.000000	34727900	-0.376312	0.007078	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
430	525	295	66	[M+Fe-2H] 1+	906.330407	100.000000	906.330180	100.000000	27687400	-0.250511	0.007333	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1146	903	293	71	[M+Fe-2H] 1+	906.330407	100.000000	906.330058	100.000000	27056300	-0.384265	0.009472	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
590	604	298	67	[M+Fe-2H] 1+	906.330407	100.000000	906.330036	100.000000	24834000	-0.408847	0.020886	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
904	758	300	69	[M+Fe-2H] 1+	906.330407	100.000000	906.330044	100.000000	24729100	-0.400171	0.015728	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1369	1059	297	73	[M+Fe-2H] 1+	906.330407	100.000000	906.330048	100.000000	23698500	-0.395833	0.021104	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
571	601	295	67	[M+Fe-2H] 1+	906.330407	100.000000	906.330045	100.000000	22588000	-0.398725	0.006476	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1362	1058	296	73	[M+Fe-2H] 1+	906.330407	100.000000	906.330041	100.000000	22390900	-0.403063	0.010433	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
853	750	292	69	[M+Fe-2H] 1+	906.330407	100.000000	906.330054	100.000000	22312200	-0.389326	0.010516	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1263	979	293	72	[M+Fe-2H] 1+	906.330407	100.000000	906.330035	100.000000	22079500	-0.409570	0.011902	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1139	902	292	71	[M+Fe-2H] 1+	906.330407	100.000000	906.330053	100.000000	21899000	-0.390049	0.018642	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
443	528	298	66	[M+Fe-2H] 1+	906.330407	100.000000	906.330045	100.000000	21725400	-0.399448	0.014992	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1039	833	299	70	[M+Fe-2H] 1+	906.330407	100.000000	906.330041	100.000000	21164600	-0.403786	0.011848	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
864	752	294	69	[M+Fe-2H] 1+	906.330407	100.000000	906.330040	100.000000	20502600	-0.404509	0.014086	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
736	679	297	68	[M+Fe-2H] 1+	906.330407	100.000000	906.330042	100.000000	19844100	-0.402340	0.013150	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
891	756	298	69	[M+Fe-2H] 1+	906.330407	100.000000	906.330043	100.000000	19414300	-0.400894	0.012358	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
1346	1056	294	73	[M+Fe-2H] 1+	906.330407	100.000000	906.330052	100.000000	18340000	-0.391495	0.020952	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC
583	603	297	67	[M+Fe-2H] 1+	906.330407	100.000000	906.330024	100.000000	17649400	-0.422584	0.006021	0.000000	1+ C39 Fe H58 N6 O15 ;	TAFC

Figure 12.14 Tutorial 4: Summary Table of Matched Peaks.

If you would like to make a fusion with another image (e.g. a histology image), click on **File->Open**, select **Histology Image**, and click **Ok**. Then select an image in ∗.jpg, ∗.jpeg, ∗.png, ∗.tif, ∗.tiff, ∗.bmp or ∗.gif file format. Click on **View->Show Selection** to see a bounding box around the opened histology image. Make sure that **Histology Image** layer is selected in the list of layers. The histology image must be correlated with optical image manually. To make the fusion, move the histology image by mouse to overlap the analyzed sample on the optical image. If needed, adjust the size and rotation angle of the histology image on the corresponding toolbar (the active toolbar is green). Optionally, adjust the transparency and order of any layer on the right panel. See more details here.

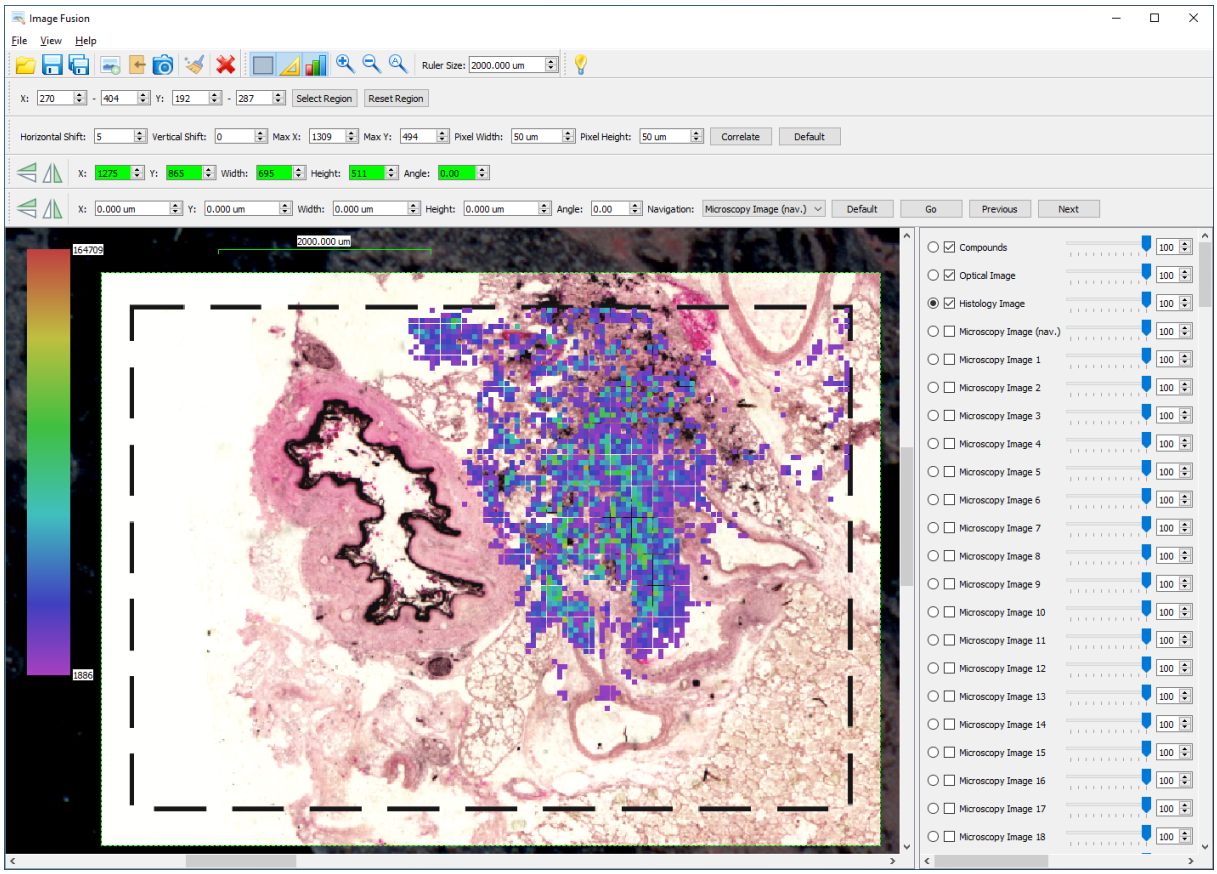


Figure 12.15 Tutorial 4: fusion of histology image (white) with an optical image (black).

12.6 Tutorial 5: Identification of unknown compounds in MSI data

In this example, any database of compounds is not used but the molecular formulas of compounds are generated from an input list of chemical elements e.g. H, C, O, N, S, and P. Because the search space grows exponentially with increasing m/z value and the number of input elements, carefully configure the following parameters to limit the number of false-positive formulas reported, to speed-up the search process and to save the computer's main memory:

Mode - select Compound Search - MS, LC-MS, MSI.

File - select an input imzML file. The file commonly contains profile mass spectra. In this case, make sure that you have installed `OpenMS`. Enable the checkbox, if you would like to visualize the profile mass spectra when the data-processing is completed (note: the input file must include profile spectra; the extraction may be time-consuming; and the visualization must be enabled in the spectrum detail window).

Charge - the maximum charge of generated theoretical peaks. If the value is equal to 2, then the theoretical peaks of singly and doubly charged ions are generated. Use -1 for negative mode.

m/z Error Tolerance - define m/z error tolerance e.g. 2 ppm. You can use bigger value e.g. 5 or 10 ppm but in this case define also the parameter **Isotope m/z Tolerance**.

Minimum Threshold of Relative Intensity [%] - use e.g. 3 or 5% for an initial check of a dataset. Check the results and optionally reduce to 1% or lower value. Use 0 to disable this feature. The value is used together with **Minimum Threshold of Absolute Intensity**.

Minimum Threshold of Absolute Intensity - use bigger value (e.g. 100000) for an initial check of a dataset and then optionally reduce to lower value. Use 0 to disable this feature. The value is used together with **Minimum Threshold of Relative Intensity**.

m/z Ratio - define the range of m/z values in which the compounds should be detected. If a narrow interval is used (e.g. 100 or 200 Da), the search process is faster. If a wide interval (e.g. 1000 Da) is used, the searching is slower.

FWHM - see the profile peaks in your spectra and estimate the full width at half maximum (i.e. the width of a representative peak at half of his height; the value corresponds to m/z ratio of a peak divided by its resolution). The theoretical peaks are split or merged using FWHM value prior they are matched with experimental peaks.
Ion Types - select the types of ions to be searched. Remember that for negative ions (e.g. [M-H]-), the value of charge must be negative (i.e. -1). For an initial check of a dataset, use only one or a very small number of ions.
Neutral Losses / **Chemical Elements** - Define an input list of chemical elements. Even if any custom element can be added, use the smallest number of elements as possible. Use the colon to limit the maximum number of occurrences of elements like sulfur or phosphorus (e.g. S:2 or P:1; double-click on an element to edit this value). If the ions like [M+Na]+, [M+K]+ and [M+Fe-2H]+ are searched, do not add the elements Na, K, and Fe, respectively. They are used automatically. Use the **HCON** button to quickly define the basic elements H, C, O, N, S, and P.
Maximum Number of Combined Losses / **Elements** - define the maximum number of elements in a molecule (e.g. 150 or 200). Example, the ion [C39H60N6O15+H]+ has 39+60+6+15+1=121 elements.
Generate Full Isotope Patterns - enable the checkbox to generate the full isotopic patterns of compounds (the parameter FWHM must be estimated correctly).
Minimum Number of Isotopic Peaks - define the minimum number of peaks which must be matched in an isotopic pattern to report a given compound. The feature **Generate Full Isotope Patterns** must be enabled. Use e.g. 2 if a big value of **Minimum Threshold of Relative Intensity [%]** is used.
Minimum Number of Spectra - the minimum number of pixels in which a compound must be matched to be reported.
Basic Formula Check - apply Senior's filtering rules to remove unlikely molecular formulas. See https://doi.org/10.1186/1471-2105-8-105 for more details.
Advanced Formula Check - advanced filtering rules are applied if molecular formulas of compounds are generated. See the tables 1, 2, and 3 in https://doi.org/10.1186/1471-2105-8-105 for more details.
N/O Ratio Check - check if the number of nitrogen atoms is less or equal to the number of oxygen atoms.
Isotope m/z Tolerance - the maximum m/z error of difference between an isotopic peak and the most intense peak in an experimental and theoretical isotopic pattern (0 = disabled); see the definition of Isotope m/z Tolerance.
Isotope Intensity Tolerance - the maximum error tolerance of intensities of matched isotopes (e.g. 10%); see the definition of Isotope Intensity Tolerance.

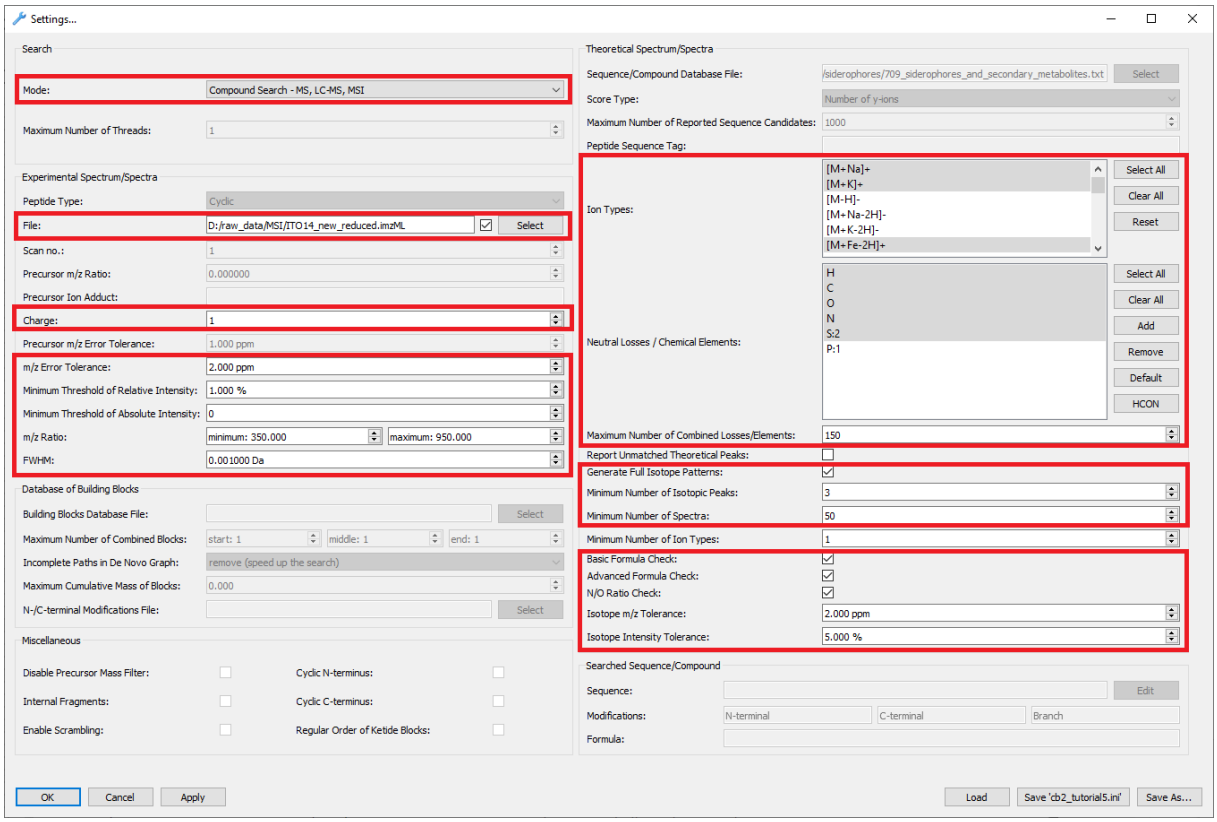
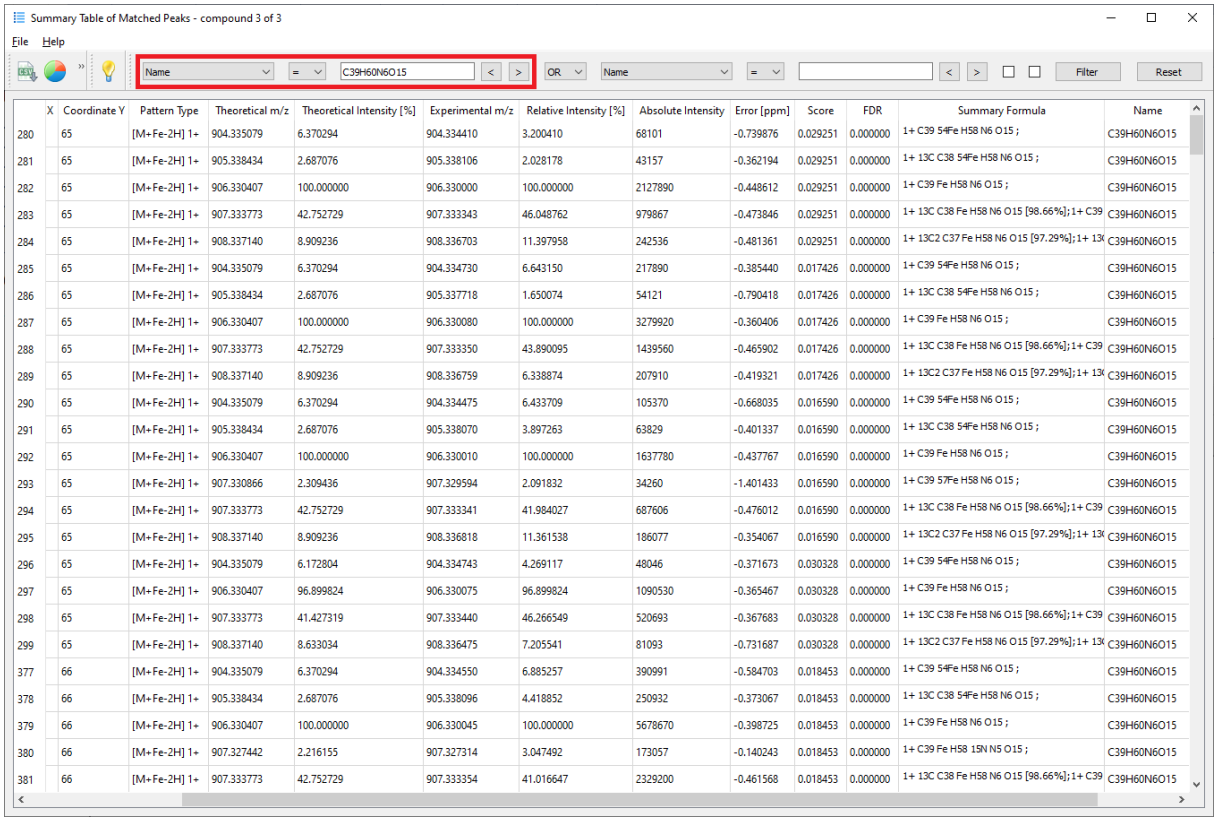


Figure 12.16 Tutorial 5: Configuration for identification of unknown compounds in MSI data.

The output report and the visualization of results are described in Tutorial 4. The only difference is that the molecular

formulas of compounds are suggested in the Summary Table of Matched Peaks instead of names of compounds.



	X	Coordinate Y	Pattern Type	Theoretical m/z	Theoretical Intensity [%]	Experimental m/z	Relative Intensity [%]	Absolute Intensity	Error [ppm]	Score	FDR	Summary Formula	Name
280		65	[M+Fe-2H] 1+	904.335079	6.370294	904.334410	3.200410	68101	-0.739876	0.029251	0.000000	1+ C39 54Fe H58 N6 O15 ;	C39H60N6O15
281		65	[M+Fe-2H] 1+	905.338434	2.687076	905.338106	2.028178	43157	-0.362194	0.029251	0.000000	1+ 13C C38 54Fe H58 N6 O15 ;	C39H60N6O15
282		65	[M+Fe-2H] 1+	906.330407	100.000000	906.330000	100.000000	2127890	-0.448612	0.029251	0.000000	1+ C39 Fe H58 N6 O15 ;	C39H60N6O15
283		65	[M+Fe-2H] 1+	907.333773	42.752729	907.333343	46.048762	979867	-0.473846	0.029251	0.000000	1+ 13C C38 Fe H58 N6 O15 [98.66%];1+ C39	C39H60N6O15
284		65	[M+Fe-2H] 1+	908.337140	8.909236	908.336703	11.397958	242536	-0.481361	0.029251	0.000000	1+ 13C2 C37 Fe H58 N6 O15 [97.29%];1+ 13	C39H60N6O15
285		65	[M+Fe-2H] 1+	904.335079	6.370294	904.334730	6.643150	217890	-0.385440	0.017426	0.000000	1+ C39 54Fe H58 N6 O15 ;	C39H60N6O15
286		65	[M+Fe-2H] 1+	905.338434	2.687076	905.337718	1.650074	54121	-0.790418	0.017426	0.000000	1+ 13C C38 54Fe H58 N6 O15 ;	C39H60N6O15
287		65	[M+Fe-2H] 1+	906.330407	100.000000	906.330080	100.000000	3279920	-0.360406	0.017426	0.000000	1+ C39 Fe H58 N6 O15 ;	C39H60N6O15
288		65	[M+Fe-2H] 1+	907.333773	42.752729	907.333350	43.890095	1439560	-0.465902	0.017426	0.000000	1+ 13C C38 Fe H58 N6 O15 [98.66%];1+ C39	C39H60N6O15
289		65	[M+Fe-2H] 1+	908.337140	8.909236	908.336759	6.338874	207910	-0.419321	0.017426	0.000000	1+ 13C2 C37 Fe H58 N6 O15 [97.29%];1+ 13	C39H60N6O15
290		65	[M+Fe-2H] 1+	904.335079	6.370294	904.334475	6.433709	105370	-0.668035	0.016590	0.000000	1+ C39 54Fe H58 N6 O15 ;	C39H60N6O15
291		65	[M+Fe-2H] 1+	905.338434	2.687076	905.338070	3.897263	63829	-0.401337	0.016590	0.000000	1+ 13C C38 54Fe H58 N6 O15 ;	C39H60N6O15
292		65	[M+Fe-2H] 1+	906.330407	100.000000	906.330010	100.000000	1637780	-0.437767	0.016590	0.000000	1+ C39 Fe H58 N6 O15 ;	C39H60N6O15
293		65	[M+Fe-2H] 1+	907.330866	2.309436	907.329594	2.091832	34260	-1.401433	0.016590	0.000000	1+ C39 57Fe H58 N6 O15 ;	C39H60N6O15
294		65	[M+Fe-2H] 1+	907.333773	42.752729	907.333341	41.984027	687606	-0.476012	0.016590	0.000000	1+ 13C C38 Fe H58 N6 O15 [98.66%];1+ C39	C39H60N6O15
295		65	[M+Fe-2H] 1+	908.337140	8.909236	908.336818	11.361538	186077	-0.354067	0.016590	0.000000	1+ 13C2 C37 Fe H58 N6 O15 [97.29%];1+ 13	C39H60N6O15
296		65	[M+Fe-2H] 1+	904.335079	6.172804	904.334743	4.269117	48046	-0.371673	0.030328	0.000000	1+ C39 54Fe H58 N6 O15 ;	C39H60N6O15
297		65	[M+Fe-2H] 1+	906.330407	96.899824	906.330075	96.899824	1090530	-0.365467	0.030328	0.000000	1+ C39 Fe H58 N6 O15 ;	C39H60N6O15
298		65	[M+Fe-2H] 1+	907.333773	41.427319	907.333440	46.266549	520693	-0.367683	0.030328	0.000000	1+ 13C C38 Fe H58 N6 O15 [98.66%];1+ C39	C39H60N6O15
299		65	[M+Fe-2H] 1+	908.337140	8.633034	908.336475	7.205541	81093	-0.731687	0.030328	0.000000	1+ 13C2 C37 Fe H58 N6 O15 [97.29%];1+ 13	C39H60N6O15
377		66	[M+Fe-2H] 1+	904.335079	6.370294	904.334550	6.885257	390991	-0.584703	0.018453	0.000000	1+ C39 54Fe H58 N6 O15 ;	C39H60N6O15
378		66	[M+Fe-2H] 1+	905.338434	2.687076	905.338096	4.418852	250932	-0.373067	0.018453	0.000000	1+ 13C C38 54Fe H58 N6 O15 ;	C39H60N6O15
379		66	[M+Fe-2H] 1+	906.330407	100.000000	906.330045	100.000000	5678670	-0.398725	0.018453	0.000000	1+ C39 Fe H58 N6 O15 ;	C39H60N6O15
380		66	[M+Fe-2H] 1+	907.327442	2.216155	907.327314	3.047492	173057	-0.140243	0.018453	0.000000	1+ C39 Fe H58 15N N5 O15 ;	C39H60N6O15
381		66	[M+Fe-2H] 1+	907.333773	42.752729	907.333354	41.016647	2329200	-0.461568	0.018453	0.000000	1+ 13C C38 Fe H58 N6 O15 [98.66%];1+ C39	C39H60N6O15

Figure 12.17 Tutorial 5: Summary Table of Matched Peaks.

12.7 Tutorial 6: Annotate MS/MS spectrum of a metabolite

The following tutorial shows how to perform the annotation of peaks corresponding to fragment ions of a small molecule in a single MS/MS spectrum. If you would like to annotate single MS/MS spectra of (non-)ribosomal peptides and siderophores, see the tutorials for CycloBranch (1.x). See also the extra pages for linear, cyclic, branched, and branch-cyclic peptides; the building blocks editor, sequence/compound database editor, modifications editor, and draw peptide tool.
In this example, we annotate MS/MS spectrum of [M+H]+ ion of pyochelin with m/z 325.06733.

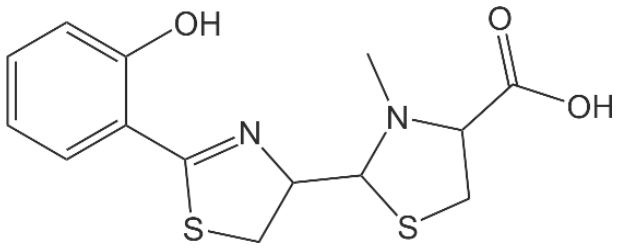


Figure 12.18 Tutorial 6: Structure of pyochelin.

The following parameters in the Settings are used:
Mode - select Compare Peaklist(s) with Spectrum of Searched Sequence - MS/MS.
Peptide Type - select **Other**.
File - select an input file with MS/MS spectrum. Because the mode **Compare Peaklist(s) with Spectrum of Searched Sequence - MS/MS** is used, the file can also include multiple MS/MS spectra. In such a case, all the spectra in the file are compared with the same theoretical spectrum and listed in the output report. (Note: This

feature is not supported if the modes **De Novo Search Engine - MS/MS** and **Compare Peaklist with Database - MS/MS** are used. These modes require only one MS/MS spectrum. If a file contains multiple spectra, the spectrum can be selected using the parameter **Scan no.** which determines the order of MS/MS spectrum in the file.)
Precursor m/z Ratio - define the m/z ratio of precursor ion. The value is not read automatically from a file.
Precursor Ion Adduct - Keep empty for [M+H]+ ion. For [M+Na]+, [M+K]+, and [M+Fe-2H]+ ions use the values Na, K, and FeH-2, respectively. See more details here.
Charge - the maximum charge of generated theoretical peaks. If the value is equal to 2, then the theoretical peaks of singly and doubly charged ions are generated. Use -1 for negative mode.
Precursor m/z Error Tolerance - define m/z error tolerance of precursor ion (the value can be the same as **m/z Error Tolerance**).
m/z Error Tolerance - define m/z error tolerance (e.g. 2 ppm).
Minimum Threshold of Relative Intensity [%] - use e.g. 3 or 5% for an initial check of a spectrum. Check the results and optionally reduce to 1% or lower value. Use 0 to disable this feature. The value is used together with **Minimum Threshold of Absolute Intensity**.
Minimum Threshold of Absolute Intensity - in this mode, you can use 0 to disable this feature. The value is used together with **Minimum Threshold of Relative Intensity**.
Minimum m/z Ratio - define the minimum m/z ratio. In this mode, it is advantageous to keep a window between the singly charged precursor ion and the minimum m/z ratio as small as possible to speed-up the search process. In our example, the window is small i.e. 325.06733 - 100 = 225.06733 and thus the number of generated combinations of elements which may drop out from the precursor ion is also small. See the parameter **Neutral Losses** / **Chemical Elements**.
FWHM - see the profile peaks in your spectra and estimate the full width at half maximum (i.e. the width of a representative peak at half of his height; the value corresponds to m/z ratio of a peak divided by its resolution). The theoretical peaks are split or merged using FWHM value prior they are matched with experimental peaks. The value is used only if the parameter **Generate Full Isotope Patterns** is enabled.
Neutral Losses / **Chemical Elements** - Define an input list of chemical elements which can drop out from the precursor ion. Use the colon to limit the maximum number of occurrences of elements like sulfur or phosphorus (e.g. S:2 or P:1; double-click on an element to edit this value). Use the **HCON** button to quickly define the basic elements H, C, O, N, S, and P.
Maximum Number of Combined Losses / **Elements** - define the maximum number of combined elements. In our example, we fragment the ion [C14H16N2O3S2+H]+. So, you do not have to use a bigger value than 14+16+2+3+2+1=38.
Generate Full Isotope Patterns - enable the checkbox to generate the full isotopic patterns of fragment ions (the parameter FWHM must be estimated correctly). In this mode, enable the feature only if the isolation window was wide enough and you expect the presence of isotopic peaks of fragment ions in the MS/MS spectrum. You can disable this feature to generate only the theoretical peaks of monoisotopic ions and to speed-up the annotation.
Minimum Number of Isotopic Peaks - define the minimum number of peaks which must be matched in an isotopic pattern to report a match of a fragment ion. The feature **Generate Full Isotope Patterns** must be enabled.
Basic Formula Check - the Senior's filtering rules are applied to the combinations of elements which may drop out from the precursor ion. See https://doi.org/10.1186/1471-2105-8-105 for more details.
Formula - define the formula corresponding to the precursor ion (neutral state; no adducts).

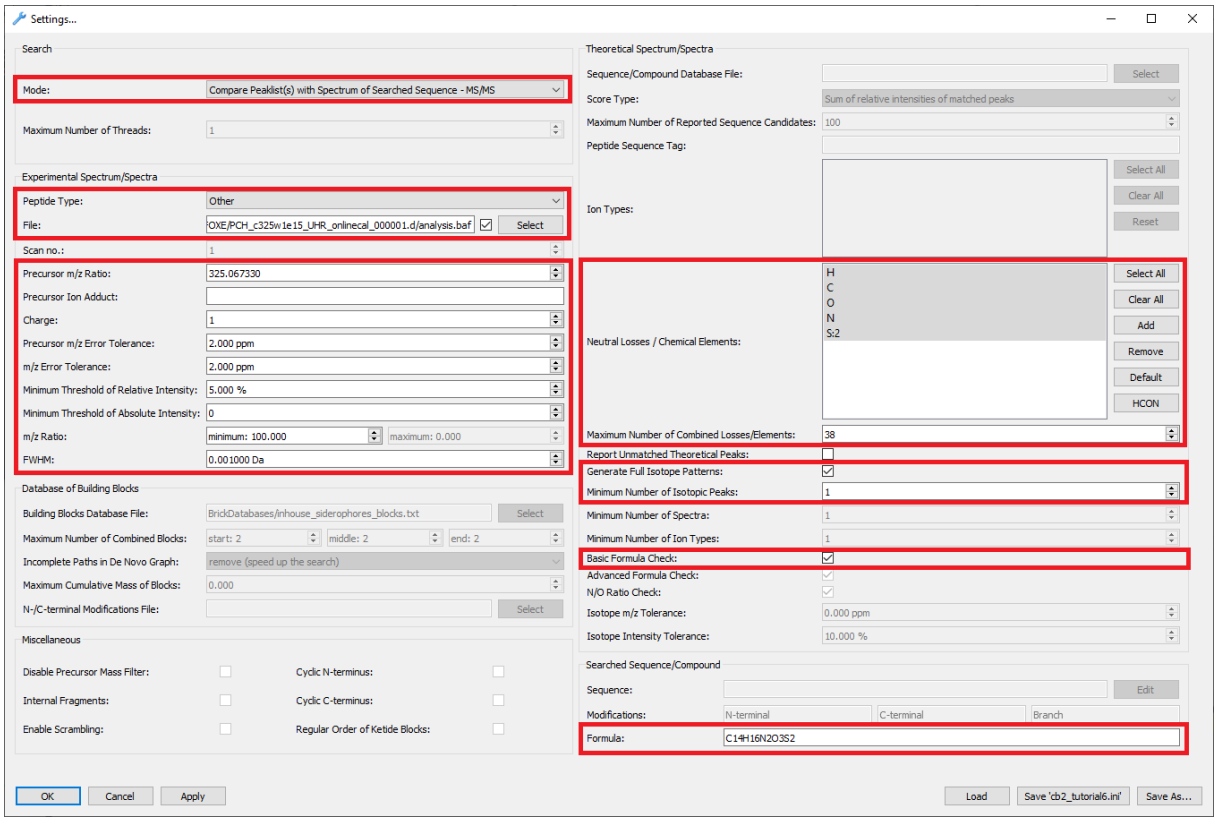


Figure 12.19 Tutorial 6: Configuration for annotation of MS/MS spectrum of a small molecule.

Click **OK**. In the main window select **Search->Run** to process the dataset. Double-click on the row in the output report to see the details of identification.

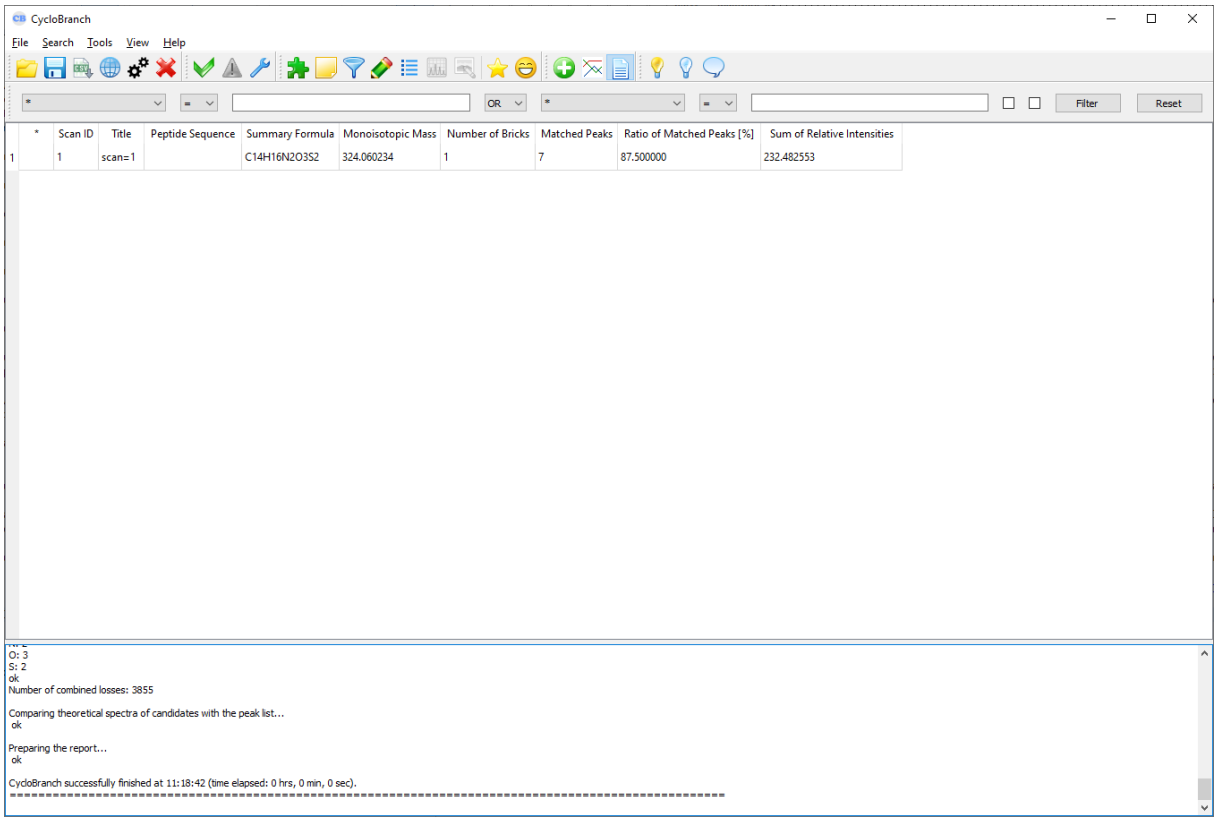
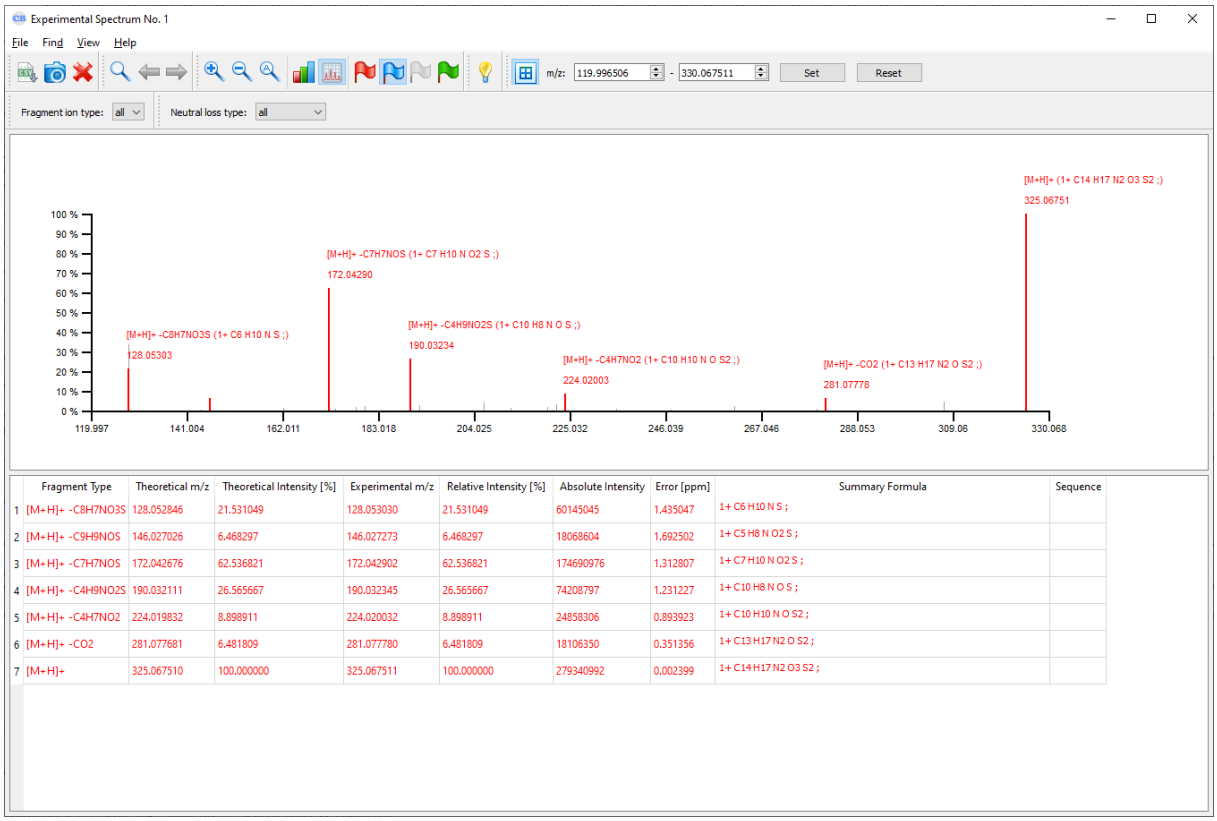


Figure 12.20 Tutorial 6: Search process completed.

In the detail window you can see the annotations of fragment ions, visualize the profile spectra (if they were included in the input file), etc.



	Fragment Type	Theoretical m/z	Theoretical Intensity [%]	Experimental m/z	Relative Intensity [%]	Absolute Intensity	Error [ppm]	Summary Formula	Sequence
1	[M+H]+ -C8H7NO3S	128.052846	21.531049	128.053030	21.531049	60145045	1.435047	1+ C6 H10 N S ;	
2	[M+H]+ -C9H9NOS	146.027026	6.468297	146.027273	6.468297	18068604	1.692502	1+ C5 H8 N O2 S ;	
3	[M+H]+ -C7H7NOS	172.042676	62.536821	172.042902	62.536821	174690976	1.312807	1+ C7 H10 N O2 S ;	
4	[M+H]+ -C4H9NO2S	190.032111	26.565667	190.032345	26.565667	74208797	1.231227	1+ C10 H8 N O S ;	
5	[M+H]+ -C4H7NO2	224.019832	8.898911	224.020032	8.898911	24858306	0.893923	1+ C10 H10 N O S2 ;	
6	[M+H]+ -CO2	281.077681	6.481809	281.077780	6.481809	18106350	0.351356	1+ C13 H17 N2 O S2 ;	
7	[M+H]+	325.067510	100.000000	325.067511	100.000000	279340992	0.002399	1+ C14 H17 N2 O3 S2 ;	

Figure 12.21 Tutorial 6: Detail of MS/MS spectrum.

Chapter 13

Hierarchical Index

13.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

cBrick . . . 191
cBricksDatabase . . . 199
cCandidate . . . 221
cCoordinateInfo . . . 251
cDefaultNeutralLosses . . . 254
cDeNovoGraph . . . 258
cDeNovoGraphNode . . . 262
cEdge . . . 268
cEnvelopeSummaryTableKeyMS_comp . . . 269
cFormulaPredictorRowItem . . . 273
cFragmentIons . . . 275
cFragmentIonType . . . 278
cGlobalPreferences . . . 279
cImzML . . . 304
cImzMLItem . . . 306
cIonLabel . . . 307
cIonSummaryTableKeyMS_comp . . . 307
cIonType . . . 308
cIsotopeMap . . . 309
cLifMetadata . . . 312
cMzML . . . 332
compareStringBySize . . . 335
cParameters . . . 336
cPeak . . . 353
cPeakListSeries . . . 356
cPeaksList . . . 361
cPeriodicTableMap . . . 375
cPubchemCIDReader . . . 385
cPubchemNameReader . . . 387
cRowItemInfo . . . 390
cSequence . . . 391
cSequenceDatabase . . . 397
cSummaryFormula . . . 425
cSummaryTableKeyMS . . . 437
cSummaryTableKeyMSMS . . . 438
cSummaryTableKeyMSMS_comp . . . 439
cTablePosition . . . 439
cTheoreticalSpectrum . . . 440
cTheoreticalSpectrumList . . . 459
cTileInfo . . . 463

hash_cCandidate . . . 466
hash_cEdge . . . 466
hash_cIonLabel . . . 467
hintStructure . . . 468
isotope . . . 468
layerInfo . . . 468
matchedSeries . . . 470
microscopyLayerInfo . . . 471
neutralLoss . . . 472
nodeEdge . . . 473
QDialog
  cExportDialog . . . 270
  cFindDialog . . . 271
  cHTMLExportDialog . . . 285
  cImageWindowImportDialog . . . 290
  cPreferencesDialog . . . 377
  cPreferencesIonTypeDialog . . . 378
QGraphicsView
  cCalibrationChartScene . . . 216
  cChromatogramWindowWidget . . . 245
  cImageWindowWidget . . . 295
  cPatternSimulatorSceneWidget . . . 347
  cSpectrumSceneWidget . . . 420
QMainWindow
  cBricksDatabaseWidget . . . 210
  cCalibrationChart . . . 211
  cChromatogramWindow . . . 241
  cFormulaPredictor . . . 272
  cImageWindow . . . 286
  cMainWindow . . . 319
  cModificationsWidget . . . 324
  cMultipleDatasetsTableWidget . . . 329
  cPatternSimulatorWidget . . . 351
  cSequenceDatabaseWidget . . . 402
  cSpectrumDetailWidget . . . 409
  cSummaryPeaksTableWidget . . . 433
QMutex
  cCandidateSet . . . 239
  cIsotopePatternCache . . . 310
QObject
  cEventFilter . . . 270
  cSpectrumAnalyzerThreadMS1 . . . 404
  cSpectrumComparatorThreadMS1 . . . 405
  cSpectrumComparatorThreadMS2 . . . 406
  cSpectrumGeneratorThreadMS1 . . . 419
QRunnable
  cSpectrumAnalyzerThreadMS1 . . . 404
  cSpectrumComparatorThreadMS1 . . . 405
  cSpectrumComparatorThreadMS2 . . . 406
  cSpectrumGeneratorThreadMS1 . . . 419
QSortFilterProxyModel
  cBricksDatabaseProxyModel . . . 207
  cCalibrationChartProxyModel . . . 215
  cMainWindowProxyModel . . . 320
  cModificationsProxyModel . . . 322
  cMultipleDatasetsTableProxyModel . . . 328
  cSequenceDatabaseProxyModel . . . 400
  cSpectrumDetailProxyModel . . . 407

cSummaryPeaksTableProxyModel . . . 430
QStyledItemDelegate
cCheckBoxDelegate . . . 240
cComboBoxDelegate . . . 249
cHTMLDelegate . . . 284
cMultipleButtonDelegate . . . 326
cViewButtonDelegate . . . 464
QThread
cFormulaPredictorThread . . . 274
cGraphReaderThread . . . 281
cMainThread . . . 315
QWidget
cAboutWidget . . . 185
cBranchCyclicWidget . . . 186
cBranchedWidget . . . 189
cCyclicWidget . . . 252
cDefineCalibrationWidget . . . 255
cDrawPeptideWidget . . . 266
cFragmentIonsListWidget . . . 277
cGraphWidget . . . 282
cImageWindowTabElectron . . . 292
cImageWindowTabOptical . . . 293
cLinearWidget . . . 313
cNeutralLossesListWidget . . . 334
cParametersWidget . . . 344
cPreferencesTabDataProcessing . . . 380
cPreferencesTabDirectories . . . 381
cPreferencesTabIonTypes . . . 382
cPreferencesTabURLs . . . 384
cPubChemSearchWidget . . . 388
splitSite . . . 474
TRotationInfo . . . 474

Chapter 14

Class Index

14.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

cAboutWidget
: The widget representing the dialog 'About' . . . 185

cBranchCyclicWidget
: Visualization of a branch-cyclic peptide . . . 186

cBranchedWidget
: Visualization of a branched peptide . . . 189

cBrick
: The class representing a building block (brick) . . . 191

cBricksDatabase
: The class representing a database of building blocks (bricks) . . . 199

cBricksDatabaseProxyModel
: A proxy model for editor of database of building blocks . . . 207

cBricksDatabaseWidget
: An editor of building blocks database . . . 210

cCalibrationChart
: Calibration chart . . . 211

cCalibrationChartProxyModel
: A table proxy model . . . 215

cCalibrationChartScene
: The implementation of an isotopic pattern simulator (graphics) . . . 216

cCandidate
: The class representing a peptide sequence candidate . . . 221

cCandidateSet
: The class representing a set of peptide sequence candidates . . . 239

cCheckBoxDelegate
: A checkbox item delegate . . . 240

cChromatogramWindow
: Chromatogram window . . . 241

cChromatogramWindowWidget
: Chromatogram window widget . . . 245

cComboBoxDelegate
: A combobox item delegate . . . 249

cCoordinateInfo
: Store aggregated information about a coordinate . . . 251

cCyclicWidget
: Visualization of a cyclic peptide . . . 252

cDefaultNeutralLosses
: The class representing a vector of default neutral losses . . . 254

cDefineCalibrationWidget
: The widget for definition of calibration curve . . . 255

cDeNovoGraph
The class implementing the de novo graph . . . 258
cDeNovoGraphNode
The class representing a node in the de novo graph . . . 262
cDrawPeptideWidget
Draw a peptide from a sequence of building blocks . . . 266
cEdge
The structure representing an outgoing edge from a node . . . 268
cEnvelopeSummaryTableKeyMS_comp
Comparison object for cSummaryTableKeyMS (whole envelope) . . . 269
cEventFilter
An event filter . . . 270
cExportDialog
The implementation of a dialog to export spectrum/peptide image . . . 270
cFindDialog
The implementation of a find dialog . . . 271
cFormulaPredictor
Predicts the molecular formula from the exact mass . . . 272
cFormulaPredictorRowItem
An auxiliary structure representing a row in the output table . . . 273
cFormulaPredictorThread
The thread which searches molecular formulas corresponding to m/z value . . . 274
cFragmentIons
The detailed definitions of all supported fragment ion types . . . 275
cFragmentIonsListWidget
The widget for selection of fragment ion types . . . 277
cFragmentIonType
The struct for definition of a fragment ion type . . . 278
cGlobalPreferences
The class storing global preferences . . . 279
cGraphReaderThread
The class representing a thread which scans the de novo graph for peptide sequence candidates 281
cGraphWidget
The widget representing the dialog 'View->Graph' . . . 282
cHTMLDelegate
A delegate to display rich text . . . 284
cHTMLExportDialog
The implementation of HTML export dialog . . . 285
cImageWindow
Image window . . . 286
cImageWindowImportDialog
Import dialog in image window . . . 290
cImageWindowTabElectron
The class representing a tab used to import images from electron microscopy . . . 292
cImageWindowTabOptical
The class representing a tab used to import images from light microscopy . . . 293
cImageWindowWidget
Image window widget . . . 295
cImzML
ImzML parser . . . 304
cImzMLItem
A parsed imzML item . . . 306
cIonLabel
Auxiliary structure for visualization of labels of fragment ions . . . 307
cIonSummaryTableKeyMS_comp
Comparison object for cSummaryTableKeyMS (independent ions) . . . 307
cIonType
The struct for definition of a ion type . . . 308

cIsotopeMap
The class representing an isotope map . . . 309
cIsotopePatternCache
Isotope pattern cache (LRU cache is implemented) . . . 310
cLifMetadata
A structure to keep parsed lif metadata . . . 312
cLinearWidget
Visualization of a linear peptide . . . 313
cMainThread
The class representing a thread launched by the command 'Search->Run' . . . 315
cMainWindow
The class representing the main window of the application . . . 319
cMainWindowProxyModel
A table proxy model . . . 320
cModificationsProxyModel
A proxy model for editor of database of modifications . . . 322
cModificationsWidget
An editor of N-/C-terminal modifications database . . . 324
cMultipleButtonDelegate
A delegate to create multiple buttons . . . 326
cMultipleDatasetsTableProxyModel
A table proxy model . . . 328
cMultipleDatasetsTableWidget
Aggregated results from multiple datasets . . . 329
cMzML
MzML parser . . . 332
cNeutralLossesListWidget
The widget for selection of types of neutral losses . . . 334
compareStringBySize
Compare items in a string vector by size . . . 335
cParameters
The class storing parameters of the application . . . 336
cParametersWidget
The class representing a widget for editing parameters of the application . . . 344
cPatternSimulatorSceneWidget
The implementation of an isotopic pattern simulator (graphics) . . . 347
cPatternSimulatorWidget
The implementation of an isotopic pattern simulator (window) . . . 351
cPeak
A structure representing a peak in a mass spectrum . . . 353
cPeakListSeries
The class representing a series of peaklists . . . 356
cPeaksList
The class representing a peak list . . . 361
cPeriodicTableMap
The class representing a periodic table . . . 375
cPreferencesDialog
Preferences dialog . . . 377
cPreferencesIonTypeDialog
Preferences dialog . . . 378
cPreferencesTabDataProcessing
The class representing a tab used to define the data processing parameters . . . 380
cPreferencesTabDirectories
The class representing a tab of default directories . . . 381
cPreferencesTabIonTypes
The class representing a tab of ion types . . . 382
cPreferencesTabURLs
The class representing a tab of URLs . . . 384

cPubchemCIDReader
The functor used to get the CIDs of compounds with the same formula . . . 385
cPubchemNameReader
The functor used to get the names of compounds from CID . . . 387
cPubChemSearchWidget
The widget representing the PubChem search widget . . . 388
cRowItemInfo
An auxiliary row info structure . . . 390
cSequence
The class representing a sequence . . . 391
cSequenceDatabase
The class representing a database of sequences . . . 397
cSequenceDatabaseProxyModel
A proxy model for editor of database of sequences . . . 400
cSequenceDatabaseWidget
An editor of sequences database . . . 402
cSpectrumAnalyzerThreadMS1
The class representing a thread for an initial check of LC-MS and MSI spectra . . . 404
cSpectrumComparatorThreadMS1
The class representing a thread for comparison of a theoretical spectrum with a peak list (MS1 level) . . . 405
cSpectrumComparatorThreadMS2
The class representing a thread for comparison of a theoretical spectrum with a peak list (MS2 level) . . . 406
cSpectrumDetailProxyModel
A table proxy model . . . 407
cSpectrumDetailWidget
The class representing a widget for visualisation of a detail of a spectrum (opened after doubleclick on a row when results are reported) . . . 409
cSpectrumGeneratorThreadMS1
The class representing a thread for generating a theoretical spectrum (MS1 level) . . . 419
cSpectrumSceneWidget
The widget representing a 2D mass spectrum . . . 420
cSummaryFormula
The representation of a summary formula . . . 425
cSummaryPeaksTableProxyModel
A table proxy model . . . 430
cSummaryPeaksTableWidget
Summary table of matched peaks . . . 433
cSummaryTableKeyMS
Key to reduce the summary table when exporting statistics for MS and MSI data . . . 437
cSummaryTableKeyMSMS
Key to reduce the summary table when exporting statistics for MS/MS data . . . 438
cSummaryTableKeyMSMS_comp
Comparison object for cSummaryTableKeyMSMS . . . 439
cTablePosition
Position in QTableWidget . . . 439
cTheoreticalSpectrum
The class representing a theoretical mass spectrum . . . 440
cTheoreticalSpectrumList
The class representing lists of theoretical spectra . . . 459
cTileInfo
Description of an image tile . . . 463
cViewButtonDelegate
A delegate to create a reference view button . . . 464
hash_cCandidate
The structure defining a hash function of cCandidate . . . 466

hash_cEdge
The structure defining a hash function of cEdge 466
hash_clonLabel
The structure defining a hash function of clonLabel 467
hintStructure
An auxiliary structure for search hints 468
isotope
Representation of an isotope 468
layerInfo
Paramaters of a layer 468
matchedSeries
An auxiliary structure for visualisation of matched series of fragment ions 470
microscopyLayerInfo
Paramaters of a microscopy layer 471
neutralLoss
The structure for a detailed definition of a neutral loss 472
nodeEdge
An auxiliary structure to store relationship between a node and an edge 473
splitSite
The structure representing a splitting site of a cyclic peptide 474
TRotationInfo
An auxiliary structure representing a branch permutation of a branched peptide 474

Chapter 15

File Index

15.1 File List

Here is a list of all documented files with brief descriptions:

core/cBrick.h
The representation of a brick . . . 477
core/cBricksDatabase.h
The database of building blocks . . . 480
core/cCandidate.h
The representation of a peptide sequence candidate . . . 483
core/cCandidateSet.h
The representation of a set of peptide sequence candidates . . . 488
core/cDeNovoGraph.h
The implementation of the de novo graph . . . 489
core/cDeNovoGraphNode.h
The representation of a node in the de novo graph . . . 491
core/cFragmentIons.h
Basic definitions of fragment ions occurring in the mass spectra . . . 495
core/cGlobalPreferences.h
The representation of a set of peptide sequence candidates . . . 525
core/cImzML.h
ImzML parser . . . 527
core/cIsotopePatternCache.h
Isotope pattern cache (LRU cache is implemented) . . . 529
core/cMzML.h
MzML parser . . . 530
core/cParameters.h
Definitions of program parameters . . . 532
core/cPeak.h
The representation of a peak in a mass spectrum . . . 540
core/cPeakListSeries.h
The representation of a series of peaklists . . . 541
core/cPeaksList.h
The representation of a peak list . . . 543
core/cSequence.h
The representation of a sequence . . . 548
core/cSequenceDatabase.h
The database of sequences . . . 549
core/cSummaryFormula.h
The representation of a summary formula . . . 551
core/cTheoreticalSpectrum.h
The representation of a theoretical mass spectrum . . . 553
core/cTheoreticalSpectrumList.h
The representation of lists of theoretical spectra . . . 560

core/utilities.h
Auxiliary funtions and structures . . . 561
gui/cAboutWidget.h
The implementation of the dialog 'Help->About' . . . 580
gui/cBranchCyclicWidget.h
Visualization of a branch-cyclic peptide . . . 581
gui/cBranchedWidget.h
Visualization of a branched peptide . . . 584
gui/cBricksDatabaseProxyModel.h
A proxy model for editor of database of building blocks . . . 585
gui/cBricksDatabaseWidget.h
An editor of building blocks database . . . 586
gui/cCalibrationChart.h
Calibration chart . . . 589
gui/cCalibrationChartProxyModel.h
A table proxy model . . . 592
gui/cCalibrationChartScene.h
Calibration chart scene . . . 592
gui/cCheckBoxDelegate.h
A checkbox item delegate . . . 597
gui/cChromatogramWindow.h
Chromatogram window . . . 598
gui/cChromatogramWindowWidget.h
Chromatogram window widget . . . 600
gui/cComboBoxDelegate.h
A combobox item delegate . . . 610
gui/cCyclicWidget.h
Visualization of a cyclic peptide . . . 611
gui/cDefineCalibrationWidget.h
The widget for definition of calibration curve . . . 616
gui/cDrawPeptideWidget.h
Draw a peptide from a sequence of building blocks . . . 618
gui/cEventFilter.h
An event filter . . . 620
gui/cExportDialog.h
The implementation of a dialog to export spectrum/peptide image . . . 621
gui/cFindDialog.h
The implementation of a find dialog . . . 621
gui/cFormulaPredictor.h
Predicts the molecular formula from the exact mass . . . 623
gui/cFragmentIonsListWidget.h
The implementation of a widget where types of fragment ions generated in theoretical spectra are selected . . . 625
gui/cGraphWidget.h
The implementation of the dialog 'View->Graph' . . . 626
gui/cHTMLDelegate.h
A delegate to display rich text . . . 628
gui/cHTMLExportDialog.h
The implementation of HTML export dialog . . . 628
gui/cImageWindow.h
Image window . . . 630
gui/cImageWindowImportDialog.h
The implementation of import dialog in image window . . . 636
gui/cImageWindowTabElectron.h
The implementation of a tab used to import images from electron microscopy . . . 638
gui/cImageWindowTabOptical.h
The implementation of a tab used to import images from light microscopy . . . 639

gui/cImageWindowWidget.h
Image window widget . . . 644
gui/cLinearWidget.h
Visualization of a linear peptide . . . 649
gui/cMainThread.h
The thread launched by the command 'Search->Run' . . . 652
gui/cMainWindow.h
The main window of the application . . . 654
gui/cMainWindowProxyModel.h
A table proxy model . . . 659
gui/cModificationsProxyModel.h
A proxy model for editor of database of modifications . . . 660
gui/cModificationsWidget.h
An editor of N-/C-terminal modifications database . . . 661
gui/cMultipleButtonDelegate.h
A delegate to create multiple buttons . . . 664
gui/cMultipleDatasetsTableProxyModel.h
A table proxy model . . . 665
gui/cMultipleDatasetsTableWidget.h
Aggregated results from multiple datasets . . . 666
gui/cNeutralLossesListWidget.h
The implementation of a widget where types of neutral losses generated in theoretical spectra are selected . . . 670
gui/cParametersWidget.h
The widget for editing parameters of the application opened using the command 'Search->Settings' . . . 672
gui/cPatternSimulatorSceneWidget.h
The implementation of an isotopic pattern simulator (graphics) . . . 677
gui/cPatternSimulatorWidget.h
The implementation of an isotopic pattern simulator (window) . . . 680
gui/cPreferencesDialog.h
The implementation of preferences dialog . . . 683
gui/cPreferencesIonTypeDialog.h
The implementation of ion type dialog . . . 684
gui/cPreferencesTabDataProcessing.h
The implementation of a tab used to define the data processing parameters in preferences dialog 685
gui/cPreferencesTabDirectories.h
The implementation of a tab used to define default directories in preferences dialog . . . 687
gui/cPreferencesTabIonTypes.h
The implementation of a tab used to define ion types in preferences dialog . . . 689
gui/cPreferencesTabURLs.h
The implementation of a tab used to define URLs in preferences dialog . . . 690
gui/cPubChemSearchWidget.h
The implementation of the dialog PubChem search widget . . . 692
gui/cSequenceDatabaseProxyModel.h
A proxy model for editor of database of sequences . . . 695
gui/cSequenceDatabaseWidget.h
An editor of sequences database . . . 697
gui/cSpectrumDetailProxyModel.h
A table proxy model . . . 699
gui/cSpectrumDetailWidget.h
The widget for visualisation of a detail of a spectrum (opened after doubleclick on a row when results are reported) . . . 700
gui/cSpectrumSceneWidget.h
The implementation of a widget painting a 2D mass spectrum . . . 705
gui/cSummaryPeaksTableProxyModel.h
A table proxy model . . . 708

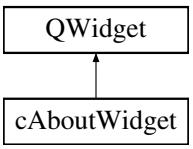
gui/cSummaryPeaksTableWidget.h
Summary table of matched peaks 710
gui/cViewButtonDelegate.h
A delegate to create a reference view button 715
parallel/cFormulaPredictorThread.h
The thread which searches molecular formulas corresponding to m/z value 716
parallel/cGraphReaderThread.h
The thread which scans the de novo graph for peptide sequence candidates 717
parallel/cSpectrumAnalyzerThreadMS1.h
The thread for an initial check of LC-MS and MSI spectra 718
parallel/cSpectrumComparatorThreadMS1.h
The thread for comparison of a theoretical spectrum with a peak list (MS1 level) 719
parallel/cSpectrumComparatorThreadMS2.h
The thread for comparison of a theoretical spectrum with a peak list (MS2 level) 720
parallel/cSpectrumGeneratorThreadMS1.h
The thread for generating a theoretical spectrum (MS1 level) 725

Chapter 16

Class Documentation

16.1 cAboutWidget Class Reference

The widget representing the dialog 'About'.
`#include <cAboutWidget.h>`
Inheritance diagram for cAboutWidget:



Public Member Functions

- cAboutWidget (QWidget *parent=(QWidget *) 0)
  The constructor.
- **~cAboutWidget** ()
  The destructor.
- void closeEvent (QCloseEvent *event)
  Handle the window close event.

Protected Member Functions

- void keyPressEvent (QKeyEvent *event)
  Handle a key press event.

16.1.1 Detailed Description

The widget representing the dialog 'About'.

16.1.2 Constructor & Destructor Documentation

16.1.2.1 cAboutWidget()

```
cAboutWidget::cAboutWidget (
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.1.3 Member Function Documentation

16.1.3.1 closeEvent()

```
void cAboutWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.1.3.2 keyPressEvent()

```
void cAboutWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

The documentation for this class was generated from the following files:

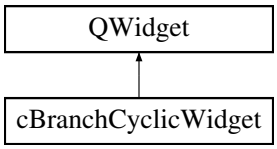
- gui/cAboutWidget.h
- gui/cAboutWidget.cpp

16.2 cBranchCyclicWidget Class Reference

Visualization of a branch-cyclic peptide.

`#include <cBranchCyclicWidget.h>`

Inheritance diagram for cBranchCyclicWidget:



Public Member Functions

- **cBranchCyclicWidget** ()

  The constructor.
- void initialize (cParameters *parameters, cTheoreticalSpectrum *theoreticalspectrum)

  Initialize the widget.
- void exportToPDF (QString filename, bool postscript)

  Export peptide into a PDF or a PS file.

- void exportToPNG (QString filename)

  Export peptide scene into a PNG file.
- void exportToSVG (QString filename)

  Export peptide scene into a SVG file.
- void setReportIsomers (bool reportisomers)

  Set the report isomers state.
- void hidePeakLabels (bool state)

  Hide/Show labels of peaks.

Protected Member Functions

- void paintEvent (QPaintEvent ∗event)

  Handle the paint event.

16.2.1 Detailed Description

Visualization of a branch-cyclic peptide.

16.2.2 Member Function Documentation

16.2.2.1 exportToPDF()

```
void cBranchCyclicWidget::exportToPDF (
            QString filename,
            bool postscript )
```

Export peptide into a PDF or a PS file.

Parameters

filename	filename
postscript	if true then PS file is generated instead of PDF

16.2.2.2 exportToPNG()

```
void cBranchCyclicWidget::exportToPNG (
            QString filename )
```

Export peptide scene into a PNG file.

Parameters

filename	filename

16.2.2.3 exportToSVG()

```
void cBranchCyclicWidget::exportToSVG (
            QString filename )
```

Export peptide scene into a SVG file.

Parameters

filename	filename

16.2.2.4 hidePeakLabels()

```
void cBranchCyclicWidget::hidePeakLabels (
            bool state )
```

Hide/Show labels of peaks.

Parameters

state	the peak labels are hidden when *state* is true

16.2.2.5 initialize()

```
void cBranchCyclicWidget::initialize (
            cParameters * parameters,
            cTheoreticalSpectrum * theoreticalspectrum )
```

Initialize the widget.

Parameters

parameters	a pointer to parameters
theoreticalspectrum	a pointer to a theoretical spectrum

16.2.2.6 paintEvent()

```
void cBranchCyclicWidget::paintEvent (
            QPaintEvent * event )  [protected]
```

Handle the paint event.

Parameters

event	pointer to QPaintEvent

16.2.2.7 setReportIsomers()

```
void cBranchCyclicWidget::setReportIsomers (
            bool reportisomers )
```

Set the report isomers state.

Parameters

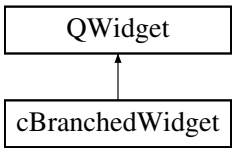
reportisomers	true if isomers are reported; false otherwise

The documentation for this class was generated from the following files:

- gui/cBranchCyclicWidget.h
- gui/cBranchCyclicWidget.cpp

16.3 cBranchedWidget Class Reference

Visualization of a branched peptide.
`#include <cBranchedWidget.h>`
Inheritance diagram for cBranchedWidget:



Public Member Functions

- **cBranchedWidget** ()

  The constructor.
- void initialize (cParameters *parameters, cTheoreticalSpectrum *theoreticalspectrum)

  Initialize the widget.
- void exportToPDF (QString filename, bool postscript)

  Export peptide into a PDF or a PS file.
- void exportToPNG (QString filename)

  Export peptide scene into a PNG file.
- void exportToSVG (QString filename)

  Export peptide scene into a SVG file.
- void setReportIsomers (bool reportisomers)

  Set the report isomers state.
- void hidePeakLabels (bool state)

  Hide/Show labels of peaks.

Protected Member Functions

- void paintEvent (QPaintEvent *event)

  Handle the paint event.

16.3.1 Detailed Description

Visualization of a branched peptide.

16.3.2 Member Function Documentation

16.3.2.1 exportToPDF()

```
void cBranchedWidget::exportToPDF (
            QString filename,
            bool postscript )
```

Export peptide into a PDF or a PS file.

Parameters

filename	filename
postscript	if true then PS file is generated instead of PDF

16.3.2.2 exportToPNG()

```
void cBranchedWidget::exportToPNG (
            QString filename )
```

Export peptide scene into a PNG file.

Parameters

filename	filename

16.3.2.3 exportToSVG()

```
void cBranchedWidget::exportToSVG (
            QString filename )
```

Export peptide scene into a SVG file.

Parameters

filename	filename

16.3.2.4 hidePeakLabels()

```
void cBranchedWidget::hidePeakLabels (
            bool state )
```

Hide/Show labels of peaks.

Parameters

state	the peak labels are hidden when *state* is true

16.3.2.5 initialize()

```
void cBranchedWidget::initialize (
            cParameters * parameters,
            cTheoreticalSpectrum * theoreticalspectrum )
```

Initialize the widget.

Parameters

parameters	a pointer to parameters
theoreticalspectrum	a pointer to a theoretical spectrum

16.3.2.6 paintEvent()

```
void cBranchedWidget::paintEvent (
            QPaintEvent * event )  [protected]
```

Handle the paint event.

Parameters

event	pointer to QPaintEvent

16.3.2.7 setReportIsomers()

```
void cBranchedWidget::setReportIsomers (
            bool reportisomers )
```

Set the report isomers state.

Parameters

reportisomers	true if isomers are reported; false otherwise

The documentation for this class was generated from the following files:

- gui/cBranchedWidget.h
- gui/cBranchedWidget.cpp

16.4 cBrick Class Reference

The class representing a building block (brick).
`#include <cBrick.h>`

Public Member Functions

- **cBrick** ()

  The constructor.
- void **clear** ()

  Clear all variables.
- bool empty ()

  Test whether the brick is empty.
- string & getName ()

  Access to a string variable which stores a full name of the brick.
- void setName (const string &name)

  Set a full name of the brick.
- string & getSummary ()

  Access to a string variable which stores a molecular formula of the brick.
- map< string, int > & getSummaryMap ()

  Access to a map which stores a molecular formula of the brick.
- vector< string > & getAcronyms ()

  Access to a vector of strings which stores acronyms of all izomers corresponding to the brick.
- string & getLosses ()

  Access to a string variable which stores a list of neutral losses.
- vector< int > & getLossIDs ()

  Access to a vector of identifiers of neutral losses.
- vector< string > & getReferences ()

  Access to a vector of strings which stores references (e.g., to ChemSpider) of all izomers corresponding to the brick.
- string & getComposition ()

  Access to a string containing composition of the brick (ids of bricks separated by '-')
- string getReverseComposition ()

  Reverse the composition of the brick.
- double getMass () const

  Get the mass of the brick.
- void setMass (double mass)

Set the mass of the brick.

- void setComposition (const string &composition, bool normalize)

  Set the composition of the brick.
- void setSummary (const string &summary)

  Set the summary molecular formula of the brick.
- void **createSummaryMap** ()

  Create a map of atoms from the molecular formula.
- void setAcronyms (const string &acronyms)

  Set acronyms of all izomers corresponding to the brick.
- void setLosses (const string &str)

  Set the list of losses.
- void setLossIDs (vector< int > &lossids)

  Set a vector of identifiers of neutral losses.
- void setReferences (const string &references)

  Set references (e.g., to ChemSpider) of all izomers corresponding to the brick.
- void **normalizeComposition** ()

  Normalize the composition of the brick (ids of bricks are sorted in ascending order).
- void explodeToIntComposition (vector< int > &bricks)

  Explode the composition of the brick into a vector of intergers (ids of subbricks).
- void explodeToStringComposition (vector< string > &bricks)

  Explode the composition of the brick into a vector of strings (ids of subbricks).
- string getAcronymsAsString ()

  Get all acronyms.
- string getReferencesAsString ()

  Get all references.
- string getFirstAcronymAsString ()

  Get the first acronym.
- string getAcronymsWithReferencesAsHTMLString ()

  Get all acronyms with HTML links to theirs references.
- void setArtificial (bool artificial)

  Set a flag determining whether the brick is a calculated mass block or a regular brick.
- bool isArtificial ()

  Get a flag determining whether the brick is a calculated mass block or a regular brick.
- void setResidueLossType (eResidueLossType residuelosstype)

  Set the residue loss type.
- eResidueLossType getResidueLossType ()

  Get the residue loss type.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is)

  Load the structure from an input stream.

16.4.1 Detailed Description

The class representing a building block (brick).

16.4.2 Member Function Documentation

16.4.2.1 empty()

```
bool cBrick::empty ( )
```

Test whether the brick is empty.

Return values

true	if the brick is empty

16.4.2.2 explodeToIntComposition()

```
void cBrick::explodeToIntComposition (
            vector< int > & bricks )
```

Explode the composition of the brick into a vector of intergers (ids of subbricks).

Parameters

bricks	reference to a vector of integers

16.4.2.3 explodeToStringComposition()

```
void cBrick::explodeToStringComposition (
            vector< string > & bricks )
```

Explode the composition of the brick into a vector of strings (ids of subbricks).

Parameters

bricks	reference to a vector of strings

16.4.2.4 getAcronyms()

```
vector< string > & cBrick::getAcronyms ( )
```

Access to a vector of strings which stores acronyms of all izomers corresponding to the brick.

Return values

reference	to a vector of strings

16.4.2.5 getAcronymsAsString()

```
string cBrick::getAcronymsAsString ( )
```

Get all acronyms.

Return values

string	acronyms of all izomers separated by '/'

16.4.2.6 getAcronymsWithReferencesAsHTMLString()

```
string cBrick::getAcronymsWithReferencesAsHTMLString ( )
```

Get all acronyms with HTML links to theirs references.

Return values

string	acronyms of all izomers with HTML references separated by '/'

16.4.2.7 getComposition()

`string & cBrick::getComposition ( )`
Access to a string containing composition of the brick (ids of bricks separated by '-')

Return values

reference	to a string

16.4.2.8 getFirstAcronymAsString()

`string cBrick::getFirstAcronymAsString ( )`
Get the first acronym.

Return values

string	acronym of the first izomer

16.4.2.9 getLosses()

`string & cBrick::getLosses ( )`
Access to a string variable which stores a list of neutral losses.

Return values

reference	to a string

16.4.2.10 getLossIDs()

`vector< int > & cBrick::getLossIDs ( )`
Access to a vector of identifiers of neutral losses.

Return values

reference	to a vector of identifiers of neutral losses

16.4.2.11 getMass()

`double cBrick::getMass ( ) const`
Get the mass of the brick.

Return values

double	mass

16.4.2.12 getName()

`string & cBrick::getName ( )`
Access to a string variable which stores a full name of the brick.

Return values

reference	to a string

16.4.2.13 getReferences()

`vector< string > & cBrick::getReferences ( )`
Access to a vector of strings which stores references (e.g., to ChemSpider) of all izomers corresponding to the brick.

Return values

reference	to a vector of strings

16.4.2.14 getReferencesAsString()

`string cBrick::getReferencesAsString ( )`
Get all references.

Return values

string	references of all izomers separated by '/'

16.4.2.15 getResidueLossType()

`eResidueLossType cBrick::getResidueLossType ( )`
Get the residue loss type.

Return values

eResidueLossType	the residue loss type.

16.4.2.16 getReverseComposition()

`string cBrick::getReverseComposition ( )`
Reverse the composition of the brick.

Return values

string	with reverted composition of bricks (ids of bricks separated by '-')

16.4.2.17 getSummary()

```
string & cBrick::getSummary ( )
```

Access to a string variable which stores a molecular formula of the brick.

Return values

reference	to a string

16.4.2.18 getSummaryMap()

```
map< string, int > & cBrick::getSummaryMap ( )
```

Access to a map which stores a molecular formula of the brick.

Return values

map<string,int>	reference to a map

16.4.2.19 isArtificial()

```
bool cBrick::isArtificial ( )
```

Get a flag determining whether the brick is a calculated mass block or a regular brick.

Return values

bool	true when the brick is a calculated mass block; false when the brick is a regular brick.

16.4.2.20 load()

```
void cBrick::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.4.2.21 setAcronyms()

```
void cBrick::setAcronyms (
            const string & acronyms )
```

Set acronyms of all izomers corresponding to the brick.

Parameters

acronyms	reference to a string where the acronyms are separated by '/'

16.4.2.22 setArtificial()

```
void cBrick::setArtificial (
            bool artificial )
```

Set a flag determining whether the brick is a calculated mass block or a regular brick.

Parameters

artificial	true when the brick is a calculated mass block; false when the brick is a regular brick.

16.4.2.23 setComposition()

```
void cBrick::setComposition (
            const string & composition,
            bool normalize )
```

Set the composition of the brick.

Parameters

composition	reference to a string with composition (ids of bricks separated by '-')
normalize	if true then ids of bricks are sorted in ascending order

16.4.2.24 setLosses()

```
void cBrick::setLosses (
            const string & str )
```

Set the list of losses.

Parameters

str	reference to a string

16.4.2.25 setLossIDs()

```
void cBrick::setLossIDs (
            vector< int > & lossids )
```

Set a vector of identifiers of neutral losses.

Parameters

lossids	a vector of identifiers of neutral losses

16.4.2.26 setMass()

```
void cBrick::setMass (
            double mass )
```

Set the mass of the brick.

Parameters

mass	mass of the brick

16.4.2.27 setName()

```
void cBrick::setName (
            const string & name )
```

Set a full name of the brick.

Parameters

name	reference to a string

16.4.2.28 setReferences()

```
void cBrick::setReferences (
            const string & references )
```

Set references (e.g., to ChemSpider) of all izomers corresponding to the brick.

Parameters

references	reference to a string where references to izomers are separated by '/'

16.4.2.29 setResidueLossType()

```
void cBrick::setResidueLossType (
            eResidueLossType residuelosstype )
```

Set the residue loss type.

Parameters

residuelosstype	the residue loss type.

16.4.2.30 setSummary()

```
void cBrick::setSummary (
            const string & summary )
```

Set the summary molecular formula of the brick.

Parameters

summary	reference to a string

16.4.2.31 store()

```
void cBrick::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

Parameters

os	an output stream

The documentation for this class was generated from the following files:

- core/cBrick.h
- core/cBrick.cpp

16.5 cBricksDatabase Class Reference

The class representing a database of building blocks (bricks).
`#include <cBricksDatabase.h>`

Public Member Functions

- bool nextCombination (vector< int > &combarray, int numberofbasicbricks, int maximumbricksincombination, double maximumcumulativemass, double neutralprecursormass)

  Get next combination of bricks.
- bool nextCombinationFast (vector< int > &combarray, vector< int > &combcounts, vector< double > &combmasses, double &mass, int numberofbasicbricks, int maximumbricksincombination, double maximumcumulativemass, double neutralprecursormass)

  Get next combination of bricks.
- bool nextCombinationFastLimited (vector< int > &combarray, vector< int > &combcounts, vector< int > &comblimits, vector< double > &combmasses, double &mass, int numberofbasicbricks, int maximumbricksincombination, double maximumcumulativemass, double neutralprecursormass)

  Get next combination of bricks.
- **cBricksDatabase** ()

  The constructor.
- int loadFromPlainTextStream (ifstream &stream, string &errormessage, bool ignoreerrors, bool skiph2blocks)

  Load the database of bricks from a plain text stream.
- void storeToPlainTextStream (ofstream &stream)

  Store the database of bricks into a plain text stream.
- int size ()

  Get the number of bricks in the database.
- void **print** ()

  Print the database of bricks.
- void **sortbyMass** ()

  Sort the database of bricks by mass in ascending order.
- cBrick & operator[ ] (int position)

  Overloaded operator [].
- double getMassOfComposition (vector< int > &combarray, int numberofbasicbricks)

  Get the mass of a composition of bricks.
- string getRealName (string &composition)

  Get the real name of a peptide from a composition.
- string getAcronymName (string &composition, bool useHTMLReferences)

  Get the name of a peptide composed from acronyms of bricks.
- string getTagName (string &composition)

  Get a peptide sequence tag from a composition.
- string getAcronymNameOfTPeptide (const string &tcomposition, bool useHTMLReferences)

  Get the name of a branched peptide composed from acronyms of bricks.

- string getTagNameOfTPeptide (string &tcomposition)

  Get a peptide sequence tag of a branched peptide from a composition.
- void **clear** ()

  Clear the vector of bricks.
- bool replaceAcronymsByIDs (string &sequence, string &errormessage)

  Replace acronyms of bricks by ids.
- void push_back (cBrick &brick)

  Push a new brick into the vector of bricks.
- void **removeLastBrick** ()

  Remove the last brick from the vector of bricks if possible.
- string calculateSummaryFromComposition (int brickid)

  Calculate the summary formula of a composition.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is)

  Load the structure from an input stream.
- bool checkKetideBlocks (cBrick &brickseries, bool regularblocksorder)

  Check if the numbers of H2 loss and H2O loss blocks are correct.
- double getMassOfNeutralLosses (vector< int > &lossids)

  Get the summary mass of neutral losses.
- string getSummaryFormulaOfNeutralLosses (vector< int > &lossids)

  Get the summary formula of neutral losses.
- void getMapOfNeutralLosses (vector< int > &lossids, map< string, int > &summarymap)

  Get the summary map of neutral losses.

16.5.1 Detailed Description

The class representing a database of building blocks (bricks).

16.5.2 Member Function Documentation

16.5.2.1 calculateSummaryFromComposition()

```
string cBricksDatabase::calculateSummaryFromComposition (
            int brickid )
```

Calculate the summary formula of a composition.

Parameters

brickid	brick id

Return values

string	summary formula

16.5.2.2 checkKetideBlocks()

```
bool cBricksDatabase::checkKetideBlocks (
            cBrick & brickseries,
```

```
            bool regularblocksorder )
```

Check if the numbers of H2 loss and H2O loss blocks are correct.

Parameters

brickseries	a tested combination of building blocks
regularblocksorder	check regular order of ketide blocks

Return values

bool	true when the numbers of blocks are correct

16.5.2.3 getAcronymName()

```
string cBricksDatabase::getAcronymName (
            string & composition,
            bool useHTMLReferences )
```

Get the name of a peptide composed from acronyms of bricks.

Parameters

composition	reference to a string containing ids of bricks separated by '-'
useHTMLReferences	if true then each acronym is converted to a link to its reference (e.g., in ChemSpider)

Return values

string	name of the peptide

16.5.2.4 getAcronymNameOfTPeptide()

```
string cBricksDatabase::getAcronymNameOfTPeptide (
            const string & tcomposition,
            bool useHTMLReferences )
```

Get the name of a branched peptide composed from acronyms of bricks.

Parameters

tcomposition	reference to a string containing ids of bricks separated by '-', the branch is specified by '(' and ')'; example: 1-2-3(4-5)6-7
useHTMLReferences	if true then each acronym is converted to a link to its reference (e.g., in ChemSpider)

Return values

string	name of the peptide

16.5.2.5 getMapOfNeutralLosses()

```
void cBricksDatabase::getMapOfNeutralLosses (
```

```
            vector< int > & lossids,
            map< string, int > & summarymap )
```

Get the summary map of neutral losses.

Parameters

lossids	an input vector of ids of neutral losses
summarymap	an output map of atoms

16.5.2.6 getMassOfComposition()

```
double cBricksDatabase::getMassOfComposition (
            vector< int > & combarray,
            int numberofbasicbricks )
```

Get the mass of a composition of bricks.

Parameters

combarray	reference to an input integer vector of ids
numberofbasicbricks	number of basic building blocks

Return values

double	cumulative mass of the bricks in the *combarray*

16.5.2.7 getMassOfNeutralLosses()

```
double cBricksDatabase::getMassOfNeutralLosses (
            vector< int > & lossids )
```

Get the summary mass of neutral losses.

Parameters

lossids	a vector of ids of neutral losses

Return values

double	mass

16.5.2.8 getRealName()

```
string cBricksDatabase::getRealName (
            string & composition )
```

Get the real name of a peptide from a composition.

Parameters

composition	reference to a string containing ids of bricks separated by '-'

Return values

string	name of the peptide

16.5.2.9 getSummaryFormulaOfNeutralLosses()

```
string cBricksDatabase::getSummaryFormulaOfNeutralLosses (
            vector< int > & lossids )
```

Get the summary formula of neutral losses.

Parameters

lossids	a vector of ids of neutral losses

Return values

string	formula

16.5.2.10 getTagName()

```
string cBricksDatabase::getTagName (
            string & composition )
```

Get a peptide sequence tag from a composition.

Parameters

composition	reference to a string containing ids of bricks separated by '-'

Return values

string	sequence tag

16.5.2.11 getTagNameOfTPeptide()

```
string cBricksDatabase::getTagNameOfTPeptide (
            string & tcomposition )
```

Get a peptide sequence tag of a branched peptide from a composition.

Parameters

tcomposition	reference to a string containing ids of bricks separated by '-', the branch is specified by '(' and ')'; example: 1-2-3(4-5)6-7

Return values

string	sequence tag

16.5.2.12 load()

```
void cBricksDatabase::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.5.2.13 loadFromPlainTextStream()

```
int cBricksDatabase::loadFromPlainTextStream (
            ifstream & stream,
            string & errormessage,
            bool ignoreerrors,
            bool skiph2blocks )
```

Load the database of bricks from a plain text stream.

Parameters

stream	reference to an input file stream
errormessage	reference to a string where an error message might be stored
ignoreerrors	if true errors are ignored
skiph2blocks	ignore ketide 2H loss blocks

Return values

int	0 when the database was successfully loaded, -1 when an error occurred (*errormessage* is filled up)

16.5.2.14 nextCombination()

```
bool cBricksDatabase::nextCombination (
            vector< int > & combarray,
            int numberofbasicbricks,
            int maximumbricksincombination,
            double maximumcumulativemass,
            double neutralprecursormass )
```

Get next combination of bricks.

Parameters

combarray	reference to an input/output vector of integers
numberofbasicbricks	initial number of bricks in a database
maximumbricksincombination	maximum number of combined bricks
maximumcumulativemass	maximum cummulative mass of combined bricks
neutralprecursormass	neutral precursor mass

Return values

bool	true when a valid vector is stored in combarray, false when all valid vectors were already generated

16.5.2.15 nextCombinationFast()

```
bool cBricksDatabase::nextCombinationFast (
            vector< int > & combarray,
            vector< int > & combcounts,
            vector< double > & combmasses,
            double & mass,
            int numberofbasicbricks,
            int maximumbricksincombination,
            double maximumcumulativemass,
            double neutralprecursormass )
```

Get next combination of bricks.

Parameters

combarray	reference to an input/output vector of integers
combcounts	reference to an input/output vector containing counts of elements in *combarray*
combmasses	reference to an input vector containing masses of elements in *combarray*
mass	sum of masses of blocks
numberofbasicbricks	initial number of bricks in a database
maximumbricksincombination	maximum number of combined bricks
maximumcumulativemass	maximum cummulative mass of combined bricks
neutralprecursormass	neutral precursor mass

Return values

bool	true when a valid vector is stored in combarray, false when all valid vectors were already generated

16.5.2.16 nextCombinationFastLimited()

```
bool cBricksDatabase::nextCombinationFastLimited (
            vector< int > & combarray,
            vector< int > & combcounts,
            vector< int > & comblimits,
            vector< double > & combmasses,
            double & mass,
            int numberofbasicbricks,
            int maximumbricksincombination,
            double maximumcumulativemass,
            double neutralprecursormass )
```

Get next combination of bricks.

Parameters

combarray	reference to an input/output vector of integers
combcounts	reference to an input/output vector containing counts of elements in *combarray*
comblimits	reference to an input vector containing maximum allowed numbers of elements in *combarray*
combmasses	reference to an input vector containing masses of elements in *combarray*
mass	sum of masses of blocks
numberofbasicbricks	initial number of bricks in a database
maximumbricksincombination	maximum number of combined bricks

Parameters

maximumcumulativemass	maximum cummulative mass of combined bricks
neutralprecursormass	neutral precursor mass

Return values

bool	true when a valid vector is stored in combarray, false when all valid vectors were already generated

16.5.2.17 operator[]()

```
cBrick & cBricksDatabase::operator[] (
            int position )
```

Overloaded operator [].

Parameters

position	of the brick in the vector of bricks

Return values

cBrick	reference to a brick

16.5.2.18 push_back()

```
void cBricksDatabase::push_back (
            cBrick & brick )
```

Push a new brick into the vector of bricks.

Parameters

brick	an inserted brick

16.5.2.19 replaceAcronymsByIDs()

```
bool cBricksDatabase::replaceAcronymsByIDs (
            string & sequence,
            string & errormessage )
```

Replace acronyms of bricks by ids.

Parameters

sequence	reference to an input/output string containing acronyms/ids of bricks
errormessage	reference to a string where an error message is stored when *sequence* contains an unknown building block

Return values

bool	true when the replacement was successful, false when an acronym of a brick is unknown

16.5.2.20 size()

```
int cBricksDatabase::size ( )
```
Get the number of bricks in the database.

Return values

int	number of bricks

16.5.2.21 store()

```
void cBricksDatabase::store (
            ofstream & os )
```
Store the structure into an output stream.

Parameters

os	an output stream

16.5.2.22 storeToPlainTextStream()

```
void cBricksDatabase::storeToPlainTextStream (
            ofstream & stream )
```
Store the database of bricks into a plain text stream.

Parameters

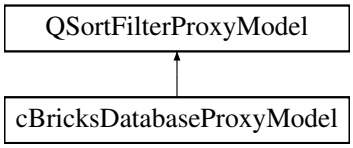
stream	reference to an output file stream

The documentation for this class was generated from the following files:

- core/cBricksDatabase.h
- core/cBricksDatabase.cpp

16.6 cBricksDatabaseProxyModel Class Reference

A proxy model for editor of database of building blocks.
`#include <cBricksDatabaseProxyModel.h>`
Inheritance diagram for cBricksDatabaseProxyModel:



Public Member Functions

- cBricksDatabaseProxyModel (QObject *parent=0)

  The constructor.

- void initialize (QComboBox ∗rowsfilteroperator, QComboBox ∗rowsfiltercombobox1, QComboBox ∗rowsfiltercomparatorcombobox1, QLineEdit ∗rowsfilterline1, QComboBox ∗rowsfiltercombobox2, QComboBox ∗rowsfiltercomparatorcombobox2, QLineEdit ∗rowsfilterline2)

  Initialize the model.
- void setWholeWord (bool wholeword)

  Set whole word option.

Protected Member Functions

- bool filterAcceptsRow (int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE

  Reimplementation of filterAcceptsRow.
- bool lessThan (const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE

  Reimplementation of lessThan.

16.6.1 Detailed Description

A proxy model for editor of database of building blocks.

16.6.2 Constructor & Destructor Documentation

16.6.2.1 cBricksDatabaseProxyModel()

```
cBricksDatabaseProxyModel::cBricksDatabaseProxyModel (
            QObject * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.6.3 Member Function Documentation

16.6.3.1 filterAcceptsRow()

```
bool cBricksDatabaseProxyModel::filterAcceptsRow (
            int sourceRow,
            const QModelIndex & sourceParent ) const  [protected]
```

Reimplementation of filterAcceptsRow.

Parameters

sourceRow	int
sourceParent	QModelIndex

Return values

bool	bool

16.6.3.2 initialize()

```
void cBricksDatabaseProxyModel::initialize (
            QComboBox * rowsfilteroperator,
            QComboBox * rowsfiltercombobox1,
            QComboBox * rowsfiltercomparatorcombobox1,
            QLineEdit * rowsfilterline1,
            QComboBox * rowsfiltercombobox2,
            QComboBox * rowsfiltercomparatorcombobox2,
            QLineEdit * rowsfilterline2 )
```

Initialize the model.

Parameters

rowsfilteroperator	reference to the combobox with operator selection
rowsfiltercombobox1	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox1	reference to the combobox with selection of comparator
rowsfilterline1	reference to a text field
rowsfiltercombobox2	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox2	reference to the combobox with selection of comparator
rowsfilterline2	reference to a text field

16.6.3.3 lessThan()

```
bool cBricksDatabaseProxyModel::lessThan (
            const QModelIndex & left,
            const QModelIndex & right ) const  [protected]
```

Reimplementation of lessThan.

Parameters

left	QModelIndex
right	QModelIndex

Return values

bool	bool

16.6.3.4 setWholeWord()

```
void cBricksDatabaseProxyModel::setWholeWord (
            bool wholeword )
```

Set whole word option.

Parameters

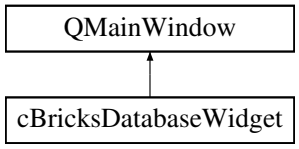
wholeword	true if whole words are matched, false otherwise

The documentation for this class was generated from the following files:

- gui/cBricksDatabaseProxyModel.h
- gui/cBricksDatabaseProxyModel.cpp

16.7 cBricksDatabaseWidget Class Reference

An editor of building blocks database.
`#include <cBricksDatabaseWidget.h>`
Inheritance diagram for cBricksDatabaseWidget:



Public Member Functions

- cBricksDatabaseWidget (cGlobalPreferences *globalpreferences, QWidget *parent=(QWidget *) 0)
  The constructor.
- **~cBricksDatabaseWidget** ()
  The destructor.
- void closeEvent (QCloseEvent *event)
  Handle the window close event.
- void applyGlobalPreferences (cGlobalPreferences *globalpreferences)
  Apply new global preferences.

Protected Member Functions

- void keyPressEvent (QKeyEvent *event)
  Handle a key press event.

16.7.1 Detailed Description

An editor of building blocks database.

16.7.2 Constructor & Destructor Documentation

16.7.2.1 cBricksDatabaseWidget()

```
cBricksDatabaseWidget::cBricksDatabaseWidget (
            cGlobalPreferences * globalpreferences,
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.7.3 Member Function Documentation

16.7.3.1 applyGlobalPreferences()

```
void cBricksDatabaseWidget::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.7.3.2 closeEvent()

```
void cBricksDatabaseWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.7.3.3 keyPressEvent()

```
void cBricksDatabaseWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

The documentation for this class was generated from the following files:

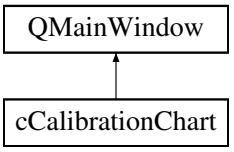
- gui/cBricksDatabaseWidget.h
- gui/cBricksDatabaseWidget.cpp

16.8 cCalibrationChart Class Reference

Calibration chart.

`#include <cCalibrationChart.h>`

Inheritance diagram for cCalibrationChart:



Signals

- void emitXValueInterval (double minxvalue, double maxxvalue)

  The signal is emitted when the range of x value is changed.

Public Member Functions

- cCalibrationChart (cGlobalPreferences ∗globalpreferences, QWidget ∗parent)

  The constructor.
- ∼**cCalibrationChart** ()

  The destructor.
- void closeEvent (QCloseEvent ∗event)

  Handle the window close event.
- void applyGlobalPreferences (cGlobalPreferences ∗globalpreferences)

  Apply new global preferences.
- void setLineParameters (int equationtype, double a, double b, vector< double > xvalues, vector< double > yvalues, vector< double > sd)

  Set parameters of calibration curve.
- void setData (vector< double > xvalues, vector< double > yvalues, vector< double > sd, vector< string > datagroups, vector< int > &datatimevalues)

  Set data points.
- void createTable (double a, double b, vector< double > &calibrationxvalues, vector< double > &calibrationyvalues, vector< double > &calibrationsd, vector< string > &datagroups, vector< double > &dataxvalues, vector< double > &datayvalues, vector< double > &datasd, vector< int > &datatimevalues)

  Create the table content.
- void **deleteTable** ()

  Delete the table content.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.

16.8.1 Detailed Description

Calibration chart.

16.8.2 Constructor & Destructor Documentation

16.8.2.1 cCalibrationChart()

```
cCalibrationChart::cCalibrationChart (
            cGlobalPreferences * globalpreferences,
            QWidget * parent )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.8.3 Member Function Documentation

16.8.3.1 applyGlobalPreferences()

```
void cCalibrationChart::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.8.3.2 closeEvent()

```
void cCalibrationChart::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.8.3.3 createTable()

```
void cCalibrationChart::createTable (
            double a,
            double b,
            vector< double > & calibrationxvalues,
            vector< double > & calibrationyvalues,
            vector< double > & calibrationsd,
            vector< string > & datagroups,
            vector< double > & dataxvalues,
            vector< double > & datayvalues,
            vector< double > & datasd,
            vector< int > & datatimevalues )
```

Create the table content.

Parameters

a	a
b	b
calibrationxvalues	x coordinates of calibration points
calibrationyvalues	y coordinates of calibration points
calibrationsd	standard deviation of y values of calibration points
datagroups	names of groups of data points
dataxvalues	x coordinates of data points
datayvalues	y coordinates of data points
datasd	standard deviation of y values of data points
datatimevalues	data collection times corresponding the groups

16.8.3.4 emitXValueInterval

```
void cCalibrationChart::emitXValueInterval (
```

```
            double minxvalue,
            double maxxvalue )  [signal]
```

The signal is emitted when the range of x value is changed.

Parameters

minxvalue	a minimum threshold of x value
maxxvalue	a maximum threshold of x value

16.8.3.5 keyPressEvent()

```
void cCalibrationChart::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.8.3.6 setData()

```
void cCalibrationChart::setData (
            vector< double > xvalues,
            vector< double > yvalues,
            vector< double > sd,
            vector< string > datagroups,
            vector< int > & datatimevalues )
```

Set data points.

Parameters

xvalues	x coordinates of points
yvalues	y coordinates of points
sd	standard deviation of y values
datagroups	vector of group names
datatimevalues	data collection times corresponding the groups

16.8.3.7 setLineParameters()

```
void cCalibrationChart::setLineParameters (
            int equationtype,
            double a,
            double b,
            vector< double > xvalues,
            vector< double > yvalues,
            vector< double > sd )
```

Set parameters of calibration curve.

Parameters

equationtype	0 (y = a∗x), 1 (y = a∗x + b), 2 (manual mode)

Parameters

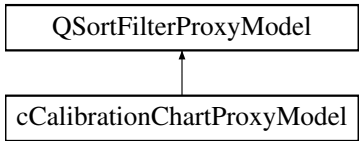
a	a
b	b
xvalues	x coordinates of points
yvalues	y coordinates of points
sd	standard deviation of y values

The documentation for this class was generated from the following files:

- gui/cCalibrationChart.h
- gui/cCalibrationChart.cpp

16.9 cCalibrationChartProxyModel Class Reference

A table proxy model.
`#include <cCalibrationChartProxyModel.h>`
Inheritance diagram for cCalibrationChartProxyModel:



Public Member Functions

- cCalibrationChartProxyModel (QObject *parent=0)

  The constructor.

Protected Member Functions

- bool lessThan (const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE

  Reimplementation of lessThan.

16.9.1 Detailed Description

A table proxy model.

16.9.2 Constructor & Destructor Documentation

16.9.2.1 cCalibrationChartProxyModel()

```
cCalibrationChartProxyModel::cCalibrationChartProxyModel (
            QObject * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.9.3 Member Function Documentation

16.9.3.1 lessThan()

```
bool cCalibrationChartProxyModel::lessThan (
            const QModelIndex & left,
            const QModelIndex & right ) const  [protected]
```

Reimplementation of lessThan.

Parameters

left	QModelIndex
right	QModelIndex

Return values

bool	bool

The documentation for this class was generated from the following files:

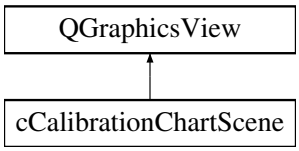
- gui/cCalibrationChartProxyModel.h
- gui/cCalibrationChartProxyModel.cpp

16.10 cCalibrationChartScene Class Reference

The implementation of an isotopic pattern simulator (graphics).

`#include <cCalibrationChartScene.h>`

Inheritance diagram for cCalibrationChartScene:



Signals

- void updateXValueInterval (double minxvalue, double maxxvalue)

  The signal is emitted when the range of x value has been changed.

Public Member Functions

- cCalibrationChartScene (QWidget ∗parent=0)

  The constructor.
- ∼**cCalibrationChartScene** ()

  The destructor.
- void **initialize** ()

  Initialize the widget.
- void exportToPDF (QString filename, bool postscript)

  Export spectrum scene into a PDF or a PS file.
- void exportToSVG (QString filename)

  Export spectrum scene into a SVG file.

- void exportToPNG (QString filename)

  Export spectrum scene into a PNG file.
- void hideLabels (bool state)

  Hide labels of calibration points.
- void setLineParameters (int equationtype, double a, double b, vector< double > xvalues, vector< double > yvalues, vector< double > sd)

  Set parameters of calibration curve.
- void setData (vector< double > datax, vector< double > datay, vector< double > datasd, vector< string > datagroups, vector< int > datatimevalues)

  Set parameters of calibration curve.
- void setGraphType (int type)

  Set graph type.
- double getMaximumConcentration ()

  Get the maximum concentration.
- int getMaximumCollectionTime ()

  Get the maximum collection time.

Protected Member Functions

- void wheelEvent (QWheelEvent *event)

  Handle the mouse wheel event.
- void mouseMoveEvent (QMouseEvent *event)

  Handle the mouse move event.
- void mouseReleaseEvent (QMouseEvent *event)

  Handle the mouse release event.
- void mousePressEvent (QMouseEvent *event)

  Handle the mouse press event.
- void showEvent (QShowEvent *event)

  Handle the widget show event.
- void resizeEvent (QResizeEvent *event)

  Handle the widget resize event.

16.10.1 Detailed Description

The implementation of an isotopic pattern simulator (graphics).

16.10.2 Constructor & Destructor Documentation

16.10.2.1 cCalibrationChartScene()

```
cCalibrationChartScene::cCalibrationChartScene (
            QWidget * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.10.3 Member Function Documentation

16.10.3.1 exportToPDF()

```
void cCalibrationChartScene::exportToPDF (
            QString filename,
            bool postscript )
```

Export spectrum scene into a PDF or a PS file.

Parameters

filename	filename
postscript	if true then PS file is generated instead of PDF

16.10.3.2 exportToPNG()

```
void cCalibrationChartScene::exportToPNG (
            QString filename )
```

Export spectrum scene into a PNG file.

Parameters

filename	filename

16.10.3.3 exportToSVG()

```
void cCalibrationChartScene::exportToSVG (
            QString filename )
```

Export spectrum scene into a SVG file.

Parameters

filename	filename

16.10.3.4 getMaximumCollectionTime()

```
int cCalibrationChartScene::getMaximumCollectionTime ( )
```

Get the maximum collection time.

Return values

int	maximum value of collection time

16.10.3.5 getMaximumConcentration()

```
double cCalibrationChartScene::getMaximumConcentration ( )
```

Get the maximum concentration.

Return values

double	maximum concentration

16.10.3.6 hideLabels()

```
void cCalibrationChartScene::hideLabels (
            bool state )
```

Hide labels of calibration points.

Parameters

state	state of labels

16.10.3.7 mouseMoveEvent()

```
void cCalibrationChartScene::mouseMoveEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse move event.

Parameters

event	pointer to QMouseEvent

16.10.3.8 mousePressEvent()

```
void cCalibrationChartScene::mousePressEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse press event.

Parameters

event	pointer to QMouseEvent

16.10.3.9 mouseReleaseEvent()

```
void cCalibrationChartScene::mouseReleaseEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse release event.

Parameters

event	pointer to QMouseEvent

16.10.3.10 resizeEvent()

```
void cCalibrationChartScene::resizeEvent (
            QResizeEvent * event )  [protected]
```

Handle the widget resize event.

Parameters

event	pointer to QResizeEvent

16.10.3.11 setData()

```
void cCalibrationChartScene::setData (
            vector< double > datax,
            vector< double > datay,
            vector< double > datasd,
            vector< string > datagroups,
            vector< int > datatimevalues )
```

Set parameters of calibration curve.

Parameters

datax	x coordinates of data points
datay	y coordinates of data points
datasd	standard deviation of y values
datagroups	vector of group names
datatimevalues	data collection times corresponding the groups

16.10.3.12 setGraphType()

```
void cCalibrationChartScene::setGraphType (
            int type )
```

Set graph type.

Parameters

type	graph type

16.10.3.13 setLineParameters()

```
void cCalibrationChartScene::setLineParameters (
            int equationtype,
            double a,
            double b,
            vector< double > xvalues,
            vector< double > yvalues,
            vector< double > sd )
```

Set parameters of calibration curve.

Parameters

equationtype	0 (y = a∗x), 1 (y = a∗x + b), 2 (manual mode)
a	a
b	b
xvalues	x coordinates of points
yvalues	y coordinates of points
sd	standard deviation of y values

16.10.3.14 showEvent()

```
void cCalibrationChartScene::showEvent (
            QShowEvent * event )  [protected]
```

Handle the widget show event.

Parameters

event	pointer to QShowEvent

16.10.3.15 updateXValueInterval

```
void cCalibrationChartScene::updateXValueInterval (
            double minxvalue,
            double maxxvalue )  [signal]
```

The signal is emitted when the range of x value has been changed.

Parameters

minxvalue	a minimum threshold of x value
maxxvalue	a maximum threshold of x value

16.10.3.16 wheelEvent()

```
void cCalibrationChartScene::wheelEvent (
            QWheelEvent * event )  [protected]
```

Handle the mouse wheel event.

Parameters

event	pointer to QWheelEvent

The documentation for this class was generated from the following files:

- gui/cCalibrationChartScene.h
- gui/cCalibrationChartScene.cpp

16.11 cCandidate Class Reference

The class representing a peptide sequence candidate.
`#include <cCandidate.h>`

Public Member Functions

- **cCandidate** ()

  The default constructor.
- cCandidate (vector< string > &composition, vector< nodeEdge > &path, eFragmentIonType startiontype, int startmodifID, int endmodifID, int middlemodifID, int middlepos)

  The constructor.
- cCandidate (vector< string > &composition, vector< nodeEdge > &path, eFragmentIonType startiontype, int startmodifID, int endmodifID, int middlemodifID, int branchstart, int branchend)

  The constructor.

- void setCandidate (vector< string > &composition, vector< nodeEdge > &path, eFragmentIonType startiontype, int startmodifID, int endmodifID, int middlemodifID, int middlepos)

  Set a peptide sequence candidate.
- void setCandidate (vector< string > &composition, vector< nodeEdge > &path, eFragmentIonType startiontype, int startmodifID, int endmodifID, int middlemodifID, int branchstart, int branchend)

  Set a peptide sequence candidate.
- void **clear** ()

  Clear the candidate.
- string & getComposition ()

  Access to the composition of the peptide sequence candidate.
- void **revertComposition** ()

  Reverse the bricks composition of the candidate.
- void prepareBranchedCandidates (cCandidateSet &result, ePeptideType peptidetype, bool *terminatecomputation)

  Prepare candidates of a branched or a branch-cyclic peptide.
- string getTComposition ()

  Get a T-composition of a branched peptide.
- string getRevertedTComposition (bool usebrackets)

  Get a reverted T-composition of a branched peptide.
- string getAcronymsTComposition (cBricksDatabase &brickdatabase)

  Get a T-composition of a branched peptide composed from acronyms of bricks.
- string getRealNameTComposition (cBricksDatabase &brickdatabase)

  Get a T-composition of a branched peptide composed from real names of bricks.
- int getBranchSize ()

  Get a number of blocks forming a branch.
- int getBranchStart ()

  Get starting position of a branch.
- int getBranchEnd ()

  Get end position of a branch.
- void getPermutationsOfBranches (vector< TRotationInfo > &tpermutations)

  Get permutations of branches a branched peptide.
- double calculatePrecursorMassFromBricks (cBricksDatabase &brickdatabasewithcombinations, cParameters *parameters)

  Calculate the precursor mass of the peptide sequence candidate from bricks.
- bool isEqualTo (cCandidate &candidate)

  Compare two candidates.
- void getPermutations (cCandidateSet &permutations, bool *terminatecomputation)

  Get a set of candidates with permutations of bricks.
- int getStartModifID ()

  Get id of a terminal modification (at the beginning of a spectrum).
- void setStartModifID (int id)

  Set id of a terminal modification (at the beginning of a spectrum).
- int getEndModifID ()

  Get id of a terminal modification (at the end of a spectrum).
- void setEndModifID (int id)

  Set id of a terminal modification (at the end of a spectrum).
- int getMiddleModifID ()

  Get id of a terminal modification of a branched peptide (in the middle of a spectrum).
- void setMiddleModifID (int id)

  Set id of a terminal modification of a branched peptide (in the middle of a spectrum).
- bool compare (cCandidate &candidate)

  Compare two candidates.

- vector< nodeEdge > & getPath ()

  Get a path in the de novo graph corresponding to the peptide sequence candidate.
- bool hasEqualTPermutations (cCandidate &candidate)

  compare TPermutations of two candidates.
- bool hasOnlyArtificialBricks (cBricksDatabase &brickdatabasewithcombinations)

  Check whether the candidate is composed exclusively from artificial bricks.
- bool hasFirstBrickArtificial (cBricksDatabase &brickdatabasewithcombinations)

  Check if the first brick is artificial.
- bool hasLastBrickArtificial (cBricksDatabase &brickdatabasewithcombinations)

  Check if the last brick is artificial.
- bool hasLastBrickInvalid (cBricksDatabase &brickdatabasewithcombinations)

  Check if the last brick is invalid.
- void getRotations (vector< string > &rotations, bool includerevertedrotations)

  Get rotations of a cyclic peptide sequence.
- void getBranchCyclicRotations (vector< cCandidate > &branchcyclicrotations, bool includerevertedrotations)

  Get branch-cyclic rotations of a branch-cyclic peptide sequence.
- cSummaryFormula calculateSummaryFormulaFromBricks (cParameters &parameters, ePeptideType peptidetype, double precursormass=0)

  Calculate the summary formula of the peptide sequence candidate.
- cSummaryFormula & getSummaryFormula ()

  Get the summary formula of the peptide sequence candidate.
- void setSummaryFormula (cSummaryFormula &summary)

  Set the summary formula of the peptide sequence candidate.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is)

  Load the structure from an input stream.
- void setName (string &name)

  Set the name of peptide.
- string & getName ()

  Get the name of peptide.
- void setAcronyms (cBricksDatabase &bricksdatabase)

  Set a vector of acronyms corresponding to a peptide sequence candidate.
- void setBackboneAcronyms (cBricksDatabase &bricksdatabase)

  Set a vector of acronyms corresponding to a backbone of a peptide sequence candidate.
- void setBranchAcronyms (cBricksDatabase &bricksdatabase)

  Set a vector of acronyms corresponding to a branch of a peptide sequence candidate.
- vector< string > & getAcronyms ()

  Get a vector of acronyms corresponding to a peptide sequence candidate.
- vector< string > & getBackboneAcronyms ()

  Get a vector of acronyms corresponding to a backbone of a peptide sequence candidate.
- vector< string > & getBranchAcronyms ()

  Get a vector of acronyms corresponding to a branch of a peptide sequence candidate.
- void setPath (cDeNovoGraph &graph, cParameters *parameters)

  Set a path in the de novo graph corresponding to the candidate.
- string & getPathAsString ()

  Get a path in the de novo graph corresponding to the candidate as a string.
- void setRealPeptideName (cBricksDatabase &bricksdatabase, ePeptideType peptidetype)

  Set a real peptide name.
- void setAcronymPeptideNameWithHTMLReferences (cBricksDatabase &bricksdatabase, ePeptideType peptidetype)

Set an acronym peptide name.

- string & getRealPeptideName ()

  Get a real peptide name.

- string & getAcronymPeptideNameWithHTMLReferences ()

  Get an acronym peptide name.

- void setStartIonType (eFragmentIonType iontype)

  Set start ion type used in the de novo graph.

- eFragmentIonType getStartIonType ()

  Get start ion type used in the de novo graph.

- bool checkKetideSequence (cBricksDatabase &bricksdatabase, ePeptideType peptidetype, bool regularblocksorder)

  Check if the order of blocks is correct.

- eResidueLossType getLeftResidueType (cBricksDatabase &bricksdatabase)

  Get the residue type of the left-most building block.

- eResidueLossType getRightResidueType (cBricksDatabase &bricksdatabase)

  Get the residue type of the right-most building block.

- bool checkKetideBlocks (cBricksDatabase &bricksdatabase, ePeptideType peptidetype, bool regularblocksorder)

  Check if the numbers of H2 loss and H2O loss blocks are correct.

16.11.1 Detailed Description

The class representing a peptide sequence candidate.

16.11.2 Constructor & Destructor Documentation

16.11.2.1 cCandidate() [1/2]

```
cCandidate::cCandidate (
            vector< string > & composition,
            vector< nodeEdge > & path,
            eFragmentIonType startiontype,
            int startmodifID,
            int endmodifID,
            int middlemodifID,
            int middlepos )
```

The constructor.

Parameters

composition	reference to a vector of strings where each string is a sequence of ids of bricks separated by '-' (each string corresponds to an edge in the de novo graph)
path	reference to a path in the de novo graph which corresponds to a peptide sequence candidate
startiontype	start ion type in the de novo graph
startmodifID	id of a terminal modification (at the beginning of a spectrum)
endmodifID	id of a terminal modification (at the end of a spectrum)
middlemodifID	id of a terminal modification of a branched peptide (in the middle of a spectrum)
middlepos	position in *composition* corresponding to a branch of a branched peptide

16.11.2.2 cCandidate() [2/2]

```
cCandidate::cCandidate (
            vector< string > & composition,
            vector< nodeEdge > & path,
            eFragmentIonType startiontype,
            int startmodifID,
            int endmodifID,
            int middlemodifID,
            int branchstart,
            int branchend )
```

The constructor.

Parameters

composition	reference to a vector of strings where each string is a sequence of ids of bricks separated by '-' (each string corresponds to an edge in the de novo graph)
path	reference to a path in the de novo graph which corresponds to a peptide sequence candidate
startiontype	start ion type in the de novo graph
startmodifID	id of a terminal modification (at the beginning of a spectrum)
endmodifID	id of a terminal modification (at the end of a spectrum)
middlemodifID	id of a terminal modification of a branched peptide (in the middle of a spectrum)
branchstart	position of a brick where a branch of a branched peptide starts (it is assumed that all strings in *composition* are concatenated)
branchend	position of a brick where a branch of a branched peptide ends (it is assumed that all strings in *composition* are concatenated)

16.11.3 Member Function Documentation

16.11.3.1 calculatePrecursorMassFromBricks()

```
double cCandidate::calculatePrecursorMassFromBricks (
            cBricksDatabase & brickdatabasewithcombinations,
            cParameters * parameters )
```

Calculate the precursor mass of the peptide sequence candidate from bricks.

Parameters

brickdatabasewithcombinations	reference to an input database of bricks with combinations of bricks
parameters	a pointer to the parameters of the application

Return values

double	precursor mass of the candidate

16.11.3.2 calculateSummaryFormulaFromBricks()

```
cSummaryFormula cCandidate::calculateSummaryFormulaFromBricks (
            cParameters & parameters,
            ePeptideType peptidetype,
            double precursormass = 0 )
```

Calculate the summary formula of the peptide sequence candidate.

Parameters

parameters	a reference to the parameters of the application
peptidetype	the type of peptide
precursormass	experimantal precursor mass

Return values

string	summary formula

16.11.3.3 checkKetideBlocks()

```
bool cCandidate::checkKetideBlocks (
            cBricksDatabase & bricksdatabase,
            ePeptideType peptidetype,
            bool regularblocksorder )
```

Check if the numbers of H2 loss and H2O loss blocks are correct.

Parameters

bricksdatabase	a database of building blocks
peptidetype	the type of peptide
regularblocksorder	check regular order of ketide blocks

Return values

bool	true when the numbers of blocks are correct

16.11.3.4 checkKetideSequence()

```
bool cCandidate::checkKetideSequence (
            cBricksDatabase & bricksdatabase,
            ePeptideType peptidetype,
            bool regularblocksorder )
```

Check if the order of blocks is correct.

Parameters

bricksdatabase	a database of building blocks
peptidetype	the type of peptide
regularblocksorder	check regular order of ketide blocks

Return values

bool	true when the order of blocks is correct; false otherwise

16.11.3.5 compare()

```
bool cCandidate::compare (
            cCandidate & candidate )
```

Compare two candidates.

Parameters

candidate	reference to the other candidate

Return values

bool	true when the other candidate is greater

16.11.3.6 getAcronymPeptideNameWithHTMLReferences()

```
string & cCandidate::getAcronymPeptideNameWithHTMLReferences ( )
```

Get an acronym peptide name.

Return values

string	reference to the acronym peptide name

16.11.3.7 getAcronyms()

```
vector< string > & cCandidate::getAcronyms ( )
```

Get a vector of acronyms corresponding to a peptide sequence candidate.

Return values

vector<string>	a vector of acronyms

16.11.3.8 getAcronymsTComposition()

```
string cCandidate::getAcronymsTComposition (
            cBricksDatabase & brickdatabase )
```

Get a T-composition of a branched peptide composed from acronyms of bricks.

Parameters

brickdatabase	reference to the database of bricks

Return values

string	containing acronyms of bricks separated by '-', the branch is specified by '(' and ')'

16.11.3.9 getBackboneAcronyms()

```
vector< string > & cCandidate::getBackboneAcronyms ( )
```

Get a vector of acronyms corresponding to a backbone of a peptide sequence candidate.

Return values

vector<string>	a vector of acronyms

16.11.3.10 getBranchAcronyms()

```
vector< string > & cCandidate::getBranchAcronyms ( )
```

Get a vector of acronyms corresponding to a branch of a peptide sequence candidate.

Return values

vector<string>	a vector of acronyms

16.11.3.11 getBranchCyclicRotations()

```
void cCandidate::getBranchCyclicRotations (
            vector< cCandidate > & branchcyclicrotations,
            bool includerevertedrotations )
```

Get branch-cyclic rotations of a branch-cyclic peptide sequence.

Parameters

branchcyclicrotations	reference to an output vector containing branch-cyclic rotations of a sequence
includerevertedrotations	if true then reverted branch-cyclic rotations are also included

16.11.3.12 getBranchEnd()

```
int cCandidate::getBranchEnd ( )
```

Get end position of a branch.

Return values

int	end position of a branch

16.11.3.13 getBranchSize()

```
int cCandidate::getBranchSize ( )
```

Get a number of blocks forming a branch.

Return values

int	number of blocks

16.11.3.14 getBranchStart()

```
int cCandidate::getBranchStart ( )
```

Get starting position of a branch.

Return values

int	starting position of a branch

16.11.3.15 getComposition()

```
string & cCandidate::getComposition ( )
```

Access to the composition of the peptide sequence candidate.

Return values

reference	to a string of ids of bricks separated by '-'

16.11.3.16 getEndModifID()

```
int cCandidate::getEndModifID ( )
```

Get id of a terminal modification (at the end of a spectrum).

Return values

int	id of a terminal modification

16.11.3.17 getLeftResidueType()

```
eResidueLossType cCandidate::getLeftResidueType (
            cBricksDatabase & bricksdatabase )
```

Get the residue type of the left-most building block.

Parameters

bricksdatabase	a database of building blocks

Return values

eResidueLossType	residue loss type of the left-most building block

16.11.3.18 getMiddleModifID()

```
int cCandidate::getMiddleModifID ( )
```

Get id of a terminal modification of a branched peptide (in the middle of a spectrum).

Return values

int	id of a terminal modification

16.11.3.19 getName()

```
string & cCandidate::getName ( )
```

Get the name of peptide.

Return values

string	name

16.11.3.20 getPath()

```
vector< nodeEdge > & cCandidate::getPath ( )
```

Get a path in the de novo graph corresponding to the peptide sequence candidate.

Return values

vector<nodeEdge>	vector of pairs node and edge

16.11.3.21 getPathAsString()

```
string & cCandidate::getPathAsString ( )
```

Get a path in the de novo graph corresponding to the candidate as a string.

Return values

string	reference to a path corresponding to the spectrum

16.11.3.22 getPermutations()

```
void cCandidate::getPermutations (
            cCandidateSet & permutations,
            bool * terminatecomputation )
```

Get a set of candidates with permutations of bricks.

Parameters

permutations	reference to an output set of candidates
terminatecomputation	pointer to a variable determining that the computation must be stopped

16.11.3.23 getPermutationsOfBranches()

```
void cCandidate::getPermutationsOfBranches (
            vector< TRotationInfo > & tpermutations )
```

Get permutations of branches a branched peptide.

Parameters

tpermutations	reference to an output vector of auxiliary structures where permutations will be stored

16.11.3.24 getRealNameTComposition()

```
string cCandidate::getRealNameTComposition (
            cBricksDatabase & brickdatabase )
```

Get a T-composition of a branched peptide composed from real names of bricks.

Parameters

brickdatabase	reference to the database of bricks

Return values

string	containing real names of bricks separated by '-', the branch is specified by '(' and ')'

16.11.3.25 getRealPeptideName()

```
string & cCandidate::getRealPeptideName ( )
```

Get a real peptide name.

Return values

string	reference to the real peptide name

16.11.3.26 getRevertedTComposition()

```
string cCandidate::getRevertedTComposition (
            bool usebrackets )
```

Get a reverted T-composition of a branched peptide.

Parameters

usebrackets	if true then the branch is separated by '(' and ')' else '-' is used instead of brackets

Return values

string	containing ids of bricks separated by '-', the branch may be specified by '(' and ')'; example: 1-2-3(4-5)6-7

16.11.3.27 getRightResidueType()

```
eResidueLossType cCandidate::getRightResidueType (
            cBricksDatabase & bricksdatabase )
```

Get the residue type of the right-most building block.

Parameters

bricksdatabase	a database of building blocks

Return values

eResidueLossType	residue loss type of the right-most building block

16.11.3.28 getRotations()

```
void cCandidate::getRotations (
            vector< string > & rotations,
            bool includerevertedrotations )
```

Get rotations of a cyclic peptide sequence.

Parameters

rotations	reference to an output vector containing rotations of a sequence
includerevertedrotations	if true then reverted rotations are also included

16.11.3.29 getStartIonType()

```
eFragmentIonType cCandidate::getStartIonType ( )
```

Get start ion type used in the de novo graph.

Return values

eFragmentIonType	fragment ion type

16.11.3.30 getStartModifID()

```
int cCandidate::getStartModifID ( )
```

Get id of a terminal modification (at the beginning of a spectrum).

Return values

int	id of a terminal modification

16.11.3.31 getSummaryFormula()

```
cSummaryFormula & cCandidate::getSummaryFormula ( )
```

Get the summary formula of the peptide sequence candidate.

Return values

cSummaryFormula	summary formula

16.11.3.32 getTComposition()

```
string cCandidate::getTComposition ( )
```

Get a T-composition of a branched peptide.

Return values

string	containing ids of bricks separated by '-', the branch is specified by '(' and ')'; example: 1-2-3(4-5)6-7

16.11.3.33 hasEqualTPermutations()

```
bool cCandidate::hasEqualTPermutations (
            cCandidate & candidate )
```

compare TPermutations of two candidates.

Parameters

candidate	reference to the second candidate

Return values

true	when candidates have equal TPermutations

16.11.3.34 hasFirstBrickArtificial()

```
bool cCandidate::hasFirstBrickArtificial (
            cBricksDatabase & brickdatabasewithcombinations )
```

Check if the first brick is artificial.

Parameters

brickdatabasewithcombinations	reference to an input database of bricks with combinations of bricks

Return values

bool	true if the first brick is artificial, false otherwise

16.11.3.35 hasLastBrickArtificial()

```
bool cCandidate::hasLastBrickArtificial (
            cBricksDatabase & brickdatabasewithcombinations )
```

Check if the last brick is artificial.

Parameters

brickdatabasewithcombinations	reference to an input database of bricks with combinations of bricks

Return values

bool	true if the last brick is artificial, false otherwise

16.11.3.36 hasLastBrickInvalid()

```
bool cCandidate::hasLastBrickInvalid (
            cBricksDatabase & brickdatabasewithcombinations )
```

Check if the last brick is invalid.

Parameters

brickdatabasewithcombinations	reference to an input database of bricks with combinations of bricks

Return values

bool	true if the last brick is invalid, false otherwise

16.11.3.37 hasOnlyArtificialBricks()

```
bool cCandidate::hasOnlyArtificialBricks (
            cBricksDatabase & brickdatabasewithcombinations )
```

Check whether the candidate is composed exclusively from artificial bricks.

Parameters

brickdatabasewithcombinations	reference to an input database of bricks with combinations of bricks

Return values

bool	true when the candidate is composed exclusively from artificial bricks; false when the candidate constains a regular brick

16.11.3.38 isEqualTo()

```
bool cCandidate::isEqualTo (
            cCandidate & candidate )
```

Compare two candidates.

Parameters

candidate	reference to the other candidate

Return values

bool	true when the candidates are equal

16.11.3.39 load()

```
void cCandidate::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.11.3.40 prepareBranchedCandidates()

```
void cCandidate::prepareBranchedCandidates (
            cCandidateSet & result,
            ePeptideType peptidetype,
            bool * terminatecomputation )
```

Prepare candidates of a branched or a branch-cyclic peptide.

Parameters

result	reference to an output set of possible peptide sequence candidates
peptidetype	a type of an analyzed peptide
terminatecomputation	pointer to a variable determining that the computation must be stopped

16.11.3.41 setAcronymPeptideNameWithHTMLReferences()

```
void cCandidate::setAcronymPeptideNameWithHTMLReferences (
            cBricksDatabase & bricksdatabase,
            ePeptideType peptidetype )
```

Set an acronym peptide name.

Parameters

bricksdatabase	the database of building blocks
peptidetype	the type of peptide

16.11.3.42 setAcronyms()

```
void cCandidate::setAcronyms (
            cBricksDatabase & bricksdatabase )
```

Set a vector of acronyms corresponding to a peptide sequence candidate.

Parameters

bricksdatabase	a database of building blocks

16.11.3.43 setBackboneAcronyms()

```
void cCandidate::setBackboneAcronyms (
            cBricksDatabase & bricksdatabase )
```

Set a vector of acronyms corresponding to a backbone of a peptide sequence candidate.

Parameters

bricksdatabase	a database of building blocks

16.11.3.44 setBranchAcronyms()

```
void cCandidate::setBranchAcronyms (
            cBricksDatabase & bricksdatabase )
```

Set a vector of acronyms corresponding to a branch of a peptide sequence candidate.

Parameters

bricksdatabase	a database of building blocks

16.11.3.45 setCandidate() [1/2]

```
void cCandidate::setCandidate (
            vector< string > & composition,
            vector< nodeEdge > & path,
            eFragmentIonType startiontype,
            int startmodifID,
            int endmodifID,
            int middlemodifID,
            int branchstart,
            int branchend )
```

Set a peptide sequence candidate.

Parameters

composition	reference to a vector of strings where each string is a sequence of ids of bricks separated by '-' (each string corresponds to an edge in the de novo graph)
path	reference to a path in the de novo graph which corresponds to a peptide sequence candidate
startiontype	start ion type in the de novo graph
startmodifID	id of a terminal modification (at the beginning of a spectrum)
endmodifID	id of a terminal modification (at the end of a spectrum)
middlemodifID	id of a terminal modification of a branched peptide (in the middle of a spectrum)
branchstart	position of a brick where a branch of a branched peptide starts (it is assumed that all strings in *composition* are concatenated)
branchend	position of a brick where a branch of a branched peptide ends (it is assumed that all strings in *composition* are concatenated)

16.11.3.46 setCandidate() [2/2]

```
void cCandidate::setCandidate (
            vector< string > & composition,
            vector< nodeEdge > & path,
            eFragmentIonType startiontype,
            int startmodifID,
            int endmodifID,
            int middlemodifID,
            int middlepos )
```

Set a peptide sequence candidate.

Parameters

composition	reference to a vector of strings where each string is a sequence of ids of bricks separated by '-' (each string corresponds to an edge in the de novo graph)
path	reference to a path in the de novo graph which corresponds to a peptide sequence candidate
startiontype	start ion type in the de novo graph
startmodifID	id of a terminal modification (at the beginning of a spectrum)
endmodifID	id of a terminal modification (at the end of a spectrum)
middlemodifID	id of a terminal modification of a branched peptide (in the middle of a spectrum)
middlepos	position in *composition* corresponding to a branch of a branched peptide

16.11.3.47 setEndModifID()

```
void cCandidate::setEndModifID (
            int id )
```

Set id of a terminal modification (at the end of a spectrum).

Parameters

id	of a terminal modification

16.11.3.48 setMiddleModifID()

```
void cCandidate::setMiddleModifID (
            int id )
```

Set id of a terminal modification of a branched peptide (in the middle of a spectrum).

Parameters

id	of a terminal modification

16.11.3.49 setName()

```
void cCandidate::setName (
            string & name )
```

Set the name of peptide.

Parameters

name	name

16.11.3.50 setPath()

```
void cCandidate::setPath (
            cDeNovoGraph & graph,
            cParameters * parameters )
```

Set a path in the de novo graph corresponding to the candidate.

Parameters

graph	reference to the de novo graph
parameters	parameters of the application

16.11.3.51 setRealPeptideName()

```
void cCandidate::setRealPeptideName (
            cBricksDatabase & bricksdatabase,
            ePeptideType peptidetype )
```

Set a real peptide name.

Parameters

bricksdatabase	the database of building blocks
peptidetype	the type of peptide

16.11.3.52 setStartIonType()

```
void cCandidate::setStartIonType (
            eFragmentIonType iontype )
```

Set start ion type used in the de novo graph.

Parameters

iontype	fragment ion type

16.11.3.53 setStartModifID()

```
void cCandidate::setStartModifID (
            int id )
```

Set id of a terminal modification (at the beginning of a spectrum).

Parameters

id	of a terminal modification

16.11.3.54 setSummaryFormula()

```
void cCandidate::setSummaryFormula (
            cSummaryFormula & summary )
```

Set the summary formula of the peptide sequence candidate.

Parameters

summary	summary formula

16.11.3.55 store()

```
void cCandidate::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

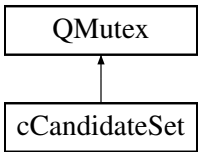
os	an output stream

The documentation for this class was generated from the following files:

- core/cCandidate.h
- core/cCandidate.cpp

16.12 cCandidateSet Class Reference

The class representing a set of peptide sequence candidates.
`#include <cCandidateSet.h>`
Inheritance diagram for cCandidateSet:



Public Member Functions

- **cCandidateSet** ()

  The constructor.
- int size ()

  Get the size of the set.
- unordered_set< cCandidate, hash_cCandidate > & getSet ()

  Access to the set of candidates.

16.12.1 Detailed Description

The class representing a set of peptide sequence candidates.

16.12.2 Member Function Documentation

16.12.2.1 getSet()

```
unordered_set< cCandidate, hash_cCandidate > & cCandidateSet::getSet ( )
```

Access to the set of candidates.

Return values

unordered_set<cCandidate,hash_cCandidate>	reference to the set of candidates

16.12.2.2 size()

```
int cCandidateSet::size ( )
```
Get the size of the set.

Return values

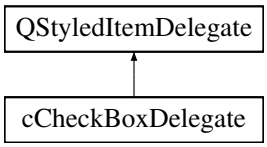
int	size of the set

The documentation for this class was generated from the following files:

- core/cCandidateSet.h
- core/cCandidateSet.cpp

16.13 cCheckBoxDelegate Class Reference

A checkbox item delegate.
`#include <cCheckBoxDelegate.h>`
Inheritance diagram for cCheckBoxDelegate:



Public Member Functions

- cCheckBoxDelegate (QObject ∗parent=(QObject ∗) 0)

  The constructor.
- void paint (QPainter ∗painter, const QStyleOptionViewItem &option, const QModelIndex &index) const

  Reimplementation of paint.

Protected Member Functions

- bool editorEvent (QEvent ∗event, QAbstractItemModel ∗model, const QStyleOptionViewItem &option, const QModelIndex &index)

  Reimplementation of editorEvent.

16.13.1 Detailed Description

A checkbox item delegate.

16.13.2 Constructor & Destructor Documentation

16.13.2.1 cCheckBoxDelegate()

```
cCheckBoxDelegate::cCheckBoxDelegate (
            QObject * parent = (QObject *)0 )
```
The constructor.

Parameters

parent	pointer to a parent widget

16.13.3 Member Function Documentation

16.13.3.1 editorEvent()

```
bool cCheckBoxDelegate::editorEvent (
            QEvent * event,
            QAbstractItemModel * model,
            const QStyleOptionViewItem & option,
            const QModelIndex & index )  [protected]
```

Reimplementation of editorEvent.

Parameters

event	QEvent
model	QAbstractItemModel
option	QStyleOptionViewItem
index	QModelIndex

16.13.3.2 paint()

```
void cCheckBoxDelegate::paint (
            QPainter * painter,
            const QStyleOptionViewItem & option,
            const QModelIndex & index ) const
```

Reimplementation of paint.

Parameters

painter	QPainter
option	QStyleOptionViewItem
index	QModelIndex

The documentation for this class was generated from the following files:

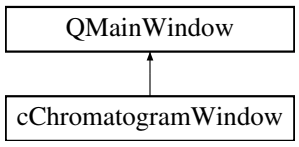
- gui/cCheckBoxDelegate.h
- gui/cCheckBoxDelegate.cpp

16.14 cChromatogramWindow Class Reference

Chromatogram window.

`#include <cChromatogramWindow.h>`

Inheritance diagram for cChromatogramWindow:



Signals

- void emitRetentionTimeInterval (double mintime, double maxtime)

The signal is emitted when the range of retention time has been changed.

- void emitScanIDInterval (int minid, int maxid)

  The signal is emitted when the range of scan numbers has been changed.
- void doubleClickedScanIDSignal (int scanid)

  The signal is emitted a chromatographic peak was double clicked.
- void calculateAvgSpectrum (int minid, int maxid)

  The signal is emitted if the average spectrum is calculated.

Public Member Functions

- cChromatogramWindow (cGlobalPreferences ∗globalpreferences, cTheoreticalSpectrumList &listoftheoreticalspectra, QWidget ∗parent=(QWidget ∗) 0)

  The constructor.
- ∼**cChromatogramWindow** ()

  The destructor.
- void closeEvent (QCloseEvent ∗event)

  Handle the window close event.
- void recalculateTICChromatogram (int activefileid)

  Recalculate TIC chromatogram.
- void setAbsoluteIntensityEnabled (bool enabled)

  Enable/disable absolute intensity mode.
- void setPeakShape (int peakshape)

  Set chromatographic peak shape value.
- void applyGlobalPreferences (cGlobalPreferences ∗globalpreferences)

  Apply new global preferences.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.

16.14.1 Detailed Description

Chromatogram window.

16.14.2 Constructor & Destructor Documentation

16.14.2.1 cChromatogramWindow()

```
cChromatogramWindow::cChromatogramWindow (
            cGlobalPreferences * globalpreferences,
            cTheoreticalSpectrumList & listoftheoreticalspectra,
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
listoftheoreticalspectra	lists of theoretical spectra
parent	pointer to a parent widget

16.14.3 Member Function Documentation

16.14.3.1 applyGlobalPreferences()

```
void cChromatogramWindow::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.14.3.2 calculateAvgSpectrum

```
void cChromatogramWindow::calculateAvgSpectrum (
            int minid,
            int maxid )  [signal]
```

The signal is emitted if the average spectrum is calculated.

Parameters

minid	minimum scan id
maxid	maximum scan id

16.14.3.3 closeEvent()

```
void cChromatogramWindow::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.14.3.4 doubleClickedScanIDSignal

```
void cChromatogramWindow::doubleClickedScanIDSignal (
            int scanid )  [signal]
```

The signal is emitted a chromatographic peak was double clicked.

Parameters

scanid	scan id

16.14.3.5 emitRetentionTimeInterval

```
void cChromatogramWindow::emitRetentionTimeInterval (
            double mintime,
```

```
            double maxtime )  [signal]
```

The signal is emitted when the range of retention time has been changed.

Parameters

mintime	minimum retention time
maxtime	maximum retention time

16.14.3.6 emitScanIDInterval

```
void cChromatogramWindow::emitScanIDInterval (
            int minid,
            int maxid )  [signal]
```

The signal is emitted when the range of scan numbers has been changed.

Parameters

minid	minimum scan id
maxid	maximum scan id

16.14.3.7 keyPressEvent()

```
void cChromatogramWindow::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.14.3.8 recalculateTICChromatogram()

```
void cChromatogramWindow::recalculateTICChromatogram (
            int activefileid )
```

Recalculate TIC chromatogram.

Parameters

activefileid	id of an active file

16.14.3.9 setAbsoluteIntensityEnabled()

```
void cChromatogramWindow::setAbsoluteIntensityEnabled (
            bool enabled )
```

Enable/disable absolute intensity mode.

Parameters

enabled	if true the absolute intensity is used, if false the relative intensity is used

16.14.3.10 setPeakShape()

```
void cChromatogramWindow::setPeakShape (
            int peakshape )
```

Set chromatographic peak shape value.

Parameters

peakshape	0 = Gaussian, 1 = ex-Gaussian (tailing), 2 = ex-Gaussian (fronting)

The documentation for this class was generated from the following files:

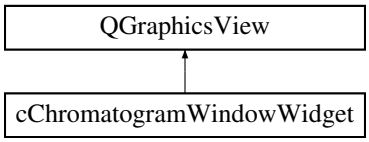
- gui/cChromatogramWindow.h
- gui/cChromatogramWindow.cpp

16.15 cChromatogramWindowWidget Class Reference

Chromatogram window widget.

`#include <cChromatogramWindowWidget.h>`

Inheritance diagram for cChromatogramWindowWidget:



Signals

- void updateRetentionTimeInterval (double mintime, double maxtime)

  The signal is emitted when the range of retention time has been changed.
- void updateScanIDInterval (int minid, int maxid)

  The signal is emitted when the range of scan numbers has been changed.
- void chromatogramWidgetDoubleClicked (int scanid)

  A chromatographic peak was double clicked.

Public Member Functions

- cChromatogramWindowWidget (cTheoreticalSpectrumList &listoftheoreticalspectra, QWidget *parent=0)

  The constructor.
- ~**cChromatogramWindowWidget** ()

  The destructor.
- void exportToPDF (QString filename, bool postscript)

  Export spectrum scene into a PDF or a PS file.
- void exportToSVG (QString filename)

  Export spectrum scene into a SVG file.
- void exportToPNG (QString filename)

  Export spectrum scene into a PNG file.
- void recalculateTICChromatogram (int activefileid)

  Recalculate TIC chromatogram.
- void **setEICChromatogram** (cPeaksList eic)

  Set EIC chromatogram.

Protected Member Functions

- void wheelEvent (QWheelEvent ∗event)

  Handle the mouse wheel event.
- void mouseMoveEvent (QMouseEvent ∗event)

  Handle the mouse move event.
- void mouseReleaseEvent (QMouseEvent ∗event)

  Handle the mouse release event.
- void mousePressEvent (QMouseEvent ∗event)

  Handle the mouse press event.
- void mouseDoubleClickEvent (QMouseEvent ∗event)

  Handle the mouse double click event.
- void showEvent (QShowEvent ∗event)

  Handle the widget show event.
- void resizeEvent (QResizeEvent ∗event)

  Handle the widget resize event.

16.15.1 Detailed Description

Chromatogram window widget.

16.15.2 Constructor & Destructor Documentation

16.15.2.1 cChromatogramWindowWidget()

```
cChromatogramWindowWidget::cChromatogramWindowWidget (
            cTheoreticalSpectrumList & listoftheoreticalspectra,
            QWidget * parent = 0 )
```

The constructor.

Parameters

listoftheoreticalspectra	lists of theoretical spectra
parent	pointer to a parent widget

16.15.3 Member Function Documentation

16.15.3.1 chromatogramWidgetDoubleClicked

```
void cChromatogramWindowWidget::chromatogramWidgetDoubleClicked (
            int scanid )  [signal]
```

A chromatographic peak was double clicked.

Parameters

scanid	id of a scan

16.15.3.2 exportToPDF()

```
void cChromatogramWindowWidget::exportToPDF (
            QString filename,
            bool postscript )
```

Export spectrum scene into a PDF or a PS file.

Parameters

filename	filename
postscript	if true then PS file is generated instead of PDF

16.15.3.3 exportToPNG()

```
void cChromatogramWindowWidget::exportToPNG (
            QString filename )
```

Export spectrum scene into a PNG file.

Parameters

filename	filename

16.15.3.4 exportToSVG()

```
void cChromatogramWindowWidget::exportToSVG (
            QString filename )
```

Export spectrum scene into a SVG file.

Parameters

filename	filename

16.15.3.5 mouseDoubleClickEvent()

```
void cChromatogramWindowWidget::mouseDoubleClickEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse double click event.

Parameters

event	pointer to QMouseEvent

16.15.3.6 mouseMoveEvent()

```
void cChromatogramWindowWidget::mouseMoveEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse move event.

Parameters

event	pointer to QMouseEvent

16.15.3.7 mousePressEvent()

```
void cChromatogramWindowWidget::mousePressEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse press event.

Parameters

event	pointer to QMouseEvent

16.15.3.8 mouseReleaseEvent()

```
void cChromatogramWindowWidget::mouseReleaseEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse release event.

Parameters

event	pointer to QMouseEvent

16.15.3.9 recalculateTICChromatogram()

```
void cChromatogramWindowWidget::recalculateTICChromatogram (
            int activefileid )
```

Recalculate TIC chromatogram.

Parameters

activefileid	id of an active file

16.15.3.10 resizeEvent()

```
void cChromatogramWindowWidget::resizeEvent (
            QResizeEvent * event )  [protected]
```

Handle the widget resize event.

Parameters

event	pointer to QResizeEvent

16.15.3.11 showEvent()

```
void cChromatogramWindowWidget::showEvent (
            QShowEvent * event )  [protected]
```

Handle the widget show event.

Parameters

event	pointer to QShowEvent

16.15.3.12 updateRetentionTimeInterval

```
void cChromatogramWindowWidget::updateRetentionTimeInterval (
            double mintime,
            double maxtime ) [signal]
```

The signal is emitted when the range of retention time has been changed.

Parameters

mintime	minimum retention time
maxtime	maximum retention time

16.15.3.13 updateScanIDInterval

```
void cChromatogramWindowWidget::updateScanIDInterval (
            int minid,
            int maxid ) [signal]
```

The signal is emitted when the range of scan numbers has been changed.

Parameters

minid	minimum scan id
maxid	maximum scan id

16.15.3.14 wheelEvent()

```
void cChromatogramWindowWidget::wheelEvent (
            QWheelEvent * event ) [protected]
```

Handle the mouse wheel event.

Parameters

event	pointer to QWheelEvent

The documentation for this class was generated from the following files:

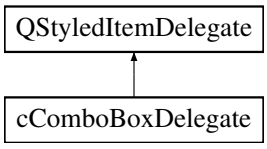
- gui/cChromatogramWindowWidget.h
- gui/cChromatogramWindowWidget.cpp

16.16 cComboBoxDelegate Class Reference

A combobox item delegate.

`#include <cComboBoxDelegate.h>`

Inheritance diagram for cComboBoxDelegate:



Public Member Functions

- cComboBoxDelegate (QObject *parent=(QObject *) 0)

  The constructor.
- void paint (QPainter *painter, const QStyleOptionViewItem &option, const QModelIndex &index) const

  Reimplementation of paint.
- void setEditorData (QWidget *editor, const QModelIndex &index) const

  Reimplementation of setEditorData.
- void setModelData (QWidget *editor, QAbstractItemModel *model, const QModelIndex &index) const

  Reimplementation of setModelData.
- QWidget * createEditor (QWidget *parent, const QStyleOptionViewItem &option, const QModelIndex &index) const

  Reimplementation of createEditor.

16.16.1 Detailed Description

A combobox item delegate.

16.16.2 Constructor & Destructor Documentation

16.16.2.1 cComboBoxDelegate()

```
cComboBoxDelegate::cComboBoxDelegate (
            QObject * parent = (QObject *)0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.16.3 Member Function Documentation

16.16.3.1 createEditor()

```
QWidget * cComboBoxDelegate::createEditor (
            QWidget * parent,
            const QStyleOptionViewItem & option,
            const QModelIndex & index ) const
```

Reimplementation of createEditor.

Parameters

parent	QWidget
option	QStyleOptionViewItem
index	QModelIndex

16.16.3.2 paint()

```
void cComboBoxDelegate::paint (
            QPainter * painter,
```

```
            const QStyleOptionViewItem & option,
            const QModelIndex & index ) const
```

Reimplementation of paint.

Parameters

painter	QPainter
option	QStyleOptionViewItem
index	QModelIndex

16.16.3.3 setEditorData()

```
void cComboBoxDelegate::setEditorData (
            QWidget * editor,
            const QModelIndex & index ) const
```

Reimplementation of setEditorData.

Parameters

editor	QWidget
index	QModelIndex

16.16.3.4 setModelData()

```
void cComboBoxDelegate::setModelData (
            QWidget * editor,
            QAbstractItemModel * model,
            const QModelIndex & index ) const
```

Reimplementation of setModelData.

Parameters

editor	QWidget
model	QAbstractItemModel
index	QModelIndex

The documentation for this class was generated from the following files:

- gui/cComboBoxDelegate.h
- gui/cComboBoxDelegate.cpp

16.17 cCoordinateInfo Struct Reference

Store aggregated information about a coordinate.

```
#include <utilities.h>
```

Public Member Functions

- **cCoordinateInfo** ()

  The constructor.

Public Attributes

- int **id**

  scan id
- int **x**

  x coordinate
- int **y**

  y coordinate
- double **relativeintensity**

  relative intensity
- double **absoluteintensity**

  absolute intensity
- string **description**

  description

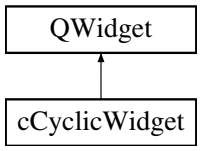
16.17.1 Detailed Description

Store aggregated information about a coordinate.
The documentation for this struct was generated from the following file:

- core/utilities.h

16.18 cCyclicWidget Class Reference

Visualization of a cyclic peptide.
`#include <cCyclicWidget.h>`
Inheritance diagram for cCyclicWidget:



Public Member Functions

- **cCyclicWidget** ()

  The constructor.
- void initialize (cParameters *parameters, cTheoreticalSpectrum *theoreticalspectrum)

  Initialize the widget.
- void exportToPDF (QString filename, bool postscript)

  Export peptide into a PDF or a PS file.
- void exportToPNG (QString filename)

  Export peptide scene into a PNG file.
- void exportToSVG (QString filename)

  Export peptide scene into a SVG file.
- void setReportIsomers (bool reportisomers)

  Set the report isomers state.
- void hidePeakLabels (bool state)

  Hide/Show labels of peaks.

Protected Member Functions

- void paintEvent (QPaintEvent *event)

  Handle the paint event.

16.18.1 Detailed Description

Visualization of a cyclic peptide.

16.18.2 Member Function Documentation

16.18.2.1 exportToPDF()

```
void cCyclicWidget::exportToPDF (
            QString filename,
            bool postscript )
```

Export peptide into a PDF or a PS file.

Parameters

filename	filename
postscript	if true then PS file is generated instead of PDF

16.18.2.2 exportToPNG()

```
void cCyclicWidget::exportToPNG (
            QString filename )
```

Export peptide scene into a PNG file.

Parameters

filename	filename

16.18.2.3 exportToSVG()

```
void cCyclicWidget::exportToSVG (
            QString filename )
```

Export peptide scene into a SVG file.

Parameters

filename	filename

16.18.2.4 hidePeakLabels()

```
void cCyclicWidget::hidePeakLabels (
            bool state )
```

Hide/Show labels of peaks.

Parameters

state	the peak labels are hidden when *state* is true

16.18.2.5 initialize()

```
void cCyclicWidget::initialize (
            cParameters * parameters,
            cTheoreticalSpectrum * theoreticalspectrum )
```

Initialize the widget.

Parameters

parameters	a pointer to parameters
theoreticalspectrum	a pointer to a theoretical spectrum

16.18.2.6 paintEvent()

```
void cCyclicWidget::paintEvent (
            QPaintEvent * event )  [protected]
```

Handle the paint event.

Parameters

event	pointer to QPaintEvent

16.18.2.7 setReportIsomers()

```
void cCyclicWidget::setReportIsomers (
            bool reportisomers )
```

Set the report isomers state.

Parameters

reportisomers	true if isomers are reported; false otherwise

The documentation for this class was generated from the following files:

- gui/cCyclicWidget.h
- gui/cCyclicWidget.cpp

16.19 cDefaultNeutralLosses Class Reference

The class representing a vector of default neutral losses.
`#include <cFragmentIons.h>`

Public Member Functions

- **cDefaultNeutralLosses** ()

  The constructor.
- vector< string > & getNeutralLosses ()

  Get the vector of neutral losses.

16.19.1 Detailed Description

The class representing a vector of default neutral losses.

16.19.2 Member Function Documentation

16.19.2.1 getNeutralLosses()

`vector< string > & cDefaultNeutralLosses::getNeutralLosses ( ) [inline]`
Get the vector of neutral losses.

Return values

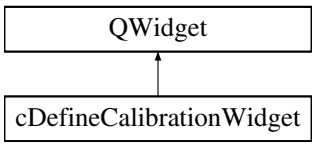
vector<string>	vector of neutral losses

The documentation for this class was generated from the following files:

- core/cFragmentIons.h
- core/cFragmentIons.cpp

16.20 cDefineCalibrationWidget Class Reference

The widget for definition of calibration curve.
`#include <cDefineCalibrationWidget.h>`
Inheritance diagram for cDefineCalibrationWidget:



Signals

- void **calibrationRedefined** ()

  Parameters of calibration curve were redefined.

Public Member Functions

- **cDefineCalibrationWidget** ()

  The constructor.
- ~**cDefineCalibrationWidget** ()

  The destructor.
- void closeEvent (QCloseEvent *event)

  Handle the window close event.
- void store (ofstream &os)

  Store the content into an output stream.
- void load (ifstream &is, int fileversionpart1, int fileversionpart2, int fileversionpart3)

  Load the content from an input stream.
- void prepareWidget (vector< string > &filenames, vector< string > &compounds)

  Prepare widget for visualization.
- void getData (vector< int > &usedvector, vector< int > &datatypesvector, vector< string > &groupsvector, vector< double > &concentrationsvector, vector< int > &timesvector, vector< int > &selectedionsvector, int &equationtype, double &manuala, double &manualb, int &eictype, int &peakshape, int &standard)

  Get data from dialog.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.

16.20.1 Detailed Description

The widget for definition of calibration curve.

16.20.2 Member Function Documentation

16.20.2.1 closeEvent()

```
void cDefineCalibrationWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.20.2.2 getData()

```
void cDefineCalibrationWidget::getData (
            vector< int > & usedvector,
            vector< int > & datatypesvector,
            vector< string > & groupsvector,
            vector< double > & concentrationsvector,
            vector< int > & timesvector,
            vector< int > & selectedionsvector,
            int & equationtype,
            double & manuala,
            double & manualb,
            int & eictype,
            int & peakshape,
            int & standard )
```

Get data from dialog.

Parameters

usedvector	vector of used items
datatypesvector	vector of sample types
groupsvector	vector of groups of names of technical replicates
concentrationsvector	vector of concentrations
timesvector	vector of time values
selectedionsvector	vector of selected ions
equationtype	type of linear equation (y = a∗x, y = a∗x + b or manual mode)
manuala	the parameter 'a' in manual mode
manualb	the parameter 'b' in manual mode
eictype	type of EIC chromatogram (highest peaks = 0, all isotopes = 1)
peakshape	shape of chromatographic peak
standard	compound standard

16.20.2.3 keyPressEvent()

```
void cDefineCalibrationWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.20.2.4 load()

```
void cDefineCalibrationWidget::load (
            ifstream & is,
            int fileversionpart1,
            int fileversionpart2,
            int fileversionpart3 )
```

Load the content from an input stream.

Parameters

is	an input stream
fileversionpart1	first number of .res the file version
fileversionpart2	second number of .res the file version
fileversionpart3	third number of .res the file version

16.20.2.5 prepareWidget()

```
void cDefineCalibrationWidget::prepareWidget (
            vector< string > & filenames,
            vector< string > & compounds )
```

Prepare widget for visualization.

Parameters

filenames	list of files
compounds	list of compounds

16.20.2.6 store()

```
void cDefineCalibrationWidget::store (
            ofstream & os )
```

Store the content into an output stream.

Parameters

os	an output stream

The documentation for this class was generated from the following files:

- gui/cDefineCalibrationWidget.h
- gui/cDefineCalibrationWidget.cpp

16.21 cDeNovoGraph Class Reference

The class implementing the de novo graph.
`#include <cDeNovoGraph.h>`

Public Member Functions

- **cDeNovoGraph** ()

  The constructor.
- void initialize (cMainThread &os, cParameters &parameters)

  Initialize the graph.
- int createGraph (bool &terminatecomputation)

  Create the graph.
- int removeEdgesFormingPathsNotFinishingInPrecursor (bool &terminatecomputation)

  Remove edges which do not form a path finishing in the precursor.
- int removeEdgesFormingPathsNotStartingFromFirstNode (bool &terminatecomputation)

  Remove edges which do not start from system nodes.
- int connectEdgesFormingPathsNotFinishingInPrecursor (bool &terminatecomputation)

  Connect edges which do not form a path finishing in the precursor.
- int connectEdgesFormingPathsNotStartingFromFirstNode (bool &terminatecomputation)

  Connect edges which do not start from system nodes.
- int removePathsWhichCanBeSubstitutedByLongerPath (bool &terminatecomputation)

  Remove paths which can be substituted by longer paths.
- void startGraphReader (cCandidateSet &candidates, long long unsigned &count, int scanmode, bool &terminatecomputation)

  Start the graph reader.
- int getNumberOfEdges ()

  Get the number of edges.
- void printPaths (cMainThread *os, cDeNovoGraphNode &node, vector< string > &path, vector< string > &composition)

  Print all paths in the graph into a stream.
- string printGraph ()

  Print the graph to a string.
- cDeNovoGraphNode & operator[ ] (int position)

  Overloaded operator [].
- void **sortEdgesByPPMError** ()

  Sort edges by ppm error.
- void **sortEdgesByTargetNodeIntensity** ()

  Sort edges by target node intensity.
- cBricksDatabase * getBrickDatabaseWithCombinations ()

  Access to the database of bricks with combinations of bricks.

16.21.1 Detailed Description

The class implementing the de novo graph.

16.21.2 Member Function Documentation

16.21.2.1 connectEdgesFormingPathsNotFinishingInPrecursor()

```
int cDeNovoGraph::connectEdgesFormingPathsNotFinishingInPrecursor (
            bool & terminatecomputation )
```

Connect edges which do not form a path finishing in the precursor.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 = successfully finished, -1 computation terminated

16.21.2.2 connectEdgesFormingPathsNotStartingFromFirstNode()

```
int cDeNovoGraph::connectEdgesFormingPathsNotStartingFromFirstNode (
            bool & terminatecomputation )
```

Connect edges which do not start from system nodes.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 = successfully finished, -1 computation terminated

16.21.2.3 createGraph()

```
int cDeNovoGraph::createGraph (
            bool & terminatecomputation )
```

Create the graph.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 = successfully finished, -1 computation terminated

graph[i].ionannotation[parameters->fragmentionsfordenovograph[k]] ||

16.21.2.4 getBrickDatabaseWithCombinations()

```
cBricksDatabase * cDeNovoGraph::getBrickDatabaseWithCombinations ( )
```

Access to the database of bricks with combinations of bricks.

Return values

cBricksDatabase	pointer to a database of bricks

16.21.2.5 getNumberOfEdges()

```
int cDeNovoGraph::getNumberOfEdges ( )
```

Get the number of edges.

Return values

int	number of edges

16.21.2.6 initialize()

```
void cDeNovoGraph::initialize (
            cMainThread & os,
            cParameters & parameters )
```

Initialize the graph.

Parameters

os	reference to the main thread of the application
parameters	reference to program parameters

16.21.2.7 operator[]()

```
cDeNovoGraphNode & cDeNovoGraph::operator[] (
            int position )
```

Overloaded operator [].

Parameters

position	position of a node

Return values

cDeNovoGraphNode	reference to a node

16.21.2.8 printGraph()

```
string cDeNovoGraph::printGraph ( )
```

Print the graph to a string.

Return values

string	the printed graph

16.21.2.9 printPaths()

```
void cDeNovoGraph::printPaths (
            cMainThread * os,
            cDeNovoGraphNode & node,
            vector< string > & path,
            vector< string > & composition )
```

Print all paths in the graph into a stream.

Parameters

os	pointer to the main thread of the application (stream, respectively).
node	reference to a starting node
path	reference to an auxiliary empty vector of strings
composition	reference to an auxiliary empty vector of strings

16.21.2.10 removeEdgesFormingPathsNotFinishingInPrecursor()

```
int cDeNovoGraph::removeEdgesFormingPathsNotFinishingInPrecursor (
            bool & terminatecomputation )
```

Remove edges which do not form a path finishing in the precursor.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 = successfully finished, -1 computation terminated

16.21.2.11 removeEdgesFormingPathsNotStartingFromFirstNode()

```
int cDeNovoGraph::removeEdgesFormingPathsNotStartingFromFirstNode (
            bool & terminatecomputation )
```

Remove edges which do not start from system nodes.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 = successfully finished, -1 computation terminated

16.21.2.12 removePathsWhichCanBeSubstitutedByLongerPath()

```
int cDeNovoGraph::removePathsWhichCanBeSubstitutedByLongerPath (
```

```
            bool & terminatecomputation )
```

Remove paths which can be substituted by longer paths.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 = successfully finished, -1 computation terminated

16.21.2.13 startGraphReader()

```
void cDeNovoGraph::startGraphReader (
            cCandidateSet & candidates,
            long long unsigned & count,
            int scanmode,
            bool & terminatecomputation )
```

Start the graph reader.

Parameters

candidates	output set of candidates
count	an output number of peptide sequence candidates
scanmode	0 = get candidates; 1 = calculate the number of peptide sequence candidates
terminatecomputation	reference to a variable determining that the computation must be stopped

The documentation for this class was generated from the following files:

- core/cDeNovoGraph.h
- core/cDeNovoGraph.cpp

16.22 cDeNovoGraphNode Class Reference

The class representing a node in the de novo graph.

```
#include <cDeNovoGraphNode.h>
```

Public Member Functions

- **cDeNovoGraphNode** ()

  The constructor.
- void **clear** ()

  Clear the class.
- double getMZRatio ()

  Get the mz ratio of the node.
- void setMZRatio (double mzratio)

  Set the mz ratio of the node.
- double getIntensity ()

  Get the intensity of the node.
- void setIntensity (double intensity)

  Set the intensity of the node.
- void insertTempEdge (cEdge &e)

Insert an edge into a temporary unordered set.

- void insertEdge (cEdge &e)

  Insert an edge into a vector of edges.
- void **sortEdgesByPPMError** ()

  Sort a vector of edges by ppm error.
- void **sortEdgesByTargetNodeIntensity** ()

  Sort a vector of edges by target node intensity.
- void **createVectorOfEdges** ()

  Create a vector of edges from an unordered set of temporary edges.
- int size ()

  Get the number of edges.
- cEdge & operator[ ] (int position)

  Overloaded operator [].
- void removeEdge (int position)

  Remove an edge.
- void **clearEdges** ()

  Clear edges.
- void addIonAnnotation (eFragmentIonType iontype)

  Add ion annotation.
- bool checkIonAnnotation (eFragmentIonType iontype)

  Check ion annotation.

16.22.1 Detailed Description

The class representing a node in the de novo graph.

16.22.2 Member Function Documentation

16.22.2.1 addIonAnnotation()

```
void cDeNovoGraphNode::addIonAnnotation (
            eFragmentIonType iontype )
```

Add ion annotation.

Parameters

iontype	type of fragment ion

16.22.2.2 checkIonAnnotation()

```
bool cDeNovoGraphNode::checkIonAnnotation (
            eFragmentIonType iontype )
```

Check ion annotation.

Parameters

iontype	type of fragment ion

Return values

bool	true when the ion type was assigned to the node

16.22.2.3 getIntensity()

```
double cDeNovoGraphNode::getIntensity ( )
```

Get the intensity of the node.

Return values

double	intensity

16.22.2.4 getMZRatio()

```
double cDeNovoGraphNode::getMZRatio ( )
```

Get the mz ratio of the node.

Return values

double	mz ratio

16.22.2.5 insertEdge()

```
void cDeNovoGraphNode::insertEdge (
            cEdge & e )
```

Insert an edge into a vector of edges.

Parameters

e	reference to an edge

16.22.2.6 insertTempEdge()

```
void cDeNovoGraphNode::insertTempEdge (
            cEdge & e )
```

Insert an edge into a temporary unordered set.

Parameters

e	reference to an edge

16.22.2.7 operator[]()

```
cEdge & cDeNovoGraphNode::operator[] (
            int position )
```

Overloaded operator [].

Parameters

position	position of an edge

Return values

edge	reference to an edge

16.22.2.8 removeEdge()

```
void cDeNovoGraphNode::removeEdge (
            int position )
```

Remove an edge.

Parameters

position	position of removed edge

16.22.2.9 setIntensity()

```
void cDeNovoGraphNode::setIntensity (
            double intensity )
```

Set the intensity of the node.

Parameters

intensity	intensity

16.22.2.10 setMZRatio()

```
void cDeNovoGraphNode::setMZRatio (
            double mzratio )
```

Set the mz ratio of the node.

Parameters

mzratio	mz ratio

16.22.2.11 size()

```
int cDeNovoGraphNode::size ( )
```

Get the number of edges.

Return values

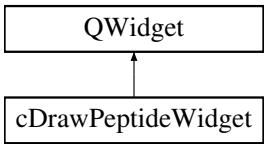
int	number of edges

The documentation for this class was generated from the following files:

- core/cDeNovoGraphNode.h
- core/cDeNovoGraphNode.cpp

16.23 cDrawPeptideWidget Class Reference

Draw a peptide from a sequence of building blocks.
`#include <cDrawPeptideWidget.h>`
Inheritance diagram for cDrawPeptideWidget:



Signals

- void sendSequence (int peptidetypeindex, QString sequence)

  Send index of peptide type and searched sequence (a request for cParametersWidget).
- void sendTag (int peptidetypeindex, QString tag)

  Send index of peptide type and sequence tag (a request for cParametersWidget).
- void sendInsert (int peptidetypeindex, QString sequence)

  Send index of peptide type and sequence (a request for cSequenceDatabaseWidget).

Public Member Functions

- cDrawPeptideWidget (QWidget ∗parent=(QWidget ∗) 0)

  The constructor.
- ∼**cDrawPeptideWidget** ()

  The destructor.
- void closeEvent (QCloseEvent ∗event)

  Handle the window close event.
- void setSequence (int peptidetypeindex, QString sequence)

  Set peptide sequence.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.

16.23.1 Detailed Description

Draw a peptide from a sequence of building blocks.

16.23.2 Constructor & Destructor Documentation

16.23.2.1 cDrawPeptideWidget()

```
cDrawPeptideWidget::cDrawPeptideWidget (
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.23.3 Member Function Documentation

16.23.3.1 closeEvent()

```
void cDrawPeptideWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.23.3.2 keyPressEvent()

```
void cDrawPeptideWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.23.3.3 sendInsert

```
void cDrawPeptideWidget::sendInsert (
            int peptidetypeindex,
            QString sequence )  [signal]
```

Send index of peptide type and sequence (a request for cSequenceDatabaseWidget).

Parameters

peptidetypeindex	an index of current peptide type
sequence	sequence

16.23.3.4 sendSequence

```
void cDrawPeptideWidget::sendSequence (
            int peptidetypeindex,
            QString sequence )  [signal]
```

Send index of peptide type and searched sequence (a request for cParametersWidget).

Parameters

peptidetypeindex	an index of current peptide type
sequence	searched sequence

16.23.3.5 sendTag

```
void cDrawPeptideWidget::sendTag (
            int peptidetypeindex,
            QString tag )  [signal]
```

Send index of peptide type and sequence tag (a request for cParametersWidget).

Parameters

peptidetypeindex	an index of current peptide type
tag	sequence tag

16.23.3.6 setSequence()

```
void cDrawPeptideWidget::setSequence (
            int peptidetypeindex,
            QString sequence )
```

Set peptide sequence.

Parameters

peptidetypeindex	an index of current peptide type
sequence	searched sequence

The documentation for this class was generated from the following files:

- gui/cDrawPeptideWidget.h
- gui/cDrawPeptideWidget.cpp

16.24 cEdge Struct Reference

The structure representing an outgoing edge from a node.
`#include <cDeNovoGraphNode.h>`

Public Member Functions

- **cEdge** ()

  The constructor.
- void **clear** ()

  Clear the structure.

Public Attributes

- int **targetnode**

  Id of a target node.
- int **brickid**

  Id of a brick.
- double **ppmerror**

  Error in ppm.
- double **massdifference**

  A mass difference.
- int **sourcecharge**

  Charge of the source node.

- int **targetcharge**

  Charge of the target node.
- double **targetintensity**

  Intensity of the target node.
- string **composition**

  ids of bricks separated by '-'.
- string **summary**

  summary formula as a string
- int **startmodifID**

  Id of a terminal modification (at the beginning of a spectrum).
- int **middlemodifID**

  Id of a terminal modification of a branched peptide (in the middle of a spectrum).
- int **endmodifID**

  Id of a terminal modification (at the end of a spectrum).
- eFragmentIonType **sourceion**

  Source ion name.
- eFragmentIonType **targetion**

  Target ion name.

16.24.1 Detailed Description

The structure representing an outgoing edge from a node.
The documentation for this struct was generated from the following file:

- core/cDeNovoGraphNode.h

16.25 cEnvelopeSummaryTableKeyMS_comp Struct Reference

Comparison object for cSummaryTableKeyMS (whole envelope).
`#include <cSummaryPeaksTableWidget.h>`

Public Member Functions

- bool operator() (const cSummaryTableKeyMS &left, const cSummaryTableKeyMS &right) const

  Operator to define the order of two items.

16.25.1 Detailed Description

Comparison object for cSummaryTableKeyMS (whole envelope).

16.25.2 Member Function Documentation

16.25.2.1 operator()()

```
bool cEnvelopeSummaryTableKeyMS_comp::operator() (
            const cSummaryTableKeyMS & left,
            const cSummaryTableKeyMS & right ) const  [inline]
```

Operator to define the order of two items.

Parameters

left	first key
right	second key

Return values

bool	true if the first object is less than the second object

The documentation for this struct was generated from the following file:

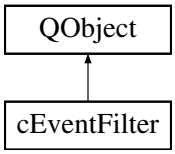
- gui/cSummaryPeaksTableWidget.h

16.26 cEventFilter Class Reference

An event filter.

`#include <cEventFilter.h>`

Inheritance diagram for cEventFilter:



Protected Member Functions

- bool eventFilter (QObject ∗obj, QEvent ∗event)

  Filter events.

16.26.1 Detailed Description

An event filter.

16.26.2 Member Function Documentation

16.26.2.1 eventFilter()

```
bool cEventFilter::eventFilter (
            QObject * obj,
            QEvent * event )  [protected]
```

Filter events.

Parameters

obj	pointer to QObject
event	pointer to QKeyEvent

The documentation for this class was generated from the following files:

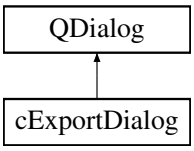
- gui/cEventFilter.h
- gui/cEventFilter.cpp

16.27 cExportDialog Class Reference

The implementation of a dialog to export spectrum/peptide image.

`#include <cExportDialog.h>`

Inheritance diagram for cExportDialog:



Public Member Functions

- cExportDialog (QWidget ∗parent=(QWidget ∗) 0)

  The constructor.
- ∼**cExportDialog** ()

  The destructor.

16.27.1 Detailed Description

The implementation of a dialog to export spectrum/peptide image.

16.27.2 Constructor & Destructor Documentation

16.27.2.1 cExportDialog()

```
cExportDialog::cExportDialog (
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

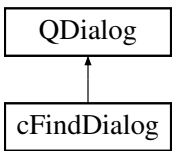
parent	pointer to a parent widget

The documentation for this class was generated from the following files:

- gui/cExportDialog.h
- gui/cExportDialog.cpp

16.28 cFindDialog Class Reference

The implementation of a find dialog.
`#include <cFindDialog.h>`
Inheritance diagram for cFindDialog:



Public Member Functions

- cFindDialog (QWidget ∗parent=(QWidget ∗) 0)

  The constructor.
- ∼**cFindDialog** ()

  The destructor.

16.28.1 Detailed Description

The implementation of a find dialog.

16.28.2 Constructor & Destructor Documentation

16.28.2.1 cFindDialog()

```
cFindDialog::cFindDialog (
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

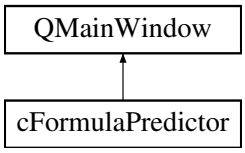
parent	pointer to a parent widget

The documentation for this class was generated from the following files:

- gui/cFindDialog.h
- gui/cFindDialog.cpp

16.29 cFormulaPredictor Class Reference

Predicts the molecular formula from the exact mass.
`#include <cFormulaPredictor.h>`
Inheritance diagram for cFormulaPredictor:



Public Member Functions

- cFormulaPredictor (cGlobalPreferences *globalpreferences, QWidget *parent)
  The constructor.
- ~**cFormulaPredictor** ()
  The destructor.
- void closeEvent (QCloseEvent *event)
  Handle the window close event.
- void **applyGlobalPreferences** ()
  Apply new global preferences.

Protected Member Functions

- void keyPressEvent (QKeyEvent *event)
  Handle a key press event.

16.29.1 Detailed Description

Predicts the molecular formula from the exact mass.

16.29.2 Constructor & Destructor Documentation

16.29.2.1 cFormulaPredictor()

```
cFormulaPredictor::cFormulaPredictor (
            cGlobalPreferences * globalpreferences,
            QWidget * parent )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.29.3 Member Function Documentation

16.29.3.1 closeEvent()

```
void cFormulaPredictor::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.29.3.2 keyPressEvent()

```
void cFormulaPredictor::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

The documentation for this class was generated from the following files:

- gui/cFormulaPredictor.h
- gui/cFormulaPredictor.cpp

16.30 cFormulaPredictorRowItem Struct Reference

An auxiliary structure representing a row in the output table.

`#include <cFormulaPredictorThread.h>`

Public Attributes

- string **ionformula**

  Ion formula.
- double **theoreticalmz**

Theoretical m/z.

- double **mzerror**

  m/z error [ppm].

- string **iontype**

  Ion type.

- int **charge**

  Charge.

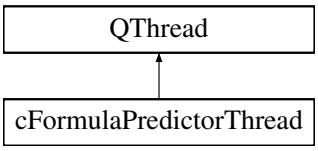
16.30.1 Detailed Description

An auxiliary structure representing a row in the output table.
The documentation for this struct was generated from the following file:

- parallel/cFormulaPredictorThread.h

16.31 cFormulaPredictorThread Class Reference

The thread which searches molecular formulas corresponding to m/z value.
`#include <cFormulaPredictorThread.h>`
Inheritance diagram for cFormulaPredictorThread:



Public Member Functions

- **cFormulaPredictorThread** ()

  The constructor.

- void initialize (int chargevalue, int maximumcombinedelements, bool basicformulacheck, bool advancedformulacheck, bool noratiocheck, double mzvalue, double mzerror, vector< clonType > &selectedions, vector< string > &selectedelements, vector< cFormulaPredictorRowItem > &ionitems, bool &success, bool &terminatecomputation, string &errmsg)

  Initialize the thread.

Protected Member Functions

- void **run** ()

  The main method of the thread.

16.31.1 Detailed Description

The thread which searches molecular formulas corresponding to m/z value.

16.31.2 Member Function Documentation

16.31.2.1 initialize()

```
void cFormulaPredictorThread::initialize (
            int chargevalue,
            int maximumcombinedelements,
            bool basicformulacheck,
            bool advancedformulacheck,
            bool noratiocheck,
            double mzvalue,
            double mzerror,
            vector< cIonType > & selectedions,
            vector< string > & selectedelements,
            vector< cFormulaPredictorRowItem > & ionitems,
            bool & success,
            bool & terminatecomputation,
            string & errmsg )
```

Initialize the thread.

Parameters

chargevalue	charge
maximumcombinedelements	maximum number of combined chemical elements
basicformulacheck	basic filtering rules
advancedformulacheck	advanced filtering rules
noratiocheck	enable N/O rule
mzvalue	experimental m/z value
mzerror	m/z error
selectedions	input list of selected ions
selectedelements	input list of selected elements
ionitems	output list of ions
success	true if the calculation was successful
terminatecomputation	reference to a variable determining that the thread must be stopped
errmsg	an error message

The documentation for this class was generated from the following files:

- parallel/cFormulaPredictorThread.h
- parallel/cFormulaPredictorThread.cpp

16.32 cFragmentIons Class Reference

The detailed definitions of all supported fragment ion types.

`#include <cFragmentIons.h>`

Public Member Functions

- **cFragmentIons** ()

  The constructor.
- cFragmentIons (bool cyclicnterminus, bool cyclicctterminus, string &precursoradduct)

  The constructor.
- cFragmentIonType & operator[ ] (eFragmentIonType iontype)

  Overloaded operator [].
- void recalculateFragments (bool cyclicnterminus, bool cyclicctterminus, string &precursoradduct)

  Recalculate all fragment ion types.
- void store (ofstream &os)

Store the structure into an output stream.

- void load (ifstream &is)

  Load the structure from an input stream.

16.32.1 Detailed Description

The detailed definitions of all supported fragment ion types.

16.32.2 Constructor & Destructor Documentation

16.32.2.1 cFragmentIons()

```
cFragmentIons::cFragmentIons (
            bool cyclicnterminus,
            bool cyclicterminus,
            string & precursoradduct )
```

The constructor.

Parameters

cyclicnterminus	true when the N-terminus of a linear peptide is cyclized
cyclicterminus	true when the C-terminus of a linear peptide is cyclized
precursoradduct	an adduct of a precursor ion

16.32.3 Member Function Documentation

16.32.3.1 load()

```
void cFragmentIons::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.32.3.2 operator[]()

```
cFragmentIonType & cFragmentIons::operator[] (
            eFragmentIonType iontype )
```

Overloaded operator [].

Parameters

iontype	type of fragment ion

Return values

cFragmentIonType	reference to a structure with detailed information about the fragment ion

16.32.3.3 recalculateFragments()

```
void cFragmentIons::recalculateFragments (
            bool cyclicnterminus,
            bool cyclicctterminus,
            string & precursoradduct )
```

Recalculate all fragment ion types.

Parameters

cyclicnterminus	true when the N-terminus of a linear peptide is cyclized
cyclicctterminus	true when the C-terminus of a linear peptide is cyclized
precursoradduct	an adduct of a precursor ion

16.32.3.4 store()

```
void cFragmentIons::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

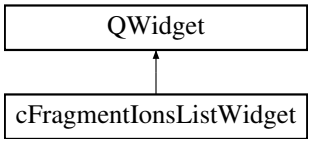
os	an output stream

The documentation for this class was generated from the following files:

- core/cFragmentIons.h
- core/cFragmentIons.cpp

16.33 cFragmentIonsListWidget Class Reference

The widget for selection of fragment ion types.
`#include <cFragmentIonsListWidget.h>`
Inheritance diagram for cFragmentIonsListWidget:



Signals

- void **resetReleased** ()

  Reset button has been released.

Public Member Functions

- cFragmentIonsListWidget (QObject ∗parent)

  The constructor.
- ∼**cFragmentIonsListWidget** ()

  The destructor.

- QListWidget * getList ()

  Access to QListWidget.

16.33.1 Detailed Description

The widget for selection of fragment ion types.

16.33.2 Constructor & Destructor Documentation

16.33.2.1 cFragmentIonsListWidget()

```
cFragmentIonsListWidget::cFragmentIonsListWidget (
            QObject * parent )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.33.3 Member Function Documentation

16.33.3.1 getList()

```
QListWidget * cFragmentIonsListWidget::getList ( )
```

Access to QListWidget.

Return values

QListWidget	pointer to QListWidget

The documentation for this class was generated from the following files:

- gui/cFragmentIonsListWidget.h
- gui/cFragmentIonsListWidget.cpp

16.34 cFragmentIonType Struct Reference

The struct for definition of a fragment ion type.

```
#include <cFragmentIons.h>
```

Public Member Functions

- **cFragmentIonType** ()

  The default constructor.
- void **clear** ()

  Clear the structure.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is)

  Load the structure from an input stream.

Public Attributes

- string **name**

  Name of the fragment.
- string **summary**

  Formula of the fragment.
- map< string, int > **summarymap**

  A map of atoms corresponding to the formula.
- double **massdifference**

  Mass difference of the fragment.
- bool **nterminal**

  True when the fragment is N-terminal.
- bool **cterminal**

  True when the fragment is C-terminal.
- bool **positive**

  Charge of the fragment.

16.34.1 Detailed Description

The struct for definition of a fragment ion type.

16.34.2 Member Function Documentation

16.34.2.1 load()

```
void cFragmentIonType::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.34.2.2 store()

```
void cFragmentIonType::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this struct was generated from the following files:

- core/cFragmentIons.h
- core/cFragmentIons.cpp

16.35 cGlobalPreferences Class Reference

The class storing global preferences.

`#include <cGlobalPreferences.h>`

Public Member Functions

- **cGlobalPreferences** ()

  The constructor.
- void **setDefaultIonTypes** ()

  Set default ion types.
- void **setDefaultDataProcessingMethods** ()

  Set default data processing methods.
- void **setDefaultDirectories** ()

  Set default directories.
- void **setDefaultURLs** ()

  Set default URLs.
- void **loadSettings** ()

  Load settings.
- void **saveSettings** ()

  Save settings.

Public Attributes

- vector< clonType > **customions**
- int **linebafprocessingmethod**
- int **profilebafprocessingmethod**
- int **rawdataprocessingmethod**
- QString **settingsdefaultdir**
- QString **peaklistsdefaultdir**
- QString **blocksdefaultdir**
- QString **sequencesdefaultdir**
- QString **modificationsdefaultdir**
- QString **openopticalimagedir**
- QString **openhistologyimagedir**
- QString **openmicroscopyimagedir**
- QString **resultsdefaultdir**
- QString **exportcsvdefaultdir**
- QString **exporthtmldefaultdir**
- QString **exportimagedefaultdir**
- QString **bookmarkname1**
- QString **bookmarkurl1**
- QString **bookmarkname2**
- QString **bookmarkurl2**
- QString **bookmarkname3**
- QString **bookmarkurl3**
- QString **bookmarkname4**
- QString **bookmarkurl4**
- QString **bookmarkname5**
- QString **bookmarkurl5**
- QString **bookmarkname6**
- QString **bookmarkurl6**
- QString **bookmarkname7**
- QString **bookmarkurl7**
- QString **bookmarkname8**
- QString **bookmarkurl8**
- QString **bookmarkname9**
- QString **bookmarkurl9**
- QString **bookmarkname10**
- QString **bookmarkurl10**
- bool **viewbookmarks**

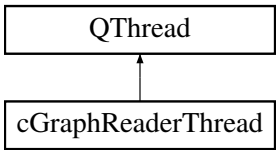
16.35.1 Detailed Description

The class storing global preferences.
The documentation for this class was generated from the following files:

- core/cGlobalPreferences.h
- core/cGlobalPreferences.cpp

16.36 cGraphReaderThread Class Reference

The class representing a thread which scans the de novo graph for peptide sequence candidates.
`#include <cGraphReaderThread.h>`
Inheritance diagram for cGraphReaderThread:



Public Member Functions

- **cGraphReaderThread** ()

  The constructor.
- void initialize (vector< cDeNovoGraphNode > &graph, cBricksDatabase &bricksdatabasewithcombinations, cCandidateSet &candidates, cParameters *parameters, cMainThread *os, int lastsystemnode, long long unsigned &count, int scanmode, bool &terminatecomputation)

  Initialize the graph reader.

Protected Member Functions

- void **run** ()

  The main method of the thread.

16.36.1 Detailed Description

The class representing a thread which scans the de novo graph for peptide sequence candidates.

16.36.2 Member Function Documentation

16.36.2.1 initialize()

```
void cGraphReaderThread::initialize (
            vector< cDeNovoGraphNode > & graph,
            cBricksDatabase & bricksdatabasewithcombinations,
            cCandidateSet & candidates,
            cParameters * parameters,
            cMainThread * os,
            int lastsystemnode,
            long long unsigned & count,
            int scanmode,
            bool & terminatecomputation )
```

Initialize the graph reader.

Parameters

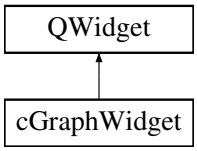
graph	reference to the de novo graph
bricksdatabasewithcombinations	reference to a database of bricks with combinations of bricks
candidates	reference to a set where peptide sequence candidates will be stored
parameters	pointer to program parameters
os	pointer to the main thread of the application
lastsystemnode	position of the last system node in the de novo graph
count	an output number of peptide sequence candidates
scanmode	0 = get candidates; 1 = calculate the number of peptide sequence candidates
terminatecomputation	reference to a variable determining that the thread must be stopped

The documentation for this class was generated from the following files:

- parallel/cGraphReaderThread.h
- parallel/cGraphReaderThread.cpp

16.37 cGraphWidget Class Reference

The widget representing the dialog 'View->Graph'.
`#include <cGraphWidget.h>`
Inheritance diagram for cGraphWidget:



Public Member Functions

- **cGraphWidget** ()

  The constructor.
- ~**cGraphWidget** ()

  The destructor.
- void setHTML (string html, bool reportisomers)

  Set the HTML text into the widget.
- string getHTML ()

  Get the HTML text from the widget.
- void updateView (bool reportisomers)

  Update HTML string if report isomers state is changed.
- void closeEvent (QCloseEvent *event)

  Handle the window close event.
- void store (ofstream &os)

  Store the content into an output stream.
- void load (ifstream &is)

  Load the content from an input stream.

Protected Member Functions

- void keyPressEvent (QKeyEvent *event)

  Handle a key press event.

16.37.1 Detailed Description

The widget representing the dialog 'View->Graph'.

16.37.2 Member Function Documentation

16.37.2.1 closeEvent()

```
void cGraphWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.37.2.2 getHTML()

```
string cGraphWidget::getHTML ( )
```

Get the HTML text from the widget.

Return values

string	HTML string

16.37.2.3 keyPressEvent()

```
void cGraphWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.37.2.4 load()

```
void cGraphWidget::load (
            ifstream & is )
```

Load the content from an input stream.

Parameters

is	an input stream

16.37.2.5 setHTML()

```
void cGraphWidget::setHTML (
```

```
            string html,
            bool reportisomers )
```

Set the HTML text into the widget.

Parameters

html	string with the HTML text
reportisomers	true if isomers of building blocks are reported; false otherwise

16.37.2.6 store()

```
void cGraphWidget::store (
            ofstream & os )
```

Store the content into an output stream.

Parameters

os	an output stream

16.37.2.7 updateView()

```
void cGraphWidget::updateView (
            bool reportisomers )
```

Update HTML string if report isomers state is changed.

Parameters

reportisomers	true if isomers of building blocks are reported; false otherwise

The documentation for this class was generated from the following files:

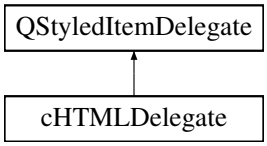
- gui/cGraphWidget.h
- gui/cGraphWidget.cpp

16.38 cHTMLDelegate Class Reference

A delegate to display rich text.

`#include <cHTMLDelegate.h>`

Inheritance diagram for cHTMLDelegate:



Public Member Functions

- cHTMLDelegate (QObject ∗parent=(QObject ∗) 0)

  The constructor.
- void paint (QPainter ∗painter, const QStyleOptionViewItem &option, const QModelIndex &index) const

  Reimplementation of paint.

16.38.1 Detailed Description

A delegate to display rich text.

16.38.2 Constructor & Destructor Documentation

16.38.2.1 cHTMLDelegate()

```
cHTMLDelegate::cHTMLDelegate (
            QObject * parent = (QObject *)0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.38.3 Member Function Documentation

16.38.3.1 paint()

```
void cHTMLDelegate::paint (
            QPainter * painter,
            const QStyleOptionViewItem & option,
            const QModelIndex & index ) const
```

Reimplementation of paint.

Parameters

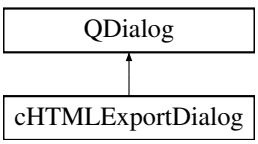
painter	QPainter
option	QStyleOptionViewItem
index	QModelIndex

The documentation for this class was generated from the following files:

- gui/cHTMLDelegate.h
- gui/cHTMLDelegate.cpp

16.39 cHTMLExportDialog Class Reference

The implementation of HTML export dialog.
`#include <cHTMLExportDialog.h>`
Inheritance diagram for cHTMLExportDialog:



Public Member Functions

- cHTMLExportDialog (QWidget ∗parent=(QWidget ∗) 0)

The constructor.

- ~**cHTMLExportDialog** ()

  The destructor.

Public Attributes

- QCheckBox * **checkboxparameters**

  Export parameters.

- QCheckBox * **checkboxdenovo**

  Export de novo graph.

- QCheckBox * **checkboxlogwindow**

  Export log window.

- QCheckBox * **checkboxsummaryresultstable**

  Export output report table.

- QCheckBox * **checkboxsummarypeakstable**

  Export summary table of matched peaks.

- QCheckBox * **checkboxpeakstable**

  Export peakss table.

- QCheckBox * **checkboxunmatchedexperimental**

  Export unmatched experimental peaks.

- QCheckBox * **checkboxunmatchedtheoretical**

  Export matched experimental peaks.

- QCheckBox * **checkboxdetails**

  Export details of identification.

16.39.1 Detailed Description

The implementation of HTML export dialog.

16.39.2 Constructor & Destructor Documentation

16.39.2.1 cHTMLExportDialog()

```
cHTMLExportDialog::cHTMLExportDialog (
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

The documentation for this class was generated from the following files:

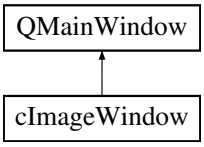
- gui/cHTMLExportDialog.h
- gui/cHTMLExportDialog.cpp

16.40 cImageWindow Class Reference

Image window.

`#include <cImageWindow.h>`

Inheritance diagram for cImageWindow:



Signals

- void updateSummaryPeaksTableFilter (int xmin, int xmax, int ymin, int ymax)

  The signal is emitted when the region selection was changed.
- void layerChanged (int layerid, bool checked, int alpha, int zvalue, bool redraw)

  The signal is emitted when a layer was changed.
- void activeLayerChanged (int layerid)

  The signal is emitted when a layer is selected.
- void doubleClickedSpectrumIDSignal (int spectrumid)

  The signal is emitted the image is double clicked.

Public Member Functions

- cImageWindow (cGlobalPreferences ∗globalpreferences, QWidget ∗parent=(QWidget ∗) 0)

  The constructor.
- ∼**cImageWindow** ()

  The destructor.
- void closeEvent (QCloseEvent ∗event)

  Handle the window close event.
- void setDefaultValues (string imzmlfilename, int defaultmaxx, int defaultmaxy, int defaultpixelsizex, int defaultpixelsizey, eVendorType vendor)

  Set the filename and some default values parsed from the imzML file.
- void addLayer (QString name)

  Add a layer.
- void applyGlobalPreferences (cGlobalPreferences ∗globalpreferences)

  Apply new global preferences.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.
- void keyReleaseEvent (QKeyEvent ∗event)

  Handle a key release event.

16.40.1 Detailed Description

Image window.

16.40.2 Constructor & Destructor Documentation

16.40.2.1 cImageWindow()

```
cImageWindow::cImageWindow (
            cGlobalPreferences * globalpreferences,
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.40.3 Member Function Documentation

16.40.3.1 activeLayerChanged

```
void cImageWindow::activeLayerChanged (
            int layerid )  [signal]
```

The signal is emitted when a layer is selected.

Parameters

layerid	id of the layer

16.40.3.2 addLayer()

```
void cImageWindow::addLayer (
            QString name )
```

Add a layer.

Parameters

name	layer name

16.40.3.3 applyGlobalPreferences()

```
void cImageWindow::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.40.3.4 closeEvent()

```
void cImageWindow::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.40.3.5 doubleClickedSpectrumIDSignal

```
void cImageWindow::doubleClickedSpectrumIDSignal (
            int spectrumid )  [signal]
```

The signal is emitted the image is double clicked.

Parameters

spectrumid	id of a spectrum

16.40.3.6 keyPressEvent()

```
void cImageWindow::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.40.3.7 keyReleaseEvent()

```
void cImageWindow::keyReleaseEvent (
            QKeyEvent * event )  [protected]
```

Handle a key release event.

Parameters

event	pointer to QKeyEvent

16.40.3.8 layerChanged

```
void cImageWindow::layerChanged (
            int layerid,
            bool checked,
            int alpha,
            int zvalue,
            bool redraw )  [signal]
```

The signal is emitted when a layer was changed.

Parameters

layerid	id of the layer
checked	true if the layer was checked, false if the layer was unchecked
alpha	transparency of the layer
zvalue	z-value of the layer
redraw	redraw scene

16.40.3.9 setDefaultValues()

```
void cImageWindow::setDefaultValues (
            string imzmlfilename,
            int defaultmaxx,
            int defaultmaxy,
            int defaultpixelsizex,
            int defaultpixelsizey,
            eVendorType vendor )
```

Set the filename and some default values parsed from the imzML file.

Parameters

imzmlfilename	name of imzML file
defaultmaxx	max count of pixel x - parsed from imzML file
defaultmaxy	max count of pixel y - parsed from imzML file
defaultpixelsizex	pixel size x - parsed from imzML file
defaultpixelsizey	pixel size y - parsed from imzML file
vendor	vendor type

16.40.3.10 updateSummaryPeaksTableFilter

```
void cImageWindow::updateSummaryPeaksTableFilter (
            int xmin,
            int xmax,
            int ymin,
            int ymax )  [signal]
```

The signal is emitted when the region selection was changed.

Parameters

xmin	minimum x coordinate
xmax	maximum x coordinate
ymin	minimum y coordinate
ymax	maximum y coordinate

The documentation for this class was generated from the following files:

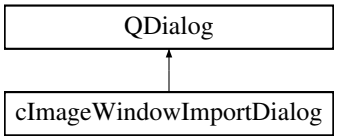
- gui/cImageWindow.h
- gui/cImageWindow.cpp

16.41 cImageWindowImportDialog Class Reference

Import dialog in image window.

`#include <cImageWindowImportDialog.h>`

Inheritance diagram for cImageWindowImportDialog:



Public Slots

- void **accept** ()

  Override the accept slot.

Public Member Functions

- cImageWindowImportDialog (QWidget *parent=0)

  The constructor.
- void setFormValues (QString &lastdirmicroscopyimage, int numberoflayers)

  Initialize values in the form.
- int getFormValues (int &startfromindex, QStringList &tiffilenames, int &navigationindex, QString &liffilename)

  Get the values from the form.
- ~**cImageWindowImportDialog** ()

  The destructor.

16.41.1 Detailed Description

Import dialog in image window.

16.41.2 Constructor & Destructor Documentation

16.41.2.1 cImageWindowImportDialog()

```
cImageWindowImportDialog::cImageWindowImportDialog (
            QWidget * parent = 0 )  [explicit]
```

The constructor.

Parameters

parent	pointer to a parent widget

16.41.3 Member Function Documentation

16.41.3.1 getFormValues()

```
int cImageWindowImportDialog::getFormValues (
            int & startfromindex,
            QStringList & tiffilenames,
            int & navigationindex,
            QString & liffilename )
```

Get the values from the form.

Parameters

startfromindex	starting index to open microscopy images
tiffilenames	a list of tif images to be opened
navigationindex	index of navigation image
liffilename	a lif file from which the metadata are read

Return values

int	tab index

16.41.3.2 setFormValues()

```
void cImageWindowImportDialog::setFormValues (
            QString & lastdirmicroscopyimage,
            int numberoflayers )
```

Initialize values in the form.

Parameters

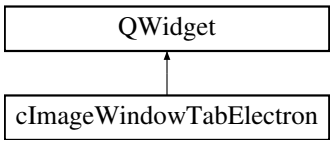
lastdirmicroscopyimage	a default directory with microscopy images
numberoflayers	number of layers

The documentation for this class was generated from the following files:

- gui/cImageWindowImportDialog.h
- gui/cImageWindowImportDialog.cpp

16.42 cImageWindowTabElectron Class Reference

The class representing a tab used to import images from electron microscopy.
`#include <cImageWindowTabElectron.h>`
Inheritance diagram for cImageWindowTabElectron:



Public Member Functions

- cImageWindowTabElectron (QWidget *parent=0)
  The constructor.
- void setFormValues (QString &lastdirmicroscopyimage, int numberoflayers)
  Initialize values in the form.
- void getFormValues (int &index, QStringList &filenames)
  Get the values from the form.
- **~cImageWindowTabElectron** ()
  The destructor.

16.42.1 Detailed Description

The class representing a tab used to import images from electron microscopy.

16.42.2 Constructor & Destructor Documentation

16.42.2.1 cImageWindowTabElectron()

```
cImageWindowTabElectron::cImageWindowTabElectron (
            QWidget * parent = 0 )  [explicit]
```

The constructor.

Parameters

parent	pointer to a parent widget

16.42.3 Member Function Documentation

16.42.3.1 getFormValues()

```
void cImageWindowTabElectron::getFormValues (
            int & index,
            QStringList & filenames )
```

Get the values from the form.

Parameters

index	starting index to open microscopy images
filenames	list of images to open

16.42.3.2 setFormValues()

```
void cImageWindowTabElectron::setFormValues (
            QString & lastdirmicroscopyimage,
            int numberoflayers )
```

Initialize values in the form.

Parameters

lastdirmicroscopyimage	a default directory with microscopy images
numberoflayers	number of layers

The documentation for this class was generated from the following files:

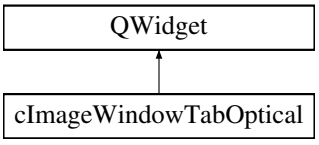
- gui/cImageWindowTabElectron.h
- gui/cImageWindowTabElectron.cpp

16.43 cImageWindowTabOptical Class Reference

The class representing a tab used to import images from light microscopy.

`#include <cImageWindowTabOptical.h>`

Inheritance diagram for cImageWindowTabOptical:



Public Member Functions

- cImageWindowTabOptical (QWidget *parent=0)

  The constructor.
- void setFormValues (QString &lastdirmicroscopyimage, int numberoflayers)

  Initialize values in the form.
- void getFormValues (int &startfromindex, QStringList &tiffilenames, int &navigationindex, QString &liffilename)

  Get the values from the form.
- ~**cImageWindowTabOptical** ()

  The destructor.

16.43.1 Detailed Description

The class representing a tab used to import images from light microscopy.

16.43.2 Constructor & Destructor Documentation

16.43.2.1 cImageWindowTabOptical()

```
cImageWindowTabOptical::cImageWindowTabOptical (
            QWidget * parent = 0 )  [explicit]
```

The constructor.

Parameters

parent	pointer to a parent widget

16.43.3 Member Function Documentation

16.43.3.1 getFormValues()

```
void cImageWindowTabOptical::getFormValues (
            int & startfromindex,
            QStringList & tiffilenames,
            int & navigationindex,
            QString & liffilename )
```

Get the values from the form.

Parameters

startfromindex	starting index to open microscopy images
tiffilenames	a list of tif images to be opened
navigationindex	index of navigation image
liffilename	a lif file from which the metadata are read

16.43.3.2 setFormValues()

```
void cImageWindowTabOptical::setFormValues (
            QString & lastdirmicroscopyimage,
```

```
int numberoflayers )
```

Initialize values in the form.

Parameters

lastdirmicroscopyimage	a default directory with microscopy images
numberoflayers	number of layers

The documentation for this class was generated from the following files:

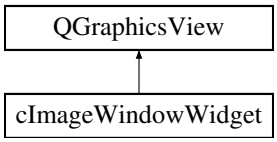
- gui/cImageWindowTabOptical.h
- gui/cImageWindowTabOptical.cpp

16.44 cImageWindowWidget Class Reference

Image window widget.

`#include <cImageWindowWidget.h>`

Inheritance diagram for cImageWindowWidget:



Signals

- void updateFilter (int xmin, int xmax, int ymin, int ymax)

  The signal is emitted when the selection of points was changed.
- void updateCorrelation (int xmin, int xmax, int ymin, int ymax)

  The signal is emitted when the points area in the optical image was changed.
- void updateHistologyPosition (int x, int y, int width, int height, double angle)

  The signal is emitted when the histology position was changed.
- void updateMicroscopyPosition (bool flipx, bool flipy, double x, double y, double width, double height, double angle)

  The signal is emitted when the microscopy position was changed.
- void updateRuler (double size)

  The signal is emitted when the ruler size was changed.
- void imageWidgetDoubleClicked (int spectrumid)

  The image was double clicked.

Public Member Functions

- **cImageWindowWidget** ()

  The constructor.
- ∼**cImageWindowWidget** ()

  The destructor.
- void setOpticalImage (QImage ∗image)

  Set the optical image.
- void setHistologyImage (QImage ∗histologyimage)

  Set the histology image.
- void setMicroscopyImage (int layer, QImage ∗microscopyimage)

  Set the microscopy image.

- QImage getImage ()

  Get the image.
- void setFilterOptions (vector< cCoordinateInfo > &coordinateinfo, bool operatortype, string &columnname1, string &comparatorname1, string &filterstring1, string &columnname2, string &comparatorname2, string &filterstring2, bool casesensitive, bool wholeword)

  Set options which was used to filter the points.
- void setCorrelationValues (int leftshift, int topshift, int maxx, int maxy, int pixelsizex, int pixelsizey)

  Set the values to corelate data with optical image.
- void setDefaultMaxXY (int defaultmaxx, int defaultmaxy, int defaultpixelsizex, int defaultpixelsizey, eVendorType vendor)

  Set the default maximum X and Y coordinates - parsed from the imzML file.
- void setHistologyPosition (int x, int y, int width, int height, double angle)

  Set the position of a histology image.
- void setMicroscopyPosition (int layer, bool flipx, bool flipy, double x, double y, double width, double height, double angle)

  Set the position of a microscopy image.
- void goToMicroscopyPosition (int layer)

  Go to the position of a microscopy image.
- void setSelectedRegion (int xfrom, int xto, int yfrom, int yto)

  Set the current region.
- void **redraw** ()

  Redraw widget.
- void **clearLayers** ()

  Clear the vector of layers.
- void **clearLayer** (int layer)

  Clear a layer.
- void setKeepAspectRatio (bool state)

  Set the flag to keep aspect ratio.
- void setNavigationLayer (int layer, int navigation)

  Set navigation layer.

Protected Member Functions

- void wheelEvent (QWheelEvent ∗event)

  Handle the mouse wheel event.
- void mousePressEvent (QMouseEvent ∗event)

  Handle the mouse press event.
- void mouseMoveEvent (QMouseEvent ∗event)

  Handle the mouse move event.
- void mouseReleaseEvent (QMouseEvent ∗event)

  Handle the mouse release event.
- void mouseDoubleClickEvent (QMouseEvent ∗event)

  Handle the mouse double click event.
- void resizeEvent (QResizeEvent ∗event)

  Handle the resize event.

16.44.1 Detailed Description

Image window widget.

16.44.2 Member Function Documentation

16.44.2.1 getImage()

```
QImage cImageWindowWidget::getImage ( )
```

Get the image.

Return values

QImage	image

16.44.2.2 goToMicroscopyPosition()

```
void cImageWindowWidget::goToMicroscopyPosition (
            int layer )
```

Go to the position of a microscopy image.

Parameters

layer	a microscopy layer

16.44.2.3 imageWidgetDoubleClicked

```
void cImageWindowWidget::imageWidgetDoubleClicked (
            int spectrumid )  [signal]
```

The image was double clicked.

Parameters

spectrumid	id of a spectrum

16.44.2.4 mouseDoubleClickEvent()

```
void cImageWindowWidget::mouseDoubleClickEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse double click event.

Parameters

event	pointer to QMouseEvent

16.44.2.5 mouseMoveEvent()

```
void cImageWindowWidget::mouseMoveEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse move event.

Parameters

event	pointer to QMouseEvent

16.44.2.6 mousePressEvent()

```
void cImageWindowWidget::mousePressEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse press event.

Parameters

event	pointer to QMouseEvent

16.44.2.7 mouseReleaseEvent()

```
void cImageWindowWidget::mouseReleaseEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse release event.

Parameters

event	pointer to QMouseEvent

16.44.2.8 resizeEvent()

```
void cImageWindowWidget::resizeEvent (
            QResizeEvent * event )  [protected]
```

Handle the resize event.

Parameters

event	pointer to QResizeEvent

16.44.2.9 setCorrelationValues()

```
void cImageWindowWidget::setCorrelationValues (
            int leftshift,
            int topshift,
            int maxx,
            int maxy,
            int pixelsizex,
            int pixelsizey )
```

Set the values to corelate data with optical image.

Parameters

leftshift	left shift
topshift	top shift
maxx	maximum X coordinate
maxy	maximum Y coordinate
pixelsizex	pixel size x
pixelsizey	pixel size y

16.44.2.10 setDefaultMaxXY()

```
void cImageWindowWidget::setDefaultMaxXY (
            int defaultmaxx,
            int defaultmaxy,
            int defaultpixelsizex,
            int defaultpixelsizey,
            eVendorType vendor )
```

Set the default maximum X and Y coordinates - parsed from the imzML file.

Parameters

defaultmaxx	maximum X coordinate
defaultmaxy	maximum Y coordinate
defaultpixelsizex	pixel size x
defaultpixelsizey	pixel size y
vendor	vendor type

16.44.2.11 setFilterOptions()

```
void cImageWindowWidget::setFilterOptions (
            vector< cCoordinateInfo > & coordinateinfo,
            bool operatortype,
            string & columnname1,
            string & comparatorname1,
            string & filterstring1,
            string & columnname2,
            string & comparatorname2,
            string & filterstring2,
            bool casesensitive,
            bool wholeword )
```

Set options which was used to filter the points.

Parameters

coordinateinfo	a vector of coordinates
operatortype	the type of operator (0 = or; 1 = and)
columnname1	name of column which was compared
comparatorname1	name of used comparator
filterstring1	a string used to filter the points
columnname2	name of column which was compared
comparatorname2	name of used comparator
filterstring2	a string used to filter the points
casesensitive	true if the string was used as a casesensitive, false otherwise
wholeword	true if whole words only are compared, false otherwise

16.44.2.12 setHistologyImage()

```
void cImageWindowWidget::setHistologyImage (
            QImage * histologyimage )
```

Set the histology image.

Parameters

histologyimage	a histology image

16.44.2.13 setHistologyPosition()

```
void cImageWindowWidget::setHistologyPosition (
            int x,
            int y,
            int width,
            int height,
            double angle )
```

Set the position of a histology image.

Parameters

x	x coordinate
y	y coordinate
width	width
height	height
angle	angle

16.44.2.14 setKeepAspectRatio()

```
void cImageWindowWidget::setKeepAspectRatio (
            bool state )
```

Set the flag to keep aspect ratio.

Parameters

state	if true, the aspect ratio is kept

16.44.2.15 setMicroscopyImage()

```
void cImageWindowWidget::setMicroscopyImage (
            int layer,
            QImage * microscopyimage )
```

Set the microscopy image.

Parameters

layer	a microscopy layer
microscopyimage	a microscopy image

16.44.2.16 setMicroscopyPosition()

```
void cImageWindowWidget::setMicroscopyPosition (
            int layer,
            bool flipx,
```

```
            bool flipy,
            double x,
            double y,
            double width,
            double height,
            double angle )
```

Set the position of a microscopy image.

Parameters

layer	a microscopy layer
flipx	flip horizontally
flipy	flip vertically
x	x offset [um]
y	y offset [um]
width	width [um]
height	height [um]
angle	angle

16.44.2.17 setNavigationLayer()

```
void cImageWindowWidget::setNavigationLayer (
            int layer,
            int navigation )
```

Set navigation layer.

Parameters

layer	current layer
navigation	navigation layer

16.44.2.18 setOpticalImage()

```
void cImageWindowWidget::setOpticalImage (
            QImage * image )
```

Set the optical image.

Parameters

image	an optical image

16.44.2.19 setSelectedRegion()

```
void cImageWindowWidget::setSelectedRegion (
            int xfrom,
            int xto,
            int yfrom,
            int yto )
```

Set the current region.

Parameters

xfrom	minimum x coordinate
xto	maximum x coordinate
yfrom	minimum y coordinate
yto	maximum y coordinate

16.44.2.20 updateCorrelation

```
void cImageWindowWidget::updateCorrelation (
            int xmin,
            int xmax,
            int ymin,
            int ymax )  [signal]
```

The signal is emitted when the points area in the optical image was changed.

Parameters

xmin	minimum x coordinate
xmax	maximum x coordinate
ymin	minimum y coordinate
ymax	maximum y coordinate

16.44.2.21 updateFilter

```
void cImageWindowWidget::updateFilter (
            int xmin,
            int xmax,
            int ymin,
            int ymax )  [signal]
```

The signal is emitted when the selection of points was changed.

Parameters

xmin	minimum x coordinate
xmax	maximum x coordinate
ymin	minimum y coordinate
ymax	maximum y coordinate

16.44.2.22 updateHistologyPosition

```
void cImageWindowWidget::updateHistologyPosition (
            int x,
            int y,
            int width,
            int height,
            double angle )  [signal]
```

The signal is emitted when the histology position was changed.

Parameters

x	x coordinate
y	y coordinate
width	width
height	height
angle	angle

16.44.2.23 updateMicroscopyPosition

```
void cImageWindowWidget::updateMicroscopyPosition (
            bool flipx,
            bool flipy,
            double x,
            double y,
            double width,
            double height,
            double angle )  [signal]
```

The signal is emitted when the microscopy position was changed.

Parameters

flipx	flip horizontally
flipy	flip vertically
x	x coordinate [um]
y	y coordinate [um]
width	width [um]
height	height [um]
angle	angle

16.44.2.24 updateRuler

```
void cImageWindowWidget::updateRuler (
            double size )  [signal]
```

The signal is emitted when the ruler size was changed.

Parameters

size	ruler size

16.44.2.25 wheelEvent()

```
void cImageWindowWidget::wheelEvent (
            QWheelEvent * event )  [protected]
```

Handle the mouse wheel event.

Parameters

event	pointer to QWheelEvent

The documentation for this class was generated from the following files:

- gui/cImageWindowWidget.h
- gui/cImageWindowWidget.cpp

16.45 cImzML Class Reference

imzML parser.
`#include <cImzML.h>`

Public Member Functions

- **cImzML** ()

  The constructor.
- ~**cImzML** ()

  The destructor.
- int parse (string &filename, int &defaultmaxx, int &defaultmaxy, int &defaultpixelsizex, int &defaultpixelsizey, eVendorType &vendor)

  Parse an imzml file.
- void updateRawDataToPeakList (vector< cPeaksList > &peaklists, string &convertedibdfilename)

  Update values after conversion of profile spectra into centroid spectra.
- vector< cImzMLItem > & getItems ()

  Get a vector of cImzML items.
- bool hasProfileSpectra ()

  Check if the file contains profile spectra.
- bool use64BitMzPrecision ()

  Check the precision of mz values.
- bool use64BitIntensityPrecision ()

  Check the precision of intensity values.
- void write (string &filename)

  Write an imzml file.

16.45.1 Detailed Description

imzML parser.

16.45.2 Member Function Documentation

16.45.2.1 getItems()

`vector< cImzMLItem > & cImzML::getItems ( )`
Get a vector of cImzML items.

Return values

vector<cImzMLItem>	a vector of cImzML items

16.45.2.2 hasProfileSpectra()

`bool cImzML::hasProfileSpectra ( )`
Check if the file contains profile spectra.

Return values

bool	true if the file contains profile spectra

16.45.2.3 parse()

```
int cImzML::parse (
            string & filename,
            int & defaultmaxx,
            int & defaultmaxy,
            int & defaultpixelsizex,
            int & defaultpixelsizey,
            eVendorType & vendor )
```

Parse an imzml file.

Parameters

filename	imzml filename
defaultmaxx	max count of pixel x - parsed from imzML file
defaultmaxy	max count of pixel y - parsed from imzML file
defaultpixelsizex	pixel size x - parsed from imzML file
defaultpixelsizey	pixel size y - parsed from imzML file
vendor	vendor type

Return values

0	= success; 1 = failed (zlib compression detected)

16.45.2.4 updateRawDataToPeakList()

```
void cImzML::updateRawDataToPeakList (
            vector< cPeaksList > & peaklists,
            string & convertedibdfilename )
```

Update values after conversion of profile spectra into centroid spectra.

Parameters

peaklists	an updated vector of peaklists
convertedibdfilename	filename of a converted ibd file

16.45.2.5 use64BitIntensityPrecision()

```
bool cImzML::use64BitIntensityPrecision ( )
```

Check the precision of intensity values.

Return values

bool	if true, 64bit precision is used; if false, 32bit precision is used

16.45.2.6 use64BitMzPrecision()

```
bool cImzML::use64BitMzPrecision ( )
```

Check the precision of mz values.

Return values

bool	if true, 64bit precision is used; if false, 32bit precision is used

16.45.2.7 write()

```
void cImzML::write (
            string & filename )
```

Write an imzml file.

Parameters

filename	an output imzml filename

The documentation for this class was generated from the following files:

- core/cImzML.h
- core/cImzML.cpp

16.46 cImzMLItem Struct Reference

A parsed imzML item.

```
#include <cImzML.h>
```

Public Member Functions

- **cImzMLItem** ()

  The constructor.

Public Attributes

- int **x**

  Image - x coordinate.
- int **y**

  Image - y coordinate.
- long long unsigned **mzstart**

  Starting offset of m/z values array in ibd file (in bytes).
- long long unsigned **mzlength**

  Length of m/z values array in ibd file (in bytes).
- long long unsigned **intensitystart**

  Starting offset of intensity values array in ibd file (in bytes).
- long long unsigned **intensitylength**

  Length of intensity values array in ibd file (in bytes).
- string **title**

  Title.

16.46.1 Detailed Description

A parsed imzML item.
The documentation for this struct was generated from the following file:

- core/clmzML.h

16.47 clonLabel Struct Reference

Auxiliary structure for visualization of labels of fragment ions.
`#include <cLinearWidget.h>`

Public Member Functions

- **clonLabel** ()

  The constructor.
- void **clear** ()

  Clear the structure.

Public Attributes

- int **x**

  x coordinate.
- int **y**

  y coordinate.
- set< string > **labelset**

  Set of labels to be merged.
- bool **alignright**

  True when the label is aligned right.

16.47.1 Detailed Description

Auxiliary structure for visualization of labels of fragment ions.
The documentation for this struct was generated from the following file:

- gui/cLinearWidget.h

16.48 clonSummaryTableKeyMS_comp Struct Reference

Comparison object for cSummaryTableKeyMS (independent ions).
`#include <cSummaryPeaksTableWidget.h>`

Public Member Functions

- bool operator() (const cSummaryTableKeyMS &left, const cSummaryTableKeyMS &right) const

  Operator to define the order of two items.

16.48.1 Detailed Description

Comparison object for cSummaryTableKeyMS (independent ions).

16.48.2 Member Function Documentation

16.48.2.1 operator()()

```
bool cIonSummaryTableKeyMS_comp::operator() (
            const cSummaryTableKeyMS & left,
            const cSummaryTableKeyMS & right ) const  [inline]
```

Operator to define the order of two items.

Parameters

left	first key
right	second key

Return values

bool	true if the first object is less than the second object

The documentation for this struct was generated from the following file:

- gui/cSummaryPeaksTableWidget.h

16.49 clonType Struct Reference

The struct for definition of a ion type.

`#include <cGlobalPreferences.h>`

Public Member Functions

- **clonType** ()

  The default constructor.
- void **clear** ()

  Clear the structure.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is)

  Load the structure from an input stream.

Public Attributes

- string **name**

  Name of the ion.
- string **formula**

  Formula of the ion.
- double **massdifference**

  Mass difference of the ion from the neutral molecule.
- bool **positive**

  Charge of the ion.
- int **multiplier**

  The multiplier n in [nM + H]+.

16.49.1 Detailed Description

The struct for definition of a ion type.

16.49.2 Member Function Documentation

16.49.2.1 load()

```
void cIonType::load (
            ifstream & is )  [inline]
```

Load the structure from an input stream.

Parameters

is	an input stream

16.49.2.2 store()

```
void cIonType::store (
            ofstream & os )  [inline]
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this struct was generated from the following file:

- core/cGlobalPreferences.h

16.50 clsotopeMap Class Reference

The class representing an isotope map.
`#include <cFragmentIons.h>`

Public Member Functions

- **clsotopeMap** ()
  The constructor.
- map< string, isotope > & operator[ ] (string &element)
  Overloaded operator [].
- int count (string &element)
  Check whether an element occurs in the table.

16.50.1 Detailed Description

The class representing an isotope map.

16.50.2 Member Function Documentation

16.50.2.1 count()

```
int cIsotopeMap::count (
            string & element )
```

Check whether an element occurs in the table.

Parameters

element	reference to a string with a name of an element

Return values

int	1 when element is presented in the table, 0 otherwise

16.50.2.2 operator[]()

```
map< string, isotope > & cIsotopeMap::operator[] (
            string & element )
```

Overloaded operator [].

Parameters

element	reference to a string with a name of an element

Return values

map	map of isotopes of the element

The documentation for this class was generated from the following files:

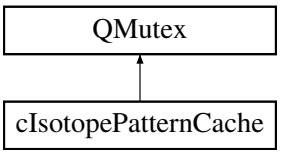
- core/cFragmentIons.h
- core/cFragmentIons.cpp

16.51 cIsotopePatternCache Class Reference

Isotope pattern cache (LRU cache is implemented).

`#include <cIsotopePatternCache.h>`

Inheritance diagram for cIsotopePatternCache:



Public Member Functions

- cIsotopePatternCache (int size=10000)

  The constructor.
- ~**cIsotopePatternCache** ()

  The destructor.
- bool get (string &formula, cPeaksList &isotopepattern)

  Get the isotope pattern corresponding to the formula from the cache.
- void put (string &formula, cPeaksList &isotopepattern)

  Put the isotope pattern corresponding to the formula into the cache.
- void **clear** ()

  Clear the cache.

16.51.1 Detailed Description

Isotope pattern cache (LRU cache is implemented).

16.51.2 Constructor & Destructor Documentation

16.51.2.1 cIsotopePatternCache()

```
cIsotopePatternCache::cIsotopePatternCache (
            int size = 10000 )
```

The constructor.

Parameters

size	maximum number of cached items

16.51.3 Member Function Documentation

16.51.3.1 get()

```
bool cIsotopePatternCache::get (
            string & formula,
            cPeaksList & isotopepattern )
```

Get the isotope pattern corresponding to the formula from the cache.

Parameters

formula	input summary formula
isotopepattern	output isotope pattern

Return values

bool	true if the isotope pattern was found; false otherwise.

16.51.3.2 put()

```
void cIsotopePatternCache::put (
            string & formula,
            cPeaksList & isotopepattern )
```

Put the isotope pattern corresponding to the formula into the cache.

Parameters

formula	input summary formula
isotopepattern	input isotope pattern

The documentation for this class was generated from the following files:

- core/cIsotopePatternCache.h
- core/cIsotopePatternCache.cpp

16.52 cLifMetadata Struct Reference

A structure to keep parsed lif metadata.
`#include <cImageWindowTabOptical.h>`

Public Member Functions

- **cLifMetadata** ()

  The constructor.
- void **print** ()

  Print metadata.

Public Attributes

- string **imagename**

  Image name.
- int **microscopytype**

  Type of microscopy technique (-1 = unknown, 0 = optical, 1 = fluorescence).
- string **microscopemodel**

  Microscope model.
- int **inversemicroscopemodel**

  Inverse microscope model.
- int **magnification**

  Magnification.
- int **flipx**

  Flip X.
- int **flipy**

  Flip Y.
- int **dimx**

  Width [pixels].
- int **dimy**

  Height [pixels].
- int **dimz**

  Depth [layers].
- int **dim10**

  Number of tiles.
- double **width**

  Width [m].
- double **height**

  Height [m].
- double **stageposx**

  Stage position x [m].
- double **stageposy**

  Stage position y [m].
- long long unsigned **memoryblocksize**

  Memory block size.
- string **memoryblockid**

  Memory block ID.
- vector< long long unsigned > **memoryblockoffsets**

  Offsets of memory blocks.
- vector< eChannelType > **channels**

  Color channels.

- vector< string > **luts**

  LUTs.
- int **tilescaninfoflipx**

  Flip X from TileScanInfo.
- int **tilescaninfoflipy**

  Flip Y from TileScanInfo.
- vector< cTileInfo > **tiles**

  Vector of tiles.

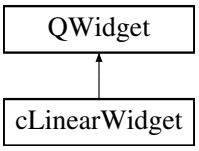
16.52.1 Detailed Description

A structure to keep parsed lif metadata.
The documentation for this struct was generated from the following file:

- gui/cImageWindowTabOptical.h

16.53 cLinearWidget Class Reference

Visualization of a linear peptide.
`#include <cLinearWidget.h>`
Inheritance diagram for cLinearWidget:



Public Member Functions

- **cLinearWidget** ()

  The constructor.
- void initialize (cParameters *parameters, cTheoreticalSpectrum *theoreticalspectrum)

  Initialize the widget.
- void exportToPDF (QString filename, bool postscript)

  Export peptide into a PDF or a PS file.
- void exportToPNG (QString filename)

  Export peptide scene into a PNG file.
- void exportToSVG (QString filename)

  Export peptide scene into a SVG file.
- void setReportIsomers (bool reportisomers)

  Set the report isomers state.
- void hidePeakLabels (bool state)

  Hide/Show labels of peaks.

Protected Member Functions

- void paintEvent (QPaintEvent *event)

  Handle the paint event.

16.53.1 Detailed Description

Visualization of a linear peptide.

16.53.2 Member Function Documentation

16.53.2.1 exportToPDF()

```
void cLinearWidget::exportToPDF (
            QString filename,
            bool postscript )
```

Export peptide into a PDF or a PS file.

Parameters

filename	filename
postscript	if true then PS file is generated instead of PDF

16.53.2.2 exportToPNG()

```
void cLinearWidget::exportToPNG (
            QString filename )
```

Export peptide scene into a PNG file.

Parameters

filename	filename

16.53.2.3 exportToSVG()

```
void cLinearWidget::exportToSVG (
            QString filename )
```

Export peptide scene into a SVG file.

Parameters

filename	filename

16.53.2.4 hidePeakLabels()

```
void cLinearWidget::hidePeakLabels (
            bool state )
```

Hide/Show labels of peaks.

Parameters

state	the peak labels are hidden when *state* is true

16.53.2.5 initialize()

```
void cLinearWidget::initialize (
            cParameters * parameters,
```

```
            cTheoreticalSpectrum * theoreticalspectrum )
```

Initialize the widget.

Parameters

parameters	a pointer to parameters
theoreticalspectrum	a pointer to a theoretical spectrum

16.53.2.6 paintEvent()

```
void cLinearWidget::paintEvent (
            QPaintEvent * event )  [protected]
```

Handle the paint event.

Parameters

event	pointer to QPaintEvent

16.53.2.7 setReportIsomers()

```
void cLinearWidget::setReportIsomers (
            bool reportisomers )
```

Set the report isomers state.

Parameters

reportisomers	true if isomers are reported; false otherwise

The documentation for this class was generated from the following files:

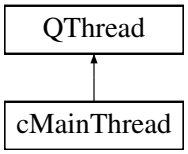
- gui/cLinearWidget.h
- gui/cLinearWidget.cpp

16.54 cMainThread Class Reference

The class representing a thread launched by the command 'Search->Run'.

`#include <cMainThread.h>`

Inheritance diagram for cMainThread:



Public Types

- typedef std::basic_ostream< char, std::char_traits< char > > **CoutType**

  This is the type of std::cout.
- typedef CoutType &(* **StandardEndLine**) (CoutType &)

  This is the function signature of std::endl.

Signals

- void message (QString s)

  A message which will be shown in log window.
- void enableRunButtonAndSettings (bool enable)

  Enable or disable options 'Search->Run' and 'Search->Settings...'.
- void enableStopButton (bool enable)

  Enable or disable the option 'Search->Stop'.
- void enableButtonsHandlingResults (bool enable)

  Enable or disable buttons handling results.
- void sendParameters (cParameters parameters)

  Send parameters to cMainWindow when the thread has finished.
- void **reportSpectra** ()

  Report results.
- void setGraph (string s)

  Set the graph into the cGraphWidget.

Public Member Functions

- cMainThread (cParameters &parameters, cTheoreticalSpectrumList &listoftheoreticalspectra, bool enablelogwindow=true, bool enablestdout=true)

  The constructor.
- void setGraphReaderIsWorking (bool working)

  Set the flag that the graph reader is working.
- bool isGraphReaderWorking ()

  Check the flag whether the graph reader is working.
- template<class T >
  cMainThread & **operator**<< (const T &x)

  Overloaded operator <<.
- cMainThread & **operator**<< (const char *x)

  Overloaded operator <<.
- cMainThread & **operator**<< (const string &x)

  Overloaded operator <<.
- cMainThread & **operator**<< (StandardEndLine manip)

  Define an operator << to take in std::endl.
- void addToParallelOutputState (int value)

  Add a value to the output state (for internal threads).

Protected Member Functions

- void **run** ()

  The main method of the thread.
- void **emitEndSignals** ()

  The signals emitted when the thread has successfully finished.

16.54.1 Detailed Description

The class representing a thread launched by the command 'Search->Run'.

16.54.2 Constructor & Destructor Documentation

16.54.2.1 cMainThread()

```
cMainThread::cMainThread (
            cParameters & parameters,
            cTheoreticalSpectrumList & listoftheoreticalspectra,
            bool enablelogwindow = true,
            bool enablestdout = true )
```

The constructor.

Parameters

parameters	reference to input paramaters
listoftheoreticalspectra	lists of theoretical spectra
enablelogwindow	if true then messages are logged into the log window
enablestdout	if true then messages are logged into the standard output

16.54.3 Member Function Documentation

16.54.3.1 addToParallelOutputState()

```
void cMainThread::addToParallelOutputState (
            int value )
```

Add a value to the output state (for internal threads).

Parameters

value	value

16.54.3.2 enableButtonsHandlingResults

```
void cMainThread::enableButtonsHandlingResults (
            bool enable )  [signal]
```

Enable or disable buttons handling results.

Parameters

enable	true == enable, false == disable

16.54.3.3 enableRunButtonAndSettings

```
void cMainThread::enableRunButtonAndSettings (
            bool enable )  [signal]
```

Enable or disable options 'Search->Run' and 'Search->Settings...'.

Parameters

enable	true == enable, false == disable

16.54.3.4 enableStopButton

```
void cMainThread::enableStopButton (
            bool enable )  [signal]
```

Enable or disable the option 'Search->Stop'.

Parameters

enable	true == enable, false == disable

16.54.3.5 isGraphReaderWorking()

```
bool cMainThread::isGraphReaderWorking ( )
```

Check the flag whether the graph reader is working.

Return values

bool	true when the graph reader is working, false otherwise

16.54.3.6 message

```
void cMainThread::message (
            QString s )  [signal]
```

A message which will be shown in log window.

Parameters

s	text of the message

16.54.3.7 sendParameters

```
void cMainThread::sendParameters (
            cParameters parameters )  [signal]
```

Send parameters to cMainWindow when the thread has finished.

Parameters

parameters	parameters

16.54.3.8 setGraph

```
void cMainThread::setGraph (
            string s )  [signal]
```

Set the graph into the cGraphWidget.

Parameters

s	graph printed as a string

16.54.3.9 setGraphReaderIsWorking()

```
void cMainThread::setGraphReaderIsWorking (
            bool working )
```

Set the flag that the graph reader is working.

Parameters

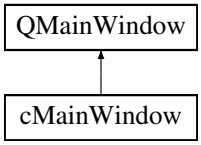
working	set true when the graph reader is working, otherwise set false

The documentation for this class was generated from the following files:

- gui/cMainThread.h
- gui/cMainThread.cpp

16.55 cMainWindow Class Reference

The class representing the main window of the application.
`#include <cMainWindow.h>`
Inheritance diagram for cMainWindow:



Signals

- void **stopComputation** ()

  The option 'Search->Stop' has been selected.

Public Member Functions

- **cMainWindow** ()

  The constructor.
- ~**cMainWindow** ()

  The destructor.

Protected Member Functions

- void keyPressEvent (QKeyEvent *event)

  Handle a key press event.

16.55.1 Detailed Description

The class representing the main window of the application.

16.55.2 Member Function Documentation

16.55.2.1 keyPressEvent()

```
void cMainWindow::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

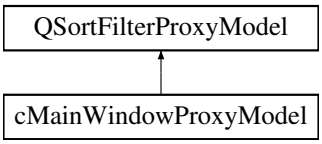
event	pointer to QKeyEvent

The documentation for this class was generated from the following files:

- gui/cMainWindow.h
- gui/cMainWindow.cpp

16.56 cMainWindowProxyModel Class Reference

A table proxy model.
`#include <cMainWindowProxyModel.h>`
Inheritance diagram for cMainWindowProxyModel:



Public Member Functions

- cMainWindowProxyModel (QObject ∗parent=0)

  The constructor.
- void initialize (QComboBox ∗rowsfilteroperator, QComboBox ∗rowsfiltercombobox1, QComboBox ∗rowsfiltercomparatorcombobox1, QLineEdit ∗rowsfilterline1, QComboBox ∗rowsfiltercombobox2, QComboBox ∗rowsfiltercomparatorcombobox2, QLineEdit ∗rowsfilterline2)

  Initialize the model.
- void setWholeWord (bool wholeword)

  Set whole word option.

Protected Member Functions

- bool filterAcceptsRow (int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE

  Reimplementation of filterAcceptsRow.
- bool lessThan (const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE

  Reimplementation of lessThan.

16.56.1 Detailed Description

A table proxy model.

16.56.2 Constructor & Destructor Documentation

16.56.2.1 cMainWindowProxyModel()

```
cMainWindowProxyModel::cMainWindowProxyModel (
            QObject * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.56.3 Member Function Documentation

16.56.3.1 filterAcceptsRow()

```
bool cMainWindowProxyModel::filterAcceptsRow (
            int sourceRow,
            const QModelIndex & sourceParent ) const  [protected]
```

Reimplementation of filterAcceptsRow.

Parameters

sourceRow	int
sourceParent	QModelIndex

Return values

bool	bool

16.56.3.2 initialize()

```
void cMainWindowProxyModel::initialize (
            QComboBox * rowsfilteroperator,
            QComboBox * rowsfiltercombobox1,
            QComboBox * rowsfiltercomparatorcombobox1,
            QLineEdit * rowsfilterline1,
            QComboBox * rowsfiltercombobox2,
            QComboBox * rowsfiltercomparatorcombobox2,
            QLineEdit * rowsfilterline2 )
```

Initialize the model.

Parameters

rowsfilteroperator	reference to the combobox with operator selection
rowsfiltercombobox1	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox1	reference to the combobox with selection of comparator
rowsfilterline1	reference to a text field
rowsfiltercombobox2	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox2	reference to the combobox with selection of comparator
rowsfilterline2	reference to a text field

16.56.3.3 lessThan()

```
bool cMainWindowProxyModel::lessThan (
            const QModelIndex & left,
            const QModelIndex & right ) const  [protected]
```

Reimplementation of lessThan.

Parameters

left	QModelIndex
right	QModelIndex

Return values

bool	bool

16.56.3.4 setWholeWord()

```
void cMainWindowProxyModel::setWholeWord (
            bool wholeword )
```

Set whole word option.

Parameters

wholeword	true if whole words are matched, false otherwise

The documentation for this class was generated from the following files:

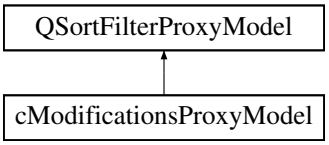
- gui/cMainWindowProxyModel.h
- gui/cMainWindowProxyModel.cpp

16.57 cModificationsProxyModel Class Reference

A proxy model for editor of database of modifications.

`#include <cModificationsProxyModel.h>`

Inheritance diagram for cModificationsProxyModel:



Public Member Functions

- cModificationsProxyModel (QObject ∗parent=0)

  The constructor.
- void initialize (QComboBox ∗rowsfilteroperator, QComboBox ∗rowsfiltercombobox1, QComboBox ∗rowsfiltercomparatorcombobox1, QLineEdit ∗rowsfilterline1, QComboBox ∗rowsfiltercombobox2, QComboBox ∗rowsfiltercomparatorcombobox2, QLineEdit ∗rowsfilterline2)

  Initialize the model.
- void setWholeWord (bool wholeword)

  Set whole word option.

Protected Member Functions

- bool filterAcceptsRow (int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE

  Reimplementation of filterAcceptsRow.
- bool lessThan (const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE

  Reimplementation of lessThan.

16.57.1 Detailed Description

A proxy model for editor of database of modifications.

16.57.2 Constructor & Destructor Documentation

16.57.2.1 cModificationsProxyModel()

```
cModificationsProxyModel::cModificationsProxyModel (
            QObject * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.57.3 Member Function Documentation

16.57.3.1 filterAcceptsRow()

```
bool cModificationsProxyModel::filterAcceptsRow (
            int sourceRow,
            const QModelIndex & sourceParent ) const  [protected]
```

Reimplementation of filterAcceptsRow.

Parameters

sourceRow	int
sourceParent	QModelIndex

Return values

bool	bool

16.57.3.2 initialize()

```
void cModificationsProxyModel::initialize (
            QComboBox * rowsfilteroperator,
            QComboBox * rowsfiltercombobox1,
            QComboBox * rowsfiltercomparatorcombobox1,
            QLineEdit * rowsfilterline1,
            QComboBox * rowsfiltercombobox2,
            QComboBox * rowsfiltercomparatorcombobox2,
            QLineEdit * rowsfilterline2 )
```

Initialize the model.

Parameters

rowsfilteroperator	reference to the combobox with operator selection
rowsfiltercombobox1	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox1	reference to the combobox with selection of comparator
rowsfilterline1	reference to a text field
rowsfiltercombobox2	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox2	reference to the combobox with selection of comparator

Parameters

rowsfilterline2	reference to a text field

16.57.3.3 lessThan()

```
bool cModificationsProxyModel::lessThan (
            const QModelIndex & left,
            const QModelIndex & right ) const  [protected]
```

Reimplementation of lessThan.

Parameters

left	QModelIndex
right	QModelIndex

Return values

bool	bool

16.57.3.4 setWholeWord()

```
void cModificationsProxyModel::setWholeWord (
            bool wholeword )
```

Set whole word option.

Parameters

wholeword	true if whole words are matched, false otherwise

The documentation for this class was generated from the following files:

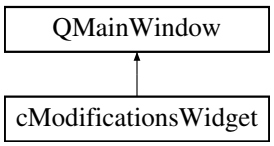
- gui/cModificationsProxyModel.h
- gui/cModificationsProxyModel.cpp

16.58 cModificationsWidget Class Reference

An editor of N-/C-terminal modifications database.

`#include <cModificationsWidget.h>`

Inheritance diagram for cModificationsWidget:



Public Member Functions

- cModificationsWidget (cGlobalPreferences *globalpreferences, QWidget *parent=(QWidget *) 0)

  The constructor.

- ~**cModificationsWidget** ()

  The destructor.
- void closeEvent (QCloseEvent *event)

  Handle the window close event.
- void applyGlobalPreferences (cGlobalPreferences *globalpreferences)

  Apply new global preferences.

Protected Member Functions

- void keyPressEvent (QKeyEvent *event)

  Handle a key press event.

16.58.1 Detailed Description

An editor of N-/C-terminal modifications database.

16.58.2 Constructor & Destructor Documentation

16.58.2.1 cModificationsWidget()

```
cModificationsWidget::cModificationsWidget (
            cGlobalPreferences * globalpreferences,
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.58.3 Member Function Documentation

16.58.3.1 applyGlobalPreferences()

```
void cModificationsWidget::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.58.3.2 closeEvent()

```
void cModificationsWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.58.3.3 keyPressEvent()

```
void cModificationsWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

The documentation for this class was generated from the following files:

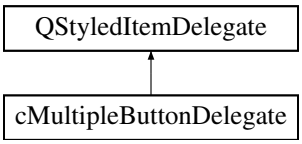
- gui/cModificationsWidget.h
- gui/cModificationsWidget.cpp

16.59 cMultipleButtonDelegate Class Reference

A delegate to create multiple buttons.

`#include <cMultipleButtonDelegate.h>`

Inheritance diagram for cMultipleButtonDelegate:



Public Member Functions

- cMultipleButtonDelegate (QObject *parent=(QObject *) 0)
  The constructor.
- QString displayText (const QVariant &value, const QLocale &locale) const
  Reimplementation of displayText.
- void paint (QPainter *painter, const QStyleOptionViewItem &option, const QModelIndex &index) const
  Reimplementation of paint.

Protected Member Functions

- bool editorEvent (QEvent *event, QAbstractItemModel *model, const QStyleOptionViewItem &option, const QModelIndex &index)
  Reimplementation of editorEvent.

16.59.1 Detailed Description

A delegate to create multiple buttons.

16.59.2 Constructor & Destructor Documentation

16.59.2.1 cMultipleButtonDelegate()

```
cMultipleButtonDelegate::cMultipleButtonDelegate (
            QObject * parent = (QObject *)0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.59.3 Member Function Documentation

16.59.3.1 displayText()

```
QString cMultipleButtonDelegate::displayText (
            const QVariant & value,
            const QLocale & locale ) const
```

Reimplementation of displayText.

Parameters

value	a value
locale	a locale

Return values

QString	the value as a string

16.59.3.2 editorEvent()

```
bool cMultipleButtonDelegate::editorEvent (
            QEvent * event,
            QAbstractItemModel * model,
            const QStyleOptionViewItem & option,
            const QModelIndex & index )  [protected]
```

Reimplementation of editorEvent.

Parameters

event	QEvent
model	QAbstractItemModel
option	QStyleOptionViewItem
index	QModelIndex

16.59.3.3 paint()

```
void cMultipleButtonDelegate::paint (
            QPainter * painter,
            const QStyleOptionViewItem & option,
            const QModelIndex & index ) const
```

Reimplementation of paint.

Parameters

painter	QPainter

Parameters

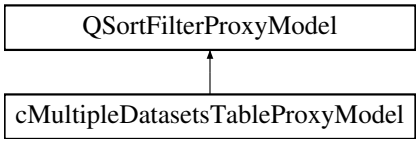
option	QStyleOptionViewItem
index	QModelIndex

The documentation for this class was generated from the following files:

- gui/cMultipleButtonDelegate.h
- gui/cMultipleButtonDelegate.cpp

16.60 cMultipleDatasetsTableProxyModel Class Reference

A table proxy model.
`#include <cMultipleDatasetsTableProxyModel.h>`
Inheritance diagram for cMultipleDatasetsTableProxyModel:



Public Member Functions

- cMultipleDatasetsTableProxyModel (QObject *parent=0)

  The constructor.

Protected Member Functions

- bool lessThan (const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE

  Reimplementation of lessThan.

16.60.1 Detailed Description

A table proxy model.

16.60.2 Constructor & Destructor Documentation

16.60.2.1 cMultipleDatasetsTableProxyModel()

```
cMultipleDatasetsTableProxyModel::cMultipleDatasetsTableProxyModel (
            QObject * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.60.3 Member Function Documentation

16.60.3.1 lessThan()

```
bool cMultipleDatasetsTableProxyModel::lessThan (
            const QModelIndex & left,
            const QModelIndex & right ) const  [protected]
```

Reimplementation of lessThan.

Parameters

left	QModelIndex
right	QModelIndex

Return values

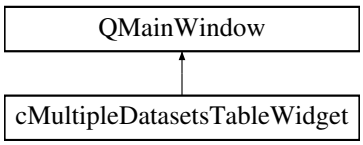
bool	bool

The documentation for this class was generated from the following files:

- gui/cMultipleDatasetsTableProxyModel.h
- gui/cMultipleDatasetsTableProxyModel.cpp

16.61 cMultipleDatasetsTableWidget Class Reference

Aggregated results from multiple datasets.
`#include <cMultipleDatasetsTableWidget.h>`
Inheritance diagram for cMultipleDatasetsTableWidget:



Signals

- void **tableCancelled** ()

  The table was not generated because the process was cancelled.
- void **filterSummaryTable** ()

  Update chromatogram via summary table of matched peaks.
- void multipleDatasetsTableRowClicked (int datasetid, int spectrumid, int peakid, int datatypeview, int peakshape, string name, string iontype, string mzstr)

  Process a row click event.

Public Member Functions

- cMultipleDatasetsTableWidget (cGlobalPreferences ∗globalpreferences, QWidget ∗parent=(QWidget ∗) 0)

  The constructor.
- ∼**cMultipleDatasetsTableWidget** ()

  The destructor.
- void closeEvent (QCloseEvent ∗event)

  Handle the window close event.
- bool prepareToShow (cParameters ∗parameters, cTheoreticalSpectrumList ∗listoftheoreticalspectra, int activefileid)

  Prepare the widget to show.

- void **updateTable** ()

  Update the table if attributes were changed.
- void **resetStandard** ()

  Reset the compound standard.
- void **updateListofStandards** ()

  Update the list of compound standards.
- void **deleteTable** ()

  Delete the table content.
- void applyGlobalPreferences (cGlobalPreferences ∗globalpreferences)

  Apply new global preferences.
- cPubChemSearchWidget ∗ getPubChemSearchWidget ()

  Get the reference to cPubChemSearchWidget.
- cDefineCalibrationWidget ∗ getDefineCalibrationWidget ()

  Get the reference to cDefineCalibrationWidget.
- cCalibrationChart ∗ getCalibrationChartWidget ()

  Get the reference to cCalibrationChart widget.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.

16.61.1 Detailed Description

Aggregated results from multiple datasets.

16.61.2 Constructor & Destructor Documentation

16.61.2.1 cMultipleDatasetsTableWidget()

```
cMultipleDatasetsTableWidget::cMultipleDatasetsTableWidget (
            cGlobalPreferences * globalpreferences,
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.61.3 Member Function Documentation

16.61.3.1 applyGlobalPreferences()

```
void cMultipleDatasetsTableWidget::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.61.3.2 closeEvent()

```
void cMultipleDatasetsTableWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.61.3.3 getCalibrationChartWidget()

```
cCalibrationChart * cMultipleDatasetsTableWidget::getCalibrationChartWidget ( )
```

Get the reference to cCalibrationChart widget.

Return values

cCalibrationChart	reference to cCalibrationChart

16.61.3.4 getDefineCalibrationWidget()

```
cDefineCalibrationWidget * cMultipleDatasetsTableWidget::getDefineCalibrationWidget ( )
```

Get the reference to cDefineCalibrationWidget.

Return values

cDefineCalibrationWidget	reference to cDefineCalibrationWidget

16.61.3.5 getPubChemSearchWidget()

```
cPubChemSearchWidget * cMultipleDatasetsTableWidget::getPubChemSearchWidget ( )
```

Get the reference to cPubChemSearchWidget.

Return values

cPubChemSearchWidget	reference to cPubChemSearchWidget

16.61.3.6 keyPressEvent()

```
void cMultipleDatasetsTableWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.61.3.7 multipleDatasetsTableRowClicked

```
void cMultipleDatasetsTableWidget::multipleDatasetsTableRowClicked (
            int datasetid,
            int spectrumid,
            int peakid,
            int datatypeview,
            int peakshape,
            string name,
            string iontype,
            string mzstr )  [signal]
```

Process a row click event.

Parameters

datasetid	id of a dataset
spectrumid	id of a spectrum
peakid	id of a peak
datatypeview	type of data shown
peakshape	type of chromatographic peak
name	compound name
iontype	ion type
mzstr	theoretical m/z value converted to a string

16.61.3.8 prepareToShow()

```
bool cMultipleDatasetsTableWidget::prepareToShow (
            cParameters * parameters,
            cTheoreticalSpectrumList * listoftheoreticalspectra,
            int activefileid )
```

Prepare the widget to show.

Parameters

parameters	parameters of the application
listoftheoreticalspectra	lists of theoretical spectra
activefileid	id of the active file

Return values

bool	true if the table was successfully prepared, false otherwise

The documentation for this class was generated from the following files:

- gui/cMultipleDatasetsTableWidget.h
- gui/cMultipleDatasetsTableWidget.cpp

16.62 cMzML Class Reference

mzML parser.

`#include <cMzML.h>`

Public Member Functions

- **cMzML** ()

  The constructor.
- ∼**cMzML** ()

  The destructor.
- int parse (string &filename, vector< cPeaksList > &peaklists, int profilespectrumid, eModeType mode, cMainThread *os, bool &terminatecomputation)

  Parse a mzml file.
- bool hasProfileSpectra ()

  Check if the file contains profile spectra.

16.62.1 Detailed Description

mzML parser.

16.62.2 Member Function Documentation

16.62.2.1 hasProfileSpectra()

```
bool cMzML::hasProfileSpectra ( )
```

Check if the file contains profile spectra.

Return values

bool	true if the file contains profile spectra

16.62.2.2 parse()

```
int cMzML::parse (
            string & filename,
            vector< cPeaksList > & peaklists,
            int profilespectrumid,
            eModeType mode,
            cMainThread * os,
            bool & terminatecomputation )
```

Parse a mzml file.

Parameters

filename	mzml filename
peaklists	mzml filename
profilespectrumid	id of a profile spectrum to be extracted, -1 = extract all profile spectra
mode	program mode
os	pointer to the main thread of the application (output stream)
terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

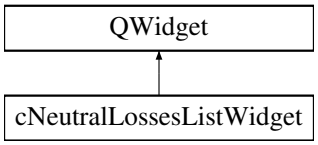
0	= success; -1 = aborted by user

The documentation for this class was generated from the following files:

- core/cMzML.h
- core/cMzML.cpp

16.63 cNeutralLossesListWidget Class Reference

The widget for selection of types of neutral losses.
`#include <cNeutralLossesListWidget.h>`
Inheritance diagram for cNeutralLossesListWidget:



Public Member Functions

- cNeutralLossesListWidget (bool showdefaultbutton, bool showhconbutton, QObject ∗parent)

  The constructor.
- ∼**cNeutralLossesListWidget** ()

  The destructor.
- QListWidget ∗ getList ()

  Access to QListWidget.
- void addItem (QString formula)

  Add a neutral loss.

16.63.1 Detailed Description

The widget for selection of types of neutral losses.

16.63.2 Constructor & Destructor Documentation

16.63.2.1 cNeutralLossesListWidget()

```
cNeutralLossesListWidget::cNeutralLossesListWidget (
            bool showdefaultbutton,
            bool showhconbutton,
            QObject * parent )
```

The constructor.

Parameters

showdefaultbutton	if true, the default button is shown
showhconbutton	if true, the HCON button is shown
parent	pointer to a parent widget

16.63.3 Member Function Documentation

16.63.3.1 addItem()

```
void cNeutralLossesListWidget::addItem (
            QString formula )
```

Add a neutral loss.

Parameters

formula	summary formula of a neutral loss

16.63.3.2 getList()

```
QListWidget * cNeutralLossesListWidget::getList ( )
```

Access to QListWidget.

Return values

QListWidget	pointer to QListWidget

The documentation for this class was generated from the following files:

- gui/cNeutralLossesListWidget.h
- gui/cNeutralLossesListWidget.cpp

16.64 compareStringBySize Struct Reference

Compare items in a string vector by size.

```
#include <utilities.h>
```

Public Member Functions

- bool operator() (const std::string &first, const std::string &second)

  Comparison function.

16.64.1 Detailed Description

Compare items in a string vector by size.

16.64.2 Member Function Documentation

16.64.2.1 operator()()

```
bool compareStringBySize::operator() (
            const std::string & first,
            const std::string & second )  [inline]
```

Comparison function.

Parameters

first	first string
second	second string

Return values

bool	true if the first string is shorter

The documentation for this struct was generated from the following file:

- core/utilities.h

16.65 cParameters Class Reference

The class storing parameters of the application.
`#include <cParameters.h>`

Public Member Functions

- **cParameters** ()

  The constructor.
- void **clear** ()

  Clear the structure.
- int checkAndPrepare (bool &terminatecomputation)

  Test restrictions of parameters and prepare some internal parameters.
- int prepareLossesAndCompounds (bool &terminatecomputation)

  Prepare neutral losses and compounds.
- bool checkModifications (cSequence &sequence, int &startmodifid, int &endmodifid, int &middlemodifid, string &errormessage)

  Check if the names of modifications used in a sequence are correct.
- string printToString ()

  Print parameters to a string.
- void setOutputStream (cMainThread &os)

  Set the ouput stream.
- void **updateFragmentDefinitions** ()

  Update definitions of fragment ions when cyclicnterminus, cyclicctterminus or precursoradduct have been changed.
- int calculateNeutralLosses (bool &terminatecomputation, string &errormessage)

  Calculate a list of neutral losses.
- int generateCompounds (bool &terminatecomputation, string &errormessage)

  Generate compounds.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is, int fileversionpart1, int fileversionpart2, int fileversionpart3)

  Load the structure from an input stream.

Public Attributes

- cFragmentIons **iondefinitions**

  Definitions of ion types.
- ePeptideType **peptidetype**

  The type of analyzed peptide.
- vector< string > **peaklistfilenames**

  The filenames of peaklists.
- string **originalpeaklistfilenames**

  Original input line with peaklist(s).
- bool **useprofiledata**

Use profile data.

- bool **convertprofiledata**

  Convert profile data.
- string **profiledatafilename**

  Profile data file name.
- int **linebafprocessing**

  Line spectra processing method (baf).
- vector< ePeakListFileFormat > **peaklistfileformats**

  The file formats of peaklists.
- vector< cPeakListSeries > **peaklistseriesvector**

  A structure representing a vector of series of peaklists.
- int **scannumber**

  Scan number.
- double **precursormass**

  Precursor mass of the peptide.
- string **precursoradduct**

  An adduct of a precursor ion.
- vector< string > **metaladducts**

  A list of parsed metal adducts from precursoradduct.
- double **precursormasserrortolerance**

  Precursor mass error tolerance (ppm).
- int **precursorcharge**

  The charge of the precursor ion.
- double **fragmentmasserrortolerance**

  Fragment mass error tolerance (ppm).
- double **minimumrelativeintensitythreshold**

  Minimum relative intensity threshold.
- unsigned **minimumabsoluteintensitythreshold**

  Minimum absolute intensity threshold.
- double **minimummz**

  Minimum m/z ratio.
- double **maximummz**

  Maximum m/z ratio.
- double **minimumrt**

  Minimum retention time.
- double **maximumrt**

  Maximum retention time.
- double **fwhm**

  FWHM.
- bool **enableratio54Fe56Fe**

  Enable 54Fe/56Fe ratio.
- double **minratio54Fe56Fe**

  Minimum ratio 54Fe/56Fe.
- double **maxratio54Fe56Fe**

  Maximum ratio 54Fe/56Fe.
- bool **enableratio60Ni58Ni**

  Enable 60Ni/58Ni ratio.
- double **minratio60Ni58Ni**

  Minimum ratio 60Ni/58Ni.
- double **maxratio60Ni58Ni**

  Maximum ratio 60Ni/58Ni.

- bool **enableratio62Ni58Ni**

  Enable 62Ni/58Ni ratio.
- double **minratio62Ni58Ni**

  Minimum ratio 62Ni/58Ni.
- double **maxratio62Ni58Ni**

  Maximum ratio 62Ni/58Ni.
- bool **enableratio65Cu63Cu**

  Enable 65Cu/63Cu ratio.
- double **minratio65Cu63Cu**

  Minimum ratio 65Cu/63Cu.
- double **maxratio65Cu63Cu**

  Maximum ratio 65Cu/63Cu.
- bool **enableratio66Zn64Zn**

  Enable 66Zn/64Zn ratio.
- double **minratio66Zn64Zn**

  Minimum ratio 66Zn/64Zn.
- double **maxratio66Zn64Zn**

  Maximum ratio 66Zn/64Zn.
- bool **enableratio67Zn64Zn**

  Enable 67Zn/64Zn ratio.
- double **minratio67Zn64Zn**

  Minimum ratio 67Zn/64Zn.
- double **maxratio67Zn64Zn**

  Maximum ratio 67Zn/64Zn.
- bool **enableratio68Zn64Zn**

  Enable 68Zn/64Zn ratio.
- double **minratio68Zn64Zn**

  Minimum ratio 68Zn/64Zn.
- double **maxratio68Zn64Zn**

  Maximum ratio 68Zn/64Zn.
- string **bricksdatabasefilename**

  A file name of a bricks database.
- cBricksDatabase **bricksdatabase**

  A structure representing a database of bricks.
- int **maximumbricksincombinationbegin**

  The maximum number of bricks in combination (at the beginning of a spectrum).
- int **maximumbricksincombinationmiddle**

  The maximum number of bricks in combination (in the middle of a spectrum).
- int **maximumbricksincombinationend**

  The maximum number of bricks in combination (at the end of a spectrum).
- int **maximumbricksincombination**

  The global maximum number of bricks in combination (maximum from maximumbricksincombinationbegin, maximumbricksincombinationmiddle and maximumbricksincombinationend).
- double **maximumcumulativemass**

  The maximum cummulative mass of bricks.
- string **modificationsfilename**

  A file with modifications.
- int **maximumnumberofthreads**

  Maximum number of threads.
- eModeType **mode**

  Program mode.

- eScoreType **scoretype**

  A type of score.
- int **maximumcombinedlosses**

  Maximum number of combined neutral losses.
- int **maximumcombinedelements**

  Maximum number of combined chemical elements.
- bool reportunmatchedtheoreticalpeaks

  True when matches of peaks which do not have parent peaks are removed.
- bool **generateisotopepattern**

  True when complete isotope pattern is generated.
- int **minimumpatternsize**

  The minimum number of peaks in a matched pattern.
- int **minimumfeaturesize**

  The minimum number of consecutive scans in which a compound must be identified to be reported (LC-MS data only).
- int **minimumiontypes**

  The minimum number of ion types which must be matched to report a given compound.
- bool **basicformulacheck**

  Apply Senior's filtering rules.
- bool **advancedformulacheck**

  Advanced filtering rules are aplied if compounds are generated in MS mode.
- bool **noratiocheck**

  Check N/O ratio.
- bool **calculatefdrs**

  Calculate FDRs.
- double **minimumannotationintensityrelative**

  Minimum relative intensity of the most intense peak in an isotopic pattern.
- unsigned **minimumannotationintensityabsolute**

  Minimum absolute intensity of the most intense peak in an isotopic pattern.
- double **mzdifftolerance**

  Maximum m/z tolerance of matched isotopes.
- double **intensitytolerance**

  Maximum tolerance of intensities of matched isotopes.
- bool **cyclicnterminus**

  True if the N-terminus of a linear peptide is cyclized.
- bool **cycliccterminus**

  True if the C-terminus of a linear peptide is cyclized.
- bool **internalfragments**

  True if the internal fragments are generated.
- bool **enablescrambling**

  True if the detection of scrambled ions is enabled.
- bool **similaritysearch**

  True if the similarity search is enabled.
- bool **regularblocksorder**

  True if the order of polyketide blocks is regularly changed.
- int **hitsreported**

  Maximum number of reported hits.
- string **sequencetag**

  A sequence tag.
- string **originalsequencetag**

  A copy of sequence tag.

- string **searchedsequence**

  A searched sequence.
- string **originalsearchedsequence**

  A copy of searched sequence.
- string **searchedsequenceNtermmodif**

  N-terminal modification of a searched peptide.
- string **searchedsequenceCtermmodif**

  C-terminal modification of a searched peptide.
- string **searchedsequenceTmodif**

  Terminal modification of a branch of searched branched peptide.
- string **searchedsequenceformula**

  Formula of searched sequence.
- int **maximumnumberofcandidates**

  Maximum number of peptide sequence candidates.
- int **blindedges**

  The identifier of an action with edges forming incomplete paths.
- string **sequencedatabasefilename**

  A file name of a sequence database.
- cSequenceDatabase **sequencedatabase**

  A structure representing a database of sequences.
- vector< cFragmentIonType > **searchedmodifications**

  A vector of searched N-terminal and C-terminal modifications.
- vector< eFragmentIonType > **fragmentionsfordenovograph**

  A vector of fragment ion types generated in the de novo graph.
- vector< cIonType > **ionsfortheoreticalspectraMS1**

  A vector of ion types generated in theoretical conventional mass spectra.
- vector< eFragmentIonType > **ionsfortheoreticalspectraMS2**

  A vector of ion types generated in theoretical fragment ion spectra.
- vector< neutralLoss > **neutrallossesdefinitions**

  A vector of all neutral losses.
- vector< neutralLoss > **originalneutrallossesdefinitions**

  A vector of all neutral losses (without combinations of losses).
- vector< int > **neutrallossesfortheoreticalspectra**

  A vector of neutral losses generated in theoretical spectra.
- int **numberofgeneratedneutrallosses**

  A number of generated combinations of neutral losses.
- vector< int > **originalneutrallossesfortheoreticalspectra**

  A vector of neutral losses generated in theoretical spectra (without combinations).
- vector< neutralLoss > **originalelementsdefinitions**

  A vector of all elements (without combinations).
- vector< int > **originalelementsfortheoreticalspectra**

  A vector of elements used when generating compounds (without combinations).
- vector< string > **peakidtodesc**

  A vector of peak descriptions.
- vector< string > **isotopeformulaidtodesc**

  A vector of isotope formula descriptions.
- map< string, int > **pchemresults**

  An auxiliary map to store search results from PubChem in Compound Search mode.
- int **defaultmaxx**

  Max count of pixel x - parsed from imzML file.
- int **defaultmaxy**

Max count of pixel y - parsed from imzML file.

- int **defaultpixelsizex**

  Pixel size x - parsed from imzML file.

- int **defaultpixelsizey**

  Pixel size y - parsed from imzML file.

- eVendorType **vendor**

  Vendor type.

16.65.1 Detailed Description

The class storing parameters of the application.

16.65.2 Member Function Documentation

16.65.2.1 calculateNeutralLosses()

```
int cParameters::calculateNeutralLosses (
            bool & terminatecomputation,
            string & errormessage )
```

Calculate a list of neutral losses.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped
errormessage	an error message if failed

Return values

int	-1 when an error occurred, 0 otherwise

16.65.2.2 checkAndPrepare()

```
int cParameters::checkAndPrepare (
            bool & terminatecomputation )
```

Test restrictions of parameters and prepare some internal parameters.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	-1 when an error occurred, 0 otherwise

16.65.2.3 checkModifications()

```
bool cParameters::checkModifications (
            cSequence & sequence,
            int & startmodifid,
```

```
            int & endmodifid,
            int & middlemodifid,
            string & errormessage )
```

Check if the names of modifications used in a sequence are correct.

Parameters

sequence	an input sequence
startmodifid	an output identifier of a start modification
endmodifid	an output identifier of an end modification
middlemodifid	an output identifier of a branch modification
errormessage	an output error message if failed

Return values

bool	true if success

16.65.2.4 generateCompounds()

```
int cParameters::generateCompounds (
            bool & terminatecomputation,
            string & errormessage )
```

Generate compounds.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped
errormessage	an error message if failed

Return values

int	-1 when an error occurred, 0 otherwise

16.65.2.5 load()

```
void cParameters::load (
            ifstream & is,
            int fileversionpart1,
            int fileversionpart2,
            int fileversionpart3 )
```

Load the structure from an input stream.

Parameters

is	an input stream
fileversionpart1	first number of .res the file version
fileversionpart2	second number of .res the file version
fileversionpart3	third number of .res the file version

16.65.2.6 prepareLossesAndCompounds()

```
int cParameters::prepareLossesAndCompounds (
            bool & terminatecomputation )
```

Prepare neutral losses and compounds.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	-1 when an error occurred, 0 otherwise

16.65.2.7 printToString()

```
string cParameters::printToString ( )
```

Print parameters to a string.

Return values

string	string with parameters

16.65.2.8 setOutputStream()

```
void cParameters::setOutputStream (
            cMainThread & os )
```

Set the ouput stream.

Parameters

os	reference to the main thread of the application (output stream)

16.65.2.9 store()

```
void cParameters::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

16.65.3 Member Data Documentation

16.65.3.1 reportunmatchedtheoreticalpeaks

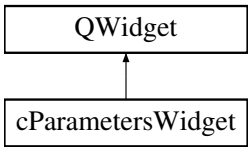
```
bool cParameters::reportunmatchedtheoreticalpeaks
```

True when matches of peaks which do not have parent peaks are removed.
Report unmatched theoretical peaks.
The documentation for this class was generated from the following files:

- core/cParameters.h
- core/cParameters.cpp

16.66 cParametersWidget Class Reference

The class representing a widget for editing parameters of the application.
`#include <cParametersWidget.h>`
Inheritance diagram for cParametersWidget:



Signals

- void sendSequenceLine (int peptidetypeindex, QString sequence)

  Send index of peptide type and searched sequence.
- void **setHCON** ()

  Set list of neutral losses to HCON.
- void **clearIonSelection** ()

  Clear the selection of ions.

Public Member Functions

- cParametersWidget (cGlobalPreferences ∗globalpreferences, QWidget ∗parent=(QWidget ∗) 0)

  The constructor.
- ∼**cParametersWidget** ()

  The destructor.
- cParameters & getParameters ()

  Get parameters.
- void setAndRestoreParameters (cParameters &parameters)

  Set and restore parameters.
- void closeEvent (QCloseEvent ∗event)

  Handle the window close event.
- void setSequence (int peptidetypeindex, QString sequence)

  Set peptide type and searched sequence.
- void setTag (int peptidetypeindex, QString tag)

  Set peptide type and sequence tag.
- void applyGlobalPreferences (cGlobalPreferences ∗globalpreferences)

  Apply new global preferences.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.

16.66.1 Detailed Description

The class representing a widget for editing parameters of the application.

16.66.2 Constructor & Destructor Documentation

16.66.2.1 cParametersWidget()

```
cParametersWidget::cParametersWidget (
            cGlobalPreferences * globalpreferences,
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.66.3 Member Function Documentation

16.66.3.1 applyGlobalPreferences()

```
void cParametersWidget::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.66.3.2 closeEvent()

```
void cParametersWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.66.3.3 getParameters()

```
cParameters & cParametersWidget::getParameters ( )
```

Get parameters.

Return values

cParameters	reference to parameters

16.66.3.4 keyPressEvent()

```
void cParametersWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.66.3.5 sendSequenceLine

```
void cParametersWidget::sendSequenceLine (
            int peptidetypeindex,
            QString sequence )  [signal]
```

Send index of peptide type and searched sequence.

Parameters

peptidetypeindex	an index of current peptide type
sequence	searched sequence

16.66.3.6 setAndRestoreParameters()

```
void cParametersWidget::setAndRestoreParameters (
            cParameters & parameters )
```

Set and restore parameters.

Parameters

parameters	reference to parameters

16.66.3.7 setSequence()

```
void cParametersWidget::setSequence (
            int peptidetypeindex,
            QString sequence )
```

Set peptide type and searched sequence.

Parameters

peptidetypeindex	an index of current peptide type
sequence	searched sequence

16.66.3.8 setTag()

```
void cParametersWidget::setTag (
```

```
int peptidetypeindex,
QString tag )
```

Set peptide type and sequence tag.

Parameters

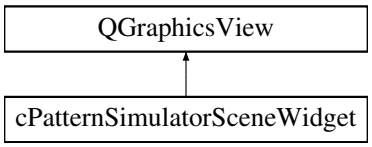
peptidetypeindex	an index of current peptide type
tag	sequence tag

The documentation for this class was generated from the following files:

- gui/cParametersWidget.h
- gui/cParametersWidget.cpp

16.67 cPatternSimulatorSceneWidget Class Reference

The implementation of an isotopic pattern simulator (graphics).
`#include <cPatternSimulatorSceneWidget.h>`
Inheritance diagram for cPatternSimulatorSceneWidget:



Signals

- void updateMZInterval (double minmz, double maxmz)

  The signal is emitted when the range of m/z ratios has been changed.

Public Member Functions

- cPatternSimulatorSceneWidget (QWidget ∗parent=0)

  The constructor.
- ∼**cPatternSimulatorSceneWidget** ()

  The destructor.
- void initialize (cPeaksList isotopicpattern, double fwhm)

  Initialize the widget.
- void exportToPDF (QString filename, bool postscript)

  Export spectrum scene into a PDF or a PS file.
- void exportToSVG (QString filename)

  Export spectrum scene into a SVG file.
- void exportToPNG (QString filename)

  Export spectrum scene into a PNG file.
- void hidePeakLabels (bool state)

  Hide/Show labels of peaks.

Protected Member Functions

- void wheelEvent (QWheelEvent ∗event)

  Handle the mouse wheel event.
- void mouseMoveEvent (QMouseEvent ∗event)

  Handle the mouse move event.

- void mouseReleaseEvent (QMouseEvent ∗event)

  Handle the mouse release event.
- void mousePressEvent (QMouseEvent ∗event)

  Handle the mouse press event.
- void showEvent (QShowEvent ∗event)

  Handle the widget show event.
- void resizeEvent (QResizeEvent ∗event)

  Handle the widget resize event.

16.67.1 Detailed Description

The implementation of an isotopic pattern simulator (graphics).

16.67.2 Constructor & Destructor Documentation

16.67.2.1 cPatternSimulatorSceneWidget()

```
cPatternSimulatorSceneWidget::cPatternSimulatorSceneWidget (
            QWidget * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.67.3 Member Function Documentation

16.67.3.1 exportToPDF()

```
void cPatternSimulatorSceneWidget::exportToPDF (
            QString filename,
            bool postscript )
```

Export spectrum scene into a PDF or a PS file.

Parameters

filename	filename
postscript	if true then PS file is generated instead of PDF

16.67.3.2 exportToPNG()

```
void cPatternSimulatorSceneWidget::exportToPNG (
            QString filename )
```

Export spectrum scene into a PNG file.

Parameters

filename	filename

16.67.3.3 exportToSVG()

```
void cPatternSimulatorSceneWidget::exportToSVG (
            QString filename )
```

Export spectrum scene into a SVG file.

Parameters

filename	filename

16.67.3.4 hidePeakLabels()

```
void cPatternSimulatorSceneWidget::hidePeakLabels (
            bool state )
```

Hide/Show labels of peaks.

Parameters

state	the peak labels are hidden when *state* is true

16.67.3.5 initialize()

```
void cPatternSimulatorSceneWidget::initialize (
            cPeaksList isotopicpattern,
            double fwhm )
```

Initialize the widget.

Parameters

isotopicpattern	an isotopic pattern
fwhm	fwhm

16.67.3.6 mouseMoveEvent()

```
void cPatternSimulatorSceneWidget::mouseMoveEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse move event.

Parameters

event	pointer to QMouseEvent

16.67.3.7 mousePressEvent()

```
void cPatternSimulatorSceneWidget::mousePressEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse press event.

Parameters

event	pointer to QMouseEvent

16.67.3.8 mouseReleaseEvent()

```
void cPatternSimulatorSceneWidget::mouseReleaseEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse release event.

Parameters

event	pointer to QMouseEvent

16.67.3.9 resizeEvent()

```
void cPatternSimulatorSceneWidget::resizeEvent (
            QResizeEvent * event )  [protected]
```

Handle the widget resize event.

Parameters

event	pointer to QResizeEvent

16.67.3.10 showEvent()

```
void cPatternSimulatorSceneWidget::showEvent (
            QShowEvent * event )  [protected]
```

Handle the widget show event.

Parameters

event	pointer to QShowEvent

16.67.3.11 updateMZInterval

```
void cPatternSimulatorSceneWidget::updateMZInterval (
            double minmz,
            double maxmz )  [signal]
```

The signal is emitted when the range of m/z ratios has been changed.

Parameters

minmz	a minimum threshold of m/z ratio
maxmz	a maximum threshold of m/z ratio

16.67.3.12 wheelEvent()

```
void cPatternSimulatorSceneWidget::wheelEvent (
            QWheelEvent * event )  [protected]
```

Handle the mouse wheel event.

Parameters

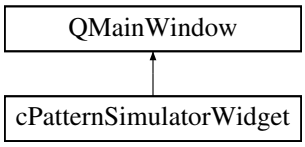
event	pointer to QWheelEvent

The documentation for this class was generated from the following files:

- gui/cPatternSimulatorSceneWidget.h
- gui/cPatternSimulatorSceneWidget.cpp

16.68 cPatternSimulatorWidget Class Reference

The implementation of an isotopic pattern simulator (window).
`#include <cPatternSimulatorWidget.h>`
Inheritance diagram for cPatternSimulatorWidget:



Signals

- void emitMZInterval (double minmz, double maxmz)

  The signal is emitted when the range of m/z ratios is changed.
- void rawDataStateChangedSignal (bool state)

  The signal is emitted when raw data state was changed.

Public Member Functions

- cPatternSimulatorWidget (cGlobalPreferences ∗globalpreferences, QWidget ∗parent)

  The constructor.
- ∼**cPatternSimulatorWidget** ()

  The destructor.
- void closeEvent (QCloseEvent ∗event)

  Handle the window close event.
- void applyGlobalPreferences (cGlobalPreferences ∗globalpreferences)

  Apply new global preferences.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.

16.68.1 Detailed Description

The implementation of an isotopic pattern simulator (window).

16.68.2 Constructor & Destructor Documentation

16.68.2.1 cPatternSimulatorWidget()

```
cPatternSimulatorWidget::cPatternSimulatorWidget (
            cGlobalPreferences * globalpreferences,
            QWidget * parent )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.68.3 Member Function Documentation

16.68.3.1 applyGlobalPreferences()

```
void cPatternSimulatorWidget::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.68.3.2 closeEvent()

```
void cPatternSimulatorWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.68.3.3 emitMZInterval

```
void cPatternSimulatorWidget::emitMZInterval (
            double minmz,
            double maxmz )  [signal]
```

The signal is emitted when the range of m/z ratios is changed.

Parameters

minmz	a minimum threshold of m/z ratio
maxmz	a maximum threshold of m/z ratio

16.68.3.4 keyPressEvent()

```
void cPatternSimulatorWidget::keyPressEvent (
```

```
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.68.3.5 rawDataStateChangedSignal

```
void cPatternSimulatorWidget::rawDataStateChangedSignal (
            bool state )  [signal]
```

The signal is emitted when raw data state was changed.

Parameters

state	current state

The documentation for this class was generated from the following files:

- gui/cPatternSimulatorWidget.h
- gui/cPatternSimulatorWidget.cpp

16.69 cPeak Struct Reference

A structure representing a peak in a mass spectrum.
`#include <cPeak.h>`

Public Member Functions

- **cPeak** ()
  The constructor.
- cPeak (const cPeak &peak)
  The copy constructor.
- cPeak & operator= (const cPeak &peak)
  Overloaded operator=.
- void **clear** ()
  Clear the structure.
- bool empty ()
  Test whether the structure is empty.
- void store (ofstream &os)
  Store the structure into an output stream.
- void load (ifstream &is)
  Load the structure from an input stream.
- bool equals (cPeak &secondpeak)
  Check if the object equals to another object.

Public Attributes

- cSummaryFormula **formula**
  Summary formlula.
- string **isotopeformula**
  Isotope summary formula.

- int **isotopeformulaid**

  Id of isotope summary formula.
- double **mzratio**

  MZ ratio.
- double **relativeintensity**

  Relative ntensity.
- double **absoluteintensity**

  Absolute intensity.
- string **description**

  Description of a peak.
- int **descriptionid**

  Description id of a peak.
- double **matchedmz**

  m/z ratio when the peak was matched with an experimental peak.
- double **matchedrelativeintensity**

  Relative intensity when the peak was matched with an experimental peak.
- double **matchedabsoluteintensity**

  Absolute intensity when the peak was matched with an experimental peak.
- double **matchedppm**

  An error in ppm when the peak was matched with an experimental peak.
- eFragmentIonType **iontype**

  Ion type corresponding to the peak.
- int **neutrallosstype**

  Neutral loss type corresponding to the peak.
- int **matched**

  Number of matches of an experimental peak (when the peak is compared to a theoretical spectrum).
- int **matchedid**

  An order of an experimental peak which has been matched with a theoretical peak.
- int **charge**

  A charge of the peak.
- int **rotationid**

  Id of a rotation of a cyclic peptide or a permutation of a branched peptide which has been matched.
- int **seriesid**

  An order of a peak in a series (e.g., B_5 in a B series).
- int **groupid**

  Identifier of a group of peaks.
- int **compoundid**

  Identifier of a compound.
- bool **isotope**

  True if the peak is an isotope peak.
- bool **removeme**

  True if the peak should be removed.
- bool **scrambled**

  True if the peak corresponds to a scrambled fragment ion.
- bool **decoy**

  True if the peak corresponds to a decoy compound.
- int **orderid**

  An order id of a peak in a peaklist.

16.69.1 Detailed Description

A structure representing a peak in a mass spectrum.

16.69.2 Constructor & Destructor Documentation

16.69.2.1 cPeak()

```
cPeak::cPeak (
            const cPeak & peak )
```

The copy constructor.

Parameters

peak	reference to cPeak

16.69.3 Member Function Documentation

16.69.3.1 empty()

```
bool cPeak::empty ( )
```

Test whether the structure is empty.

Return values

bool	true when the structure is empty

16.69.3.2 equals()

```
bool cPeak::equals (
            cPeak & secondpeak )
```

Check if the object equals to another object.

Parameters

secondpeak	object for comparison

Return values

bool	true if the objects are equals

16.69.3.3 load()

```
void cPeak::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.69.3.4 operator=()

```
cPeak & cPeak::operator= (
            const cPeak & peak )
```

Overloaded operator=.

Parameters

peak	reference to cPeak

16.69.3.5 store()

```
void cPeak::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this struct was generated from the following files:

- core/cPeak.h
- core/cPeak.cpp

16.70 cPeakListSeries Class Reference

The class representing a series of peaklists.
`#include <cPeakListSeries.h>`

Public Member Functions

- **cPeakListSeries** ()
  The constructor.
- cPeakListSeries (const cPeakListSeries &peaklistseries)
  The copy constructor.
- cPeakListSeries & operator= (const cPeakListSeries &peaklistseries)
  Overloaded operator=.
- cPeaksList & operator[ ] (int position)
  Overloaded operator [].
- void **clear** ()
  Clear the peaklist series.
- void resize (int size)
  Resize the peaklist series.
- void addPeakList (cPeaksList &peaklist)
  Add a peaklist.
- void loadFromPlainTextStream (ifstream &stream)
  Load the series of peaklists from a plain text stream.
- void loadFromBAFStream (ifstream &stream)
  Load the series of peaklists from a .csv file converted from a .baf file by CompassXport.
- int loadFromProfileApexStream (string &filename, ifstream &stream, ifstream &titlestream, double fwhm, cMainThread *os, bool &terminatecomputation)
  Load the series of peaklists from a .csv file converted from apex data by CompassXport.

- int loadFromMZMLStream (string &mzmlfilename, ifstream &mzmlstream, double fwhm, eModeType mode, cMainThread *os, bool &terminatecomputation)

  Load the series of peaklists from .mzML file.
- int loadFromIMZMLStream (string &imzmlfilename, ifstream &ibdstream, double fwhm, int &defaultmaxx, int &defaultmaxy, int &defaultpixelsizex, int &defaultpixelsizey, eVendorType &vendor, cMainThread *os, bool &terminatecomputation)

  Load the series of peaklists from .imzML file.
- void loadSpotList (ifstream &stream)

  Load a spot list from a stream.
- void loadFromMGFStream (ifstream &stream, int timeunit)

  Load the series of peaklists from a .mgf file.
- int size ()

  The number of peaklists in the series.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is, int fileversionpart1, int fileversionpart2, int fileversionpart3)

  Load the structure from an input stream.

16.70.1 Detailed Description

The class representing a series of peaklists.

16.70.2 Constructor & Destructor Documentation

16.70.2.1 cPeakListSeries()

```
cPeakListSeries::cPeakListSeries (
            const cPeakListSeries & peaklistseries )
```

The copy constructor.

Parameters

peaklistseries	reference to a cPeakListSeries

16.70.3 Member Function Documentation

16.70.3.1 addPeakList()

```
void cPeakListSeries::addPeakList (
            cPeaksList & peaklist )
```

Add a peaklist.

Parameters

peaklist	peaklist which will be added

16.70.3.2 load()

```
void cPeakListSeries::load (
```

```
            ifstream & is,
            int fileversionpart1,
            int fileversionpart2,
            int fileversionpart3 )
```

Load the structure from an input stream.

Parameters

is	an input stream
fileversionpart1	first number of .res the file version
fileversionpart2	second number of .res the file version
fileversionpart3	third number of .res the file version

16.70.3.3 loadFromBAFStream()

```
void cPeakListSeries::loadFromBAFStream (
            ifstream & stream )
```

Load the series of peaklists from a .csv file converted from a .baf file by CompassXport.

Parameters

stream	source stream

16.70.3.4 loadFromIMZMLStream()

```
int cPeakListSeries::loadFromIMZMLStream (
            string & imzmlfilename,
            ifstream & ibdstream,
            double fwhm,
            int & defaultmaxx,
            int & defaultmaxy,
            int & defaultpixelsizex,
            int & defaultpixelsizey,
            eVendorType & vendor,
            cMainThread * os,
            bool & terminatecomputation )
```

Load the series of peaklists from .imzML file.

Parameters

imzmlfilename	imzML filename
ibdstream	ibd binary file stream
fwhm	FWHM
defaultmaxx	max count of pixel x - parsed from imzML file
defaultmaxy	max count of pixel y - parsed from imzML file
defaultpixelsizex	pixel size x - parsed from imzML file
defaultpixelsizey	pixel size y - parsed from imzML file
vendor	vendor type
os	pointer to the main thread of the application (output stream)
terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 = success; -1 = aborted by user, -2 = error

16.70.3.5 loadFromMGFStream()

```
void cPeakListSeries::loadFromMGFStream (
            ifstream & stream,
            int timeunit )
```

Load the series of peaklists from a .mgf file.

Parameters

stream	source stream
timeunit	rt unit

16.70.3.6 loadFromMZMLStream()

```
int cPeakListSeries::loadFromMZMLStream (
            string & mzmlfilename,
            ifstream & mzmlstream,
            double fwhm,
            eModeType mode,
            cMainThread * os,
            bool & terminatecomputation )
```

Load the series of peaklists from .mzML file.

Parameters

mzmlfilename	mzML filename
mzmlstream	mzML file stream
fwhm	FWHM
mode	program mode
os	pointer to the main thread of the application (output stream)
terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 = success; -1 = aborted by user, -2 = error

16.70.3.7 loadFromPlainTextStream()

```
void cPeakListSeries::loadFromPlainTextStream (
            ifstream & stream )
```

Load the series of peaklists from a plain text stream.

Parameters

stream	source stream

16.70.3.8 loadFromProfileApexStream()

```
int cPeakListSeries::loadFromProfileApexStream (
            string & filename,
            ifstream & stream,
            ifstream & titlestream,
            double fwhm,
            cMainThread * os,
            bool & terminatecomputation )
```

Load the series of peaklists from a .csv file converted from apex data by CompassXport.

Parameters

filename	filename
stream	source stream
titlestream	title stream
fwhm	FWHM
os	pointer to the main thread of the application (output stream)
terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 = success; -1 = aborted by user, -2 = error

16.70.3.9 loadSpotList()

```
void cPeakListSeries::loadSpotList (
            ifstream & stream )
```

Load a spot list from a stream.

Parameters

stream	source stream

16.70.3.10 operator=()

```
cPeakListSeries & cPeakListSeries::operator= (
            const cPeakListSeries & peaklistseries )
```

Overloaded operator=.

Parameters

peaklistseries	reference to a cPeakListSeries

16.70.3.11 operator[]()

```
cPeaksList & cPeakListSeries::operator[] (
            int position )
```

Overloaded operator [].

Parameters

position	position of a peaklist

Return values

cPeaksList	reference to a peaklist

16.70.3.12 resize()

```
void cPeakListSeries::resize (
            int size )
```

Resize the peaklist series.

Parameters

size	new size

16.70.3.13 size()

```
int cPeakListSeries::size ( )
```

The number of peaklists in the series.

Return values

int	number of peaklists in the series

16.70.3.14 store()

```
void cPeakListSeries::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this class was generated from the following files:

- core/cPeakListSeries.h
- core/cPeakListSeries.cpp

16.71 cPeaksList Class Reference

The class representing a peak list.

```
#include <cPeaksList.h>
```

Public Member Functions

- **cPeaksList** ()

  The constructor.
- cPeaksList (const cPeaksList &peakslist)

  The copy constructor.
- cPeaksList & operator= (const cPeaksList &peakslist)

  Overloaded operator=.
- void **clear** ()

  Clear the peak list.
- void attach (cPeaksList &peaklist)

  Attach a peak list.
- void add (cPeak &peak)

  Add a peak.
- void loadFromPlainTextStream (ifstream &stream)

  Load the spectrum from a plain text stream (tab separated values m/z ratio and intensity).
- void loadFromBAFStream (ifstream &stream)

  Load the spectrum from a .csv file converted from a .baf file by CompassXport.
- void loadFromIBDStream (cImzMLItem &imzmlitem, ifstream &ibdstream, bool use_64bit_float_mz_↩ precision, bool use_64bit_float_intensity_precision)

  Load the spectrum from .ibd file.
- void storeToIBDStream (ofstream &ibdstream, bool use_64bit_float_mz_precision, bool use_64bit_float_↩ intensity_precision)

  Store the spectrum into .ibd file.
- void loadFromMGFStream (ifstream &stream, int timeunit)

  Load the spectrum from a .mgf file.
- int size ()

  The number of peaks.
- string print (bool htmlterminatelines=false, bool printgroupid=false)

  Print the spectrum.
- void sortbyMass (int limit=-1)

  Sort the peaks by mass.
- void **sortbyRelativeIntensityDesc** ()

  Sort the peaks by relative intensity in the descending order.
- void **sortbyAbsoluteIntensityDesc** ()

  Sort the peaks by absolute intensity in the descending order.
- void **sortbyGroupId** ()

  Sort the peaks by group id.
- void **sortbyOrderId** ()

  Sort the peaks by order id.
- int normalizeIntenzity ()

  Normalize intensities of peaks.
- void normalizeIntenzityByValue (double value)

  Normalize intensities of peaks using a value.
- void cropRelativeIntenzity (double minimumrelativeintensitythreshold)

  Remove peaks with relative intensities lower than the threshold.
- void cropAbsoluteIntenzity (unsigned minimumabsoluteintensitythreshold)

  Remove peaks with absolute intensities lower than the threshold.
- void cropMinimumMZRatio (double minimummz, double errortolerance)

  Crop mz ratio lower than a threshold.
- void cropMaximumMZRatio (double maximummz, double errortolerance)

Crop mz ratio bigger than a threshold.

- cPeak & operator[ ] (int position)

  Overloaded operator [].
- void resize (int size)

  Resize the peak list.
- double getMinimumMZRatio ()

  Get the minimum mz ratio.
- double getMaximumMZRatio ()

  Get the maximum mz ratio.
- void removeIsotopes (int maximumcharge, double fragmentmasserrortolerance, cMainThread *os=0)

  Remove isotopic peaks.
- int find (double mzratio, double fragmentmasserrortolerance)

  Get a position of a peak with specified mzratio.
- void remove (int position)

  Remove a peak.
- void removeChargeVariants (int maximumcharge, double fragmentmasserrortolerance)

  Perform a deconvolution.
- void removeNeutralLoss (double loss, int maximumcharge, double fragmentmasserrortolerance)

  Remove neutral losses.
- void **removeObsolete** ()

  Remove peaks having set up the flag removeme == true.
- void maxHighestPeaksInWindow (int maximumnumberofpeaksinwindow, double windowsize)

  Keep only n highest peaks in a window of a specified size.
- double getMaximumRelativeIntensityFromMZInterval (double minmz, double maxmz, bool hidematched, bool hideunmatched, ePeptideType peptidetype, bool hidescrambled)

  Get the maximum relative intensity from an m/z interval.
- double getMaximumAbsoluteIntensityFromMZInterval (double minmz, double maxmz, bool hidematched, bool hideunmatched, ePeptideType peptidetype, bool hidescrambled)

  Get the maximum absolute intensity from an m/z interval.
- double getMaximumRelativeIntensity ()

  Get the maximum relative intensity.
- double getMaximumAbsoluteIntensity ()

  Get the maximum absolute intensity.
- void setRetentionTime (double rt)

  Set the retention time.
- double getRetentionTime ()

  Get the retention time.
- void setRetentionTimeUnit (int unit)

  Set the retention time unit.
- int getRetentionTimeUnit ()

  Get the retention time unit.
- void setCoordinates (int x, int y)

  Set the coordinates.
- int getCoordinateX ()

  Get the X coordinate.
- int getCoordinateY ()

  Get the Y coordinate.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is, int fileversionpart1, int fileversionpart2, int fileversionpart3)

  Load the structure from an input stream.

- string & getTitle ()

  Get the title.
- void setTitle (string &title)

  Set the title.
- void reducePeakDescriptions (unordered_map< string, int > &peakdesctoid)

  Reduce redundant descriptions of peaks.
- void reduceIsotopeFormulaDescriptions (unordered_map< string, int > &isotopeformuladesctoid)

  Reduce redundant descriptions of isotope formulas.
- void **fillOrderIDs** ()

  Define order ids of peaks.
- void setIsotopeFlags (bool value)

  Set isotope flags.
- void **markIsotopes** ()

  Mark isotopic peaks in isotopic patterns.
- bool equals (cPeaksList &secondpeaklist)

  Check if the object equals to another object.

16.71.1 Detailed Description

The class representing a peak list.

16.71.2 Constructor & Destructor Documentation

16.71.2.1 cPeaksList()

```
cPeaksList::cPeaksList (
            const cPeaksList & peakslist )
```

The copy constructor.

Parameters

peakslist	reference to a cPeaksList

16.71.3 Member Function Documentation

16.71.3.1 add()

```
void cPeaksList::add (
            cPeak & peak )
```

Add a peak.

Parameters

peak	peak which will be added

16.71.3.2 attach()

```
void cPeaksList::attach (
            cPeaksList & peaklist )
```

Attach a peak list.

Parameters

peaklist	peaklist which will be attached

16.71.3.3 cropAbsoluteIntenzity()

```
void cPeaksList::cropAbsoluteIntenzity (
            unsigned minimumabsoluteintensitythreshold )
```

Remove peaks with absolute intensities lower than the threshold.

Parameters

minimumabsoluteintensitythreshold	minimum threshold of absolute intensity

16.71.3.4 cropMaximumMZRatio()

```
void cPeaksList::cropMaximumMZRatio (
            double maximummz,
            double errortolerance )
```

Crop mz ratio bigger than a threshold.

Parameters

maximummz	maximum threshold of mz ratio
errortolerance	mz error tolerance in ppm

16.71.3.5 cropMinimumMZRatio()

```
void cPeaksList::cropMinimumMZRatio (
            double minimummz,
            double errortolerance )
```

Crop mz ratio lower than a threshold.

Parameters

minimummz	minimum threshold of mz ratio
errortolerance	mz error tolerance in ppm

16.71.3.6 cropRelativeIntenzity()

```
void cPeaksList::cropRelativeIntenzity (
            double minimumrelativeintensitythreshold )
```

Remove peaks with relative intensities lower than the threshold.

Parameters

minimumrelativeintensitythreshold	minimum threshold of relative intensity

16.71.3.7 equals()

```
bool cPeaksList::equals (
            cPeaksList & secondpeaklist )
```

Check if the object equals to another object.

Parameters

secondpeaklist	object for comparison

Return values

bool	true if the objects are equals

16.71.3.8 find()

```
int cPeaksList::find (
            double mzratio,
            double fragmentmasserrortolerance )
```

Get a position of a peak with specified mzratio.

Parameters

mzratio	of searched peak
fragmentmasserrortolerance	mz error tolerance

Return values

int	position; -1 when the peak is not found

16.71.3.9 getCoordinateX()

```
int cPeaksList::getCoordinateX ( )
```

Get the X coordinate.

Return values

int	X coordinate

16.71.3.10 getCoordinateY()

```
int cPeaksList::getCoordinateY ( )
```

Get the Y coordinate.

Return values

int	Y coordinate

16.71.3.11 getMaximumAbsoluteIntensity()

```
double cPeaksList::getMaximumAbsoluteIntensity ( )
```

Get the maximum absolute intensity.

Return values

double	maximum absolute intensity

16.71.3.12 getMaximumAbsoluteIntensityFromMZInterval()

```
double cPeaksList::getMaximumAbsoluteIntensityFromMZInterval (
            double minmz,
            double maxmz,
            bool hidematched,
            bool hideunmatched,
            ePeptideType peptidetype,
            bool hidescrambled )
```

Get the maximum absolute intensity from an m/z interval.

Parameters

minmz	minimum mz ratio
maxmz	maximum mz ratio
hidematched	true, if matched peaks are hidden
hideunmatched	true, if unmatched peaks are hidden
peptidetype	the type of peptide
hidescrambled	true, if scrambled peaks are hidden

Return values

double	maximum absolute intensity

16.71.3.13 getMaximumMZRatio()

```
double cPeaksList::getMaximumMZRatio ( )
```

Get the maximum mz ratio.

Return values

double	maximum mz ratio

16.71.3.14 getMaximumRelativeIntensity()

```
double cPeaksList::getMaximumRelativeIntensity ( )
```

Get the maximum relative intensity.

Return values

double	maximum relative intensity

16.71.3.15 getMaximumRelativeIntensityFromMZInterval()

```
double cPeaksList::getMaximumRelativeIntensityFromMZInterval (
            double minmz,
            double maxmz,
            bool hidematched,
            bool hideunmatched,
            ePeptideType peptidetype,
            bool hidescrambled )
```

Get the maximum relative intensity from an m/z interval.

Parameters

minmz	minimum mz ratio
maxmz	maximum mz ratio
hidematched	true, if matched peaks are hidden
hideunmatched	true, if unmatched peaks are hidden
peptidetype	the type of peptide
hidescrambled	true, if scrambled peaks are hidden

Return values

double	maximum relative intensity

16.71.3.16 getMinimumMZRatio()

```
double cPeaksList::getMinimumMZRatio ( )
```

Get the minimum mz ratio.

Return values

double	minimum mz ratio

16.71.3.17 getRetentionTime()

```
double cPeaksList::getRetentionTime ( )
```

Get the retention time.

Return values

double	retention time

16.71.3.18 getRetentionTimeUnit()

```
int cPeaksList::getRetentionTimeUnit ( )
```

Get the retention time unit.

Return values

int	0 == second, 1 == minute

16.71.3.19 getTitle()

```
string & cPeaksList::getTitle ( )
```

Get the title.

Return values

string	title

16.71.3.20 load()

```
void cPeaksList::load (
            ifstream & is,
            int fileversionpart1,
            int fileversionpart2,
            int fileversionpart3 )
```

Load the structure from an input stream.

Parameters

is	an input stream
fileversionpart1	first number of .res the file version
fileversionpart2	second number of .res the file version
fileversionpart3	third number of .res the file version

16.71.3.21 loadFromBAFStream()

```
void cPeaksList::loadFromBAFStream (
            ifstream & stream )
```

Load the spectrum from a .csv file converted from a .baf file by CompassXport.

Parameters

stream	source stream

16.71.3.22 loadFromIBDStream()

```
void cPeaksList::loadFromIBDStream (
            cImzMLItem & imzmlitem,
            ifstream & ibdstream,
            bool use_64bit_float_mz_precision,
```

```
            bool use_64bit_float_intensity_precision )
```

Load the spectrum from .ibd file.

Parameters

imzmlitem	clmzML containing the offset in the ibd file
ibdstream	an input binary file stream
use_64bit_float_mz_precision	if true, 64bit mz precision is used; if false, 32bit precision is used
use_64bit_float_intensity_precision	if true, 64bit intensity precision is used; if false, 32bit precision is used

16.71.3.23 loadFromMGFStream()

```
void cPeaksList::loadFromMGFStream (
            ifstream & stream,
            int timeunit )
```

Load the spectrum from a .mgf file.

Parameters

stream	source stream
timeunit	rt unit

16.71.3.24 loadFromPlainTextStream()

```
void cPeaksList::loadFromPlainTextStream (
            ifstream & stream )
```

Load the spectrum from a plain text stream (tab separated values m/z ratio and intensity).

Parameters

stream	source stream

16.71.3.25 maxHighestPeaksInWindow()

```
void cPeaksList::maxHighestPeaksInWindow (
            int maximumnumberofpeaksinwindow,
            double windowsize )
```

Keep only n highest peaks in a window of a specified size.

Parameters

maximumnumberofpeaksinwindow	maximum number of peaks in a window
windowsize	size of a window

16.71.3.26 normalizeIntenzity()

```
int cPeaksList::normalizeIntenzity ( )
```

Normalize intensities of peaks.

Return values

int	0 == success, -1 == maximum intensity <= 0

16.71.3.27 normalizeIntenzityByValue()

```
void cPeaksList::normalizeIntenzityByValue (
            double value )
```

Normalize intensities of peaks using a value.

Parameters

value	normalization value

16.71.3.28 operator=()

```
cPeaksList & cPeaksList::operator= (
            const cPeaksList & peakslist )
```

Overloaded operator=.

Parameters

peakslist	reference to a cPeaksList

16.71.3.29 operator[]()

```
cPeak & cPeaksList::operator[] (
            int position )
```

Overloaded operator [].

Parameters

position	position of a peak in the peak list

Return values

cPeak	reference to the peak

16.71.3.30 print()

```
string cPeaksList::print (
            bool htmlterminatelines = false,
            bool printgroupid = false )
```

Print the spectrum.

Parameters

htmlterminatelines	true if the html line endings should be used
printgroupid	true if the groupid should be printed

Return values

string	string with a printed spectrum

16.71.3.31 reduceIsotopeFormulaDescriptions()

```
void cPeaksList::reduceIsotopeFormulaDescriptions (
            unordered_map< string, int > & isotopeformuladesctoid )
```

Reduce redundant descriptions of isotope formulas.

Parameters

isotopeformuladesctoid	map of isotope formula descriptions to ids

16.71.3.32 reducePeakDescriptions()

```
void cPeaksList::reducePeakDescriptions (
            unordered_map< string, int > & peakdesctoid )
```

Reduce redundant descriptions of peaks.

Parameters

peakdesctoid	map of peak descriptions to ids

16.71.3.33 remove()

```
void cPeaksList::remove (
            int position )
```

Remove a peak.

Parameters

position	position of a peak

16.71.3.34 removeChargeVariants()

```
void cPeaksList::removeChargeVariants (
            int maximumcharge,
            double fragmentmasserrortolerance )
```

Perform a deconvolution.

Parameters

maximumcharge	charge of the precursor
fragmentmasserrortolerance	mz error tolerance

16.71.3.35 removeIsotopes()

```
void cPeaksList::removeIsotopes (
            int maximumcharge,
            double fragmentmasserrortolerance,
            cMainThread * os = 0 )
```

Remove isotopic peaks.

Parameters

maximumcharge	charge of the precursor
fragmentmasserrortolerance	mz error tolerance
os	reference to a main thread of the application (stream, respectively)

16.71.3.36 removeNeutralLoss()

```
void cPeaksList::removeNeutralLoss (
            double loss,
            int maximumcharge,
            double fragmentmasserrortolerance )
```

Remove neutral losses.

Parameters

loss	mass change
maximumcharge	charge of the precursor
fragmentmasserrortolerance	mz error tolerance

16.71.3.37 resize()

```
void cPeaksList::resize (
            int size )
```

Resize the peak list.

Parameters

size	number of peaks

16.71.3.38 setCoordinates()

```
void cPeaksList::setCoordinates (
            int x,
            int y )
```

Set the coordinates.

Parameters

x	X coordinate
y	Y coordinate

16.71.3.39 setIsotopeFlags()

```
void cPeaksList::setIsotopeFlags (
            bool value )
```

Set isotope flags.

Parameters

value	flag value

16.71.3.40 setRetentionTime()

```
void cPeaksList::setRetentionTime (
            double rt )
```

Set the retention time.

Parameters

rt	retention time

16.71.3.41 setRetentionTimeUnit()

```
void cPeaksList::setRetentionTimeUnit (
            int unit )
```

Set the retention time unit.

Parameters

unit	0 == second, 1 == minute

16.71.3.42 setTitle()

```
void cPeaksList::setTitle (
            string & title )
```

Set the title.

Parameters

title	title

16.71.3.43 size()

```
int cPeaksList::size ( )
```

The number of peaks.

Return values

int	number of peaks

16.71.3.44 sortbyMass()

```
void cPeaksList::sortbyMass (
            int limit = -1 )
```

Sort the peaks by mass.

Parameters

limit	sort first *limit* peaks only; if *limit* == -1 then all peaks are sorted

16.71.3.45 store()

```
void cPeaksList::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

16.71.3.46 storeToIBDStream()

```
void cPeaksList::storeToIBDStream (
            ofstream & ibdstream,
            bool use_64bit_float_mz_precision,
            bool use_64bit_float_intensity_precision )
```

Store the spectrum into .ibd file.

Parameters

ibdstream	an output binary file stream
use_64bit_float_mz_precision	if true, 64bit mz precision is used; if false, 32bit precision is used
use_64bit_float_intensity_precision	if true, 64bit intensity precision is used; if false, 32bit precision is used

The documentation for this class was generated from the following files:

- core/cPeaksList.h
- core/cPeaksList.cpp

16.72 cPeriodicTableMap Class Reference

The class representing a periodic table.
`#include <cFragmentIons.h>`

Public Member Functions

- **cPeriodicTableMap** ()

  The constructor.
- double & operator[ ] (string &element)

  Overloaded operator [].
- int count (string &element)

  Check whether an element occurs in the table.

- map< string, double > & getElements ()

  Reference to the map of elements.
- vector< string > & getAdductElements ()

  Vector of elements which are allowed to be adducts.

16.72.1 Detailed Description

The class representing a periodic table.

16.72.2 Member Function Documentation

16.72.2.1 count()

```
int cPeriodicTableMap::count (
            string & element )
```

Check whether an element occurs in the table.

Parameters

element	reference to a string with a name of an element

Return values

int	1 when element is presented in the table, 0 otherwise

16.72.2.2 getAdductElements()

```
vector< string > & cPeriodicTableMap::getAdductElements ( )
```

Vector of elements which are allowed to be adducts.

Return values

vector<string>	adducts

16.72.2.3 getElements()

```
map< string, double > & cPeriodicTableMap::getElements ( )
```

Reference to the map of elements.

Return values

map<string,double>	map of elements

16.72.2.4 operator[]()

```
double & cPeriodicTableMap::operator[] (
            string & element )
```

Overloaded operator [].

Parameters

element	reference to a string with a name of an element

Return values

double	mass of an element

The documentation for this class was generated from the following files:

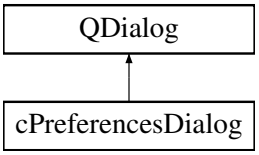
- core/cFragmentIons.h
- core/cFragmentIons.cpp

16.73 cPreferencesDialog Class Reference

Preferences dialog.

`#include <cPreferencesDialog.h>`

Inheritance diagram for cPreferencesDialog:



Public Member Functions

- cPreferencesDialog (cGlobalPreferences &globalpreferences, QWidget *parent=0)
  The constructor.
- void setPreferences (cGlobalPreferences &globalpreferences)
  Set preferences.
- void resetPreferences (cGlobalPreferences &globalpreferences)
  Reset preferences.
- ~**cPreferencesDialog** ()
  The destructor.

Protected Member Functions

- void keyPressEvent (QKeyEvent *event)
  Handle a key press event.

16.73.1 Detailed Description

Preferences dialog.

16.73.2 Constructor & Destructor Documentation

16.73.2.1 cPreferencesDialog()

```
cPreferencesDialog::cPreferencesDialog (
            cGlobalPreferences & globalpreferences,
            QWidget * parent = 0 )  [explicit]
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.73.3 Member Function Documentation

16.73.3.1 keyPressEvent()

```
void cPreferencesDialog::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.73.3.2 resetPreferences()

```
void cPreferencesDialog::resetPreferences (
            cGlobalPreferences & globalpreferences )
```

Reset preferences.

Parameters

globalpreferences	global preferences of the application

16.73.3.3 setPreferences()

```
void cPreferencesDialog::setPreferences (
            cGlobalPreferences & globalpreferences )
```

Set preferences.

Parameters

globalpreferences	global preferences of the application

The documentation for this class was generated from the following files:

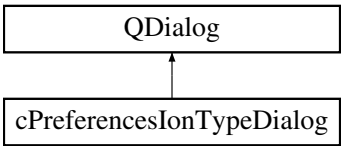
- gui/cPreferencesDialog.h
- gui/cPreferencesDialog.cpp

16.74 cPreferencesIonTypeDialog Class Reference

Preferences dialog.

`#include <cPreferencesIonTypeDialog.h>`

Inheritance diagram for cPreferencesIonTypeDialog:



Public Member Functions

- cPreferencesIonTypeDialog (QWidget *parent=0)

  The constructor.
- void getValues (clonType &ion)

  Get values.
- void setValues (clonType &ion)

  Set values.
- **~cPreferencesIonTypeDialog** ()

  The destructor.

16.74.1 Detailed Description

Preferences dialog.

16.74.2 Constructor & Destructor Documentation

16.74.2.1 cPreferencesIonTypeDialog()

```
cPreferencesIonTypeDialog::cPreferencesIonTypeDialog (
            QWidget * parent = 0 )  [explicit]
```

The constructor.

Parameters

parent	pointer to a parent widget

16.74.3 Member Function Documentation

16.74.3.1 getValues()

```
void cPreferencesIonTypeDialog::getValues (
            cIonType & ion )
```

Get values.

Parameters

ion	ion

16.74.3.2 setValues()

```
void cPreferencesIonTypeDialog::setValues (
            cIonType & ion )
```

Set values.

Parameters

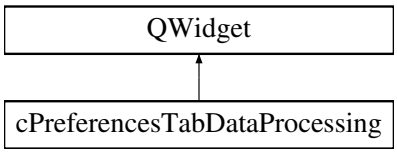
ion	ion

The documentation for this class was generated from the following files:

- gui/cPreferencesIonTypeDialog.h
- gui/cPreferencesIonTypeDialog.cpp

16.75 cPreferencesTabDataProcessing Class Reference

The class representing a tab used to define the data processing parameters.
`#include <cPreferencesTabDataProcessing.h>`
Inheritance diagram for cPreferencesTabDataProcessing:



Public Member Functions

- cPreferencesTabDataProcessing (cGlobalPreferences &globalpreferences, QWidget ∗parent=0)

  The constructor.
- void setPreferences (cGlobalPreferences &globalpreferences)

  Set preferences.
- void resetPreferences (cGlobalPreferences &globalpreferences)

  Reset preferences.
- ∼**cPreferencesTabDataProcessing** ()

  The destructor.

16.75.1 Detailed Description

The class representing a tab used to define the data processing parameters.

16.75.2 Constructor & Destructor Documentation

16.75.2.1 cPreferencesTabDataProcessing()

```
cPreferencesTabDataProcessing::cPreferencesTabDataProcessing (
            cGlobalPreferences & globalpreferences,
            QWidget * parent = 0 )  [explicit]
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.75.3 Member Function Documentation

16.75.3.1 resetPreferences()

```
void cPreferencesTabDataProcessing::resetPreferences (
            cGlobalPreferences & globalpreferences )
```

Reset preferences.

Parameters

globalpreferences	global preferences of the application

16.75.3.2 setPreferences()

```
void cPreferencesTabDataProcessing::setPreferences (
            cGlobalPreferences & globalpreferences )
```

Set preferences.

Parameters

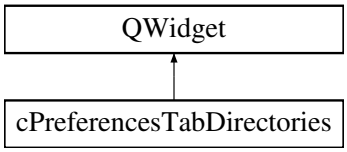
globalpreferences	global preferences of the application

The documentation for this class was generated from the following files:

- gui/cPreferencesTabDataProcessing.h
- gui/cPreferencesTabDataProcessing.cpp

16.76 cPreferencesTabDirectories Class Reference

The class representing a tab of default directories.
`#include <cPreferencesTabDirectories.h>`
Inheritance diagram for cPreferencesTabDirectories:



Public Member Functions

- cPreferencesTabDirectories (cGlobalPreferences &globalpreferences, QWidget *parent=0)
  The constructor.
- void setPreferences (cGlobalPreferences &globalpreferences)
  Set preferences.
- void resetPreferences (cGlobalPreferences &globalpreferences)
  Reset preferences.
- **~cPreferencesTabDirectories** ()
  The destructor.

16.76.1 Detailed Description

The class representing a tab of default directories.

16.76.2 Constructor & Destructor Documentation

16.76.2.1 cPreferencesTabDirectories()

```
cPreferencesTabDirectories::cPreferencesTabDirectories (
            cGlobalPreferences & globalpreferences,
            QWidget * parent = 0 )  [explicit]
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.76.3 Member Function Documentation

16.76.3.1 resetPreferences()

```
void cPreferencesTabDirectories::resetPreferences (
            cGlobalPreferences & globalpreferences )
```

Reset preferences.

Parameters

globalpreferences	global preferences of the application

16.76.3.2 setPreferences()

```
void cPreferencesTabDirectories::setPreferences (
            cGlobalPreferences & globalpreferences )
```

Set preferences.

Parameters

globalpreferences	global preferences of the application

The documentation for this class was generated from the following files:

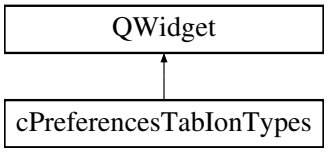
- gui/cPreferencesTabDirectories.h
- gui/cPreferencesTabDirectories.cpp

16.77 cPreferencesTabIonTypes Class Reference

The class representing a tab of ion types.

`#include <cPreferencesTabIonTypes.h>`

Inheritance diagram for cPreferencesTabIonTypes:



Public Member Functions

- cPreferencesTabIonTypes (cGlobalPreferences &globalpreferences, QWidget *parent=0)

  The constructor.
- void setPreferences (cGlobalPreferences &globalpreferences)

  Set preferences.
- void resetPreferences (cGlobalPreferences &globalpreferences)

  Reset preferences.
- **~cPreferencesTabIonTypes** ()

  The destructor.

16.77.1 Detailed Description

The class representing a tab of ion types.

16.77.2 Constructor & Destructor Documentation

16.77.2.1 cPreferencesTabIonTypes()

```
cPreferencesTabIonTypes::cPreferencesTabIonTypes (
            cGlobalPreferences & globalpreferences,
            QWidget * parent = 0 )  [explicit]
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.77.3 Member Function Documentation

16.77.3.1 resetPreferences()

```
void cPreferencesTabIonTypes::resetPreferences (
            cGlobalPreferences & globalpreferences )
```

Reset preferences.

Parameters

globalpreferences	global preferences of the application

16.77.3.2 setPreferences()

```
void cPreferencesTabIonTypes::setPreferences (
            cGlobalPreferences & globalpreferences )
```

Set preferences.

Parameters

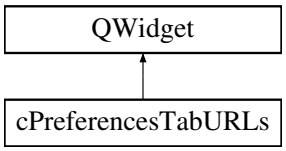
globalpreferences	global preferences of the application

The documentation for this class was generated from the following files:

- gui/cPreferencesTabIonTypes.h
- gui/cPreferencesTabIonTypes.cpp

16.78 cPreferencesTabURLs Class Reference

The class representing a tab of URLs.
`#include <cPreferencesTabURLs.h>`
Inheritance diagram for cPreferencesTabURLs:



Public Member Functions

- cPreferencesTabURLs (cGlobalPreferences &globalpreferences, QWidget *parent=0)
  The constructor.
- void setPreferences (cGlobalPreferences &globalpreferences)
  Set preferences.
- void resetPreferences (cGlobalPreferences &globalpreferences)
  Reset preferences.
- **~cPreferencesTabURLs** ()
  The destructor.

16.78.1 Detailed Description

The class representing a tab of URLs.

16.78.2 Constructor & Destructor Documentation

16.78.2.1 cPreferencesTabURLs()

```
cPreferencesTabURLs::cPreferencesTabURLs (
            cGlobalPreferences & globalpreferences,
            QWidget * parent = 0 )  [explicit]
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.78.3 Member Function Documentation

16.78.3.1 resetPreferences()

```
void cPreferencesTabURLs::resetPreferences (
            cGlobalPreferences & globalpreferences )
```

Reset preferences.

Parameters

globalpreferences	global preferences of the application

16.78.3.2 setPreferences()

```
void cPreferencesTabURLs::setPreferences (
            cGlobalPreferences & globalpreferences )
```

Set preferences.

Parameters

globalpreferences	global preferences of the application

The documentation for this class was generated from the following files:

- gui/cPreferencesTabURLs.h
- gui/cPreferencesTabURLs.cpp

16.79 cPubchemCIDReader Struct Reference

The functor used to get the CIDs of compounds with the same formula.
`#include <cPubChemSearchWidget.h>`

Public Types

- typedef string **result_type**

  result_type

Public Member Functions

- string getCIDList (const string &compound)

  Get the list of CIDs.
- void getCommaSeparatedCIDs (string &cids, string &reducedcids)

  Get a list of CIDs separated by comma.
- void attachVectorOfCIDs (string &cids, vector< string > &cidsvector)

  Attach a vector of CIDs.
- int getNumberOfCIDs (string &cids)

  Get the number of compounds with the same formula.
- string operator() (const string &compound)

  Overloaded operator ().

16.79.1 Detailed Description

The functor used to get the CIDs of compounds with the same formula.

16.79.2 Member Function Documentation

16.79.2.1 attachVectorOfCIDs()

```
void cPubchemCIDReader::attachVectorOfCIDs (
            string & cids,
            vector< string > & cidsvector )  [inline]
```

Attach a vector of CIDs.

Parameters

cids	input list of CIDs
cidsvector	input/output vector of CIDs

16.79.2.2 getCIDList()

```
string cPubchemCIDReader::getCIDList (
            const string & compound )  [inline]
```

Get the list of CIDs.

Parameters

compound	formula

Return values

string	list of CIDs

16.79.2.3 getCommaSeparatedCIDs()

```
void cPubchemCIDReader::getCommaSeparatedCIDs (
            string & cids,
            string & reducedcids )  [inline]
```

Get a list of CIDs separated by comma.

Parameters

cids	input list of CIDs
reducedcids	output list of CIDs separated by comma

16.79.2.4 getNumberOfCIDs()

```
int cPubchemCIDReader::getNumberOfCIDs (
            string & cids )  [inline]
```

Get the number of compounds with the same formula.

Parameters

cids	list of CIDs

Return values

int	number of CIDs

16.79.2.5 operator()()

```
string cPubchemCIDReader::operator() (
            const string & compound )  [inline]
```

Overloaded operator ().

Parameters

compound	formula

Return values

string	list of CIDs

The documentation for this struct was generated from the following file:

- gui/cPubChemSearchWidget.h

16.80 cPubchemNameReader Struct Reference

The functor used to get the names of compounds from CID.
`#include <cPubChemSearchWidget.h>`

Public Types

- typedef string **result_type**
  result_type

Public Member Functions

- string getNameList (const string &cid)
  Get the list of names.
- void getReducedNames (string &names, set< string > &reducednames)
  Get the reduded list of names.
- string operator() (const string &cid)
  Overloaded operator ().

16.80.1 Detailed Description

The functor used to get the names of compounds from CID.

16.80.2 Member Function Documentation

16.80.2.1 getNameList()

```
string cPubchemNameReader::getNameList (
            const string & cid )  [inline]
```

Get the list of names.

Parameters

cid	CID

Return values

string	list of names

16.80.2.2 getReducedNames()

```
void cPubchemNameReader::getReducedNames (
            string & names,
            set< string > & reducednames )  [inline]
```

Get the reduded list of names.

Parameters

names	input list of names
reducednames	ouput set of reduced names

16.80.2.3 operator()()

```
string cPubchemNameReader::operator() (
            const string & cid )  [inline]
```

Overloaded operator ().

Parameters

cid	CID

Return values

string	list of names

The documentation for this struct was generated from the following file:

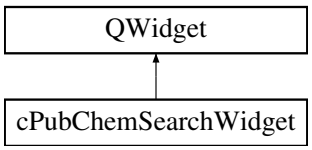
- gui/cPubChemSearchWidget.h

16.81 cPubChemSearchWidget Class Reference

The widget representing the PubChem search widget.

`#include <cPubChemSearchWidget.h>`

Inheritance diagram for cPubChemSearchWidget:



Public Member Functions

- **cPubChemSearchWidget** ()

  The constructor.
- ∼**cPubChemSearchWidget** ()

  The destructor.
- void setHTML (string html)

  Set the HTML text into the widget.
- string getHTML ()

  Get the HTML text from the widget.
- void closeEvent (QCloseEvent ∗event)

  Handle the window close event.
- void store (ofstream &os)

  Store the content into an output stream.
- void load (ifstream &is)

  Load the content from an input stream.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.

16.81.1 Detailed Description

The widget representing the PubChem search widget.

16.81.2 Member Function Documentation

16.81.2.1 closeEvent()

```
void cPubChemSearchWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.81.2.2 getHTML()

```
string cPubChemSearchWidget::getHTML ( )
```

Get the HTML text from the widget.

Return values

string	HTML string

16.81.2.3 keyPressEvent()

```
void cPubChemSearchWidget::keyPressEvent (
```

```
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.81.2.4 load()

```
void cPubChemSearchWidget::load (
            ifstream & is )
```

Load the content from an input stream.

Parameters

is	an input stream

16.81.2.5 setHTML()

```
void cPubChemSearchWidget::setHTML (
            string html )
```

Set the HTML text into the widget.

Parameters

html	string with the HTML text

16.81.2.6 store()

```
void cPubChemSearchWidget::store (
            ofstream & os )
```

Store the content into an output stream.

Parameters

os	an output stream

The documentation for this class was generated from the following files:

- gui/cPubChemSearchWidget.h
- gui/cPubChemSearchWidget.cpp

16.82 cRowItemInfo Struct Reference

An auxiliary row info structure.

`#include <cPubChemSearchWidget.h>`

Public Attributes

- int **id**

  Identification ID.

- double **theoreticalmz**

  Theoretical m/z.
- double **experimentalint**

  Experimental intensity.
- string **iontype**

  Ion type.

16.82.1 Detailed Description

An auxiliary row info structure.
The documentation for this struct was generated from the following file:

- gui/cPubChemSearchWidget.h

16.83 cSequence Class Reference

The class representing a sequence.
`#include <cSequence.h>`

Public Member Functions

- **cSequence** ()

  The constructor.
- void **clear** ()

  Clear the structure.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is, int fileversionpart1, int fileversionpart2, int fileversionpart3)

  Load the structure from an input stream.
- ePeptideType getPeptideType ()

  Get peptide type.
- string & getSequence ()

  Get sequence.
- string & getName ()

  Get name.
- double getMinimumRetentionTime ()

  Get minimum retention time.
- double getMaximumRetentionTime ()

  Get maximum retention time.
- string & getReference ()

  Get reference.
- void setPeptideType (ePeptideType peptidetype)

  Set peptide type.
- void setSequence (const string &sequence)

  Set sequence.
- void setName (const string &name)

  Set name.
- void setRetentionTime (double rtmin, double rtmax)

  Set retention time.
- void setReference (const string &reference)

  Set reference.
- void setNTterminalModification (const string &modification)

Set N-terminal modification.

- void setCTterminalModification (const string &modification)

  Set C-terminal modification.
- void setBranchModification (const string &modification)

  Set branch modification.
- string & getNTterminalModification ()

  Get N-terminal modification.
- string & getCTterminalModification ()

  Get C-terminal modification.
- string & getBranchModification ()

  Get branch modification.
- string getNameWithReferenceAsHTMLString ()

  Get name of sequence as a HTML link to sequence reference.
- void setSummaryFormula (const string &formula)

  Set the summary formula.
- string getSummaryFormula ()

  Get the summary formula.
- bool isDecoy ()

  Check if sequence/compound is decoy.
- void setDecoy (bool decoy)

  Set decoy flag.

16.83.1 Detailed Description

The class representing a sequence.

16.83.2 Member Function Documentation

16.83.2.1 getBranchModification()

```
string & cSequence::getBranchModification ( )
```

Get branch modification.

Return values

string	name of a branch modification

16.83.2.2 getCTterminalModification()

```
string & cSequence::getCTterminalModification ( )
```

Get C-terminal modification.

Return values

string	name of C-terminal modification

16.83.2.3 getMaximumRetentionTime()

```
double cSequence::getMaximumRetentionTime ( )
```

Get maximum retention time.

Return values

double	maximum retention time

16.83.2.4 getMinimumRetentionTime()

```
double cSequence::getMinimumRetentionTime ( )
```

Get minimum retention time.

Return values

double	minimum retention time

16.83.2.5 getName()

```
string & cSequence::getName ( )
```

Get name.

Return values

string	name

16.83.2.6 getNameWithReferenceAsHTMLString()

```
string cSequence::getNameWithReferenceAsHTMLString ( )
```

Get name of sequence as a HTML link to sequence reference.

Return values

string	HTML link

16.83.2.7 getNTterminalModification()

```
string & cSequence::getNTterminalModification ( )
```

Get N-terminal modification.

Return values

string	name of N-terminal modification

16.83.2.8 getPeptideType()

```
ePeptideType cSequence::getPeptideType ( )
```

Get peptide type.

Return values

ePeptideType	peptide type

16.83.2.9 getReference()

```
string & cSequence::getReference ( )
```

Get reference.

Return values

string	reference

16.83.2.10 getSequence()

```
string & cSequence::getSequence ( )
```

Get sequence.

Return values

string	sequence

16.83.2.11 getSummaryFormula()

```
string cSequence::getSummaryFormula ( )
```

Get the summary formula.

Return values

string	summary formula

16.83.2.12 isDecoy()

```
bool cSequence::isDecoy ( )
```

Check if sequence/compound is decoy.

Return values

bool	true if sequence is decoy, false otherwise

16.83.2.13 load()

```
void cSequence::load (
            ifstream & is,
            int fileversionpart1,
            int fileversionpart2,
            int fileversionpart3 )
```

Load the structure from an input stream.

Parameters

is	an input stream
fileversionpart1	first number of .res the file version
fileversionpart2	second number of .res the file version
fileversionpart3	third number of .res the file version

16.83.2.14 setBranchModification()

```
void cSequence::setBranchModification (
            const string & modification )
```

Set branch modification.

Parameters

modification	name of a branch modification

16.83.2.15 setCTterminalModification()

```
void cSequence::setCTterminalModification (
            const string & modification )
```

Set C-terminal modification.

Parameters

modification	name of a C-terminal modification

16.83.2.16 setDecoy()

```
void cSequence::setDecoy (
            bool decoy )
```

Set decoy flag.

Parameters

decoy	true if sequence is decoy, false otherwise

16.83.2.17 setName()

```
void cSequence::setName (
            const string & name )
```

Set name.

Parameters

name	name

16.83.2.18 setNTterminalModification()

```
void cSequence::setNTterminalModification (
            const string & modification )
```

Set N-terminal modification.

Parameters

modification	name of an N-terminal modification

16.83.2.19 setPeptideType()

```
void cSequence::setPeptideType (
            ePeptideType peptidetype )
```

Set peptide type.

Parameters

peptidetype	peptide type

16.83.2.20 setReference()

```
void cSequence::setReference (
            const string & reference )
```

Set reference.

Parameters

reference	reference

16.83.2.21 setRetentionTime()

```
void cSequence::setRetentionTime (
            double rtmin,
            double rtmax )
```

Set retention time.

Parameters

rtmin	minimum retention time
rtmax	maximum retention time

16.83.2.22 setSequence()

```
void cSequence::setSequence (
            const string & sequence )
```

Set sequence.

Parameters

sequence	sequence

16.83.2.23 setSummaryFormula()

```
void cSequence::setSummaryFormula (
            const string & formula )
```

Set the summary formula.

Parameters

formula	summary formula

16.83.2.24 store()

```
void cSequence::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this class was generated from the following files:

- core/cSequence.h
- core/cSequence.cpp

16.84 cSequenceDatabase Class Reference

The class representing a database of sequences.
`#include <cSequenceDatabase.h>`

Public Member Functions

- **cSequenceDatabase** ()
  The constructor.
- void loadFromPlainTextStream (ifstream &stream, bool keeprt)
  Load the database of sequences from a plain text stream.
- void storeToPlainTextStream (ofstream &stream, bool keeprt)
  Store the database of sequences into a plain text stream.
- int size ()
  Get the number of sequences in the database.
- cSequence & operator[ ] (int position)
  Overloaded operator [].
- void **clear** ()
  Clear the vector of sequences.
- void push_back (cSequence &sequence)
  Push a new sequence into the vector of sequences.
- void erase (int index)
  Erase a sequence.
- void store (ofstream &os)
  Store the structure into an output stream.
- void load (ifstream &is, int fileversionpart1, int fileversionpart2, int fileversionpart3)

Load the structure from an input stream.

- void attachDecoys (bool &terminatecomputation)

  Attach decoys.

16.84.1 Detailed Description

The class representing a database of sequences.

16.84.2 Member Function Documentation

16.84.2.1 attachDecoys()

```
void cSequenceDatabase::attachDecoys (
            bool & terminatecomputation )
```

Attach decoys.

Parameters

terminatecomputation	reference to a variable determining that the computation must be stopped

16.84.2.2 erase()

```
void cSequenceDatabase::erase (
            int index )
```

Erase a sequence.

Parameters

index	index of a sequence

16.84.2.3 load()

```
void cSequenceDatabase::load (
            ifstream & is,
            int fileversionpart1,
            int fileversionpart2,
            int fileversionpart3 )
```

Load the structure from an input stream.

Parameters

is	an input stream
fileversionpart1	first number of .res the file version
fileversionpart2	second number of .res the file version
fileversionpart3	third number of .res the file version

16.84.2.4 loadFromPlainTextStream()

```
void cSequenceDatabase::loadFromPlainTextStream (
```

```
            ifstream & stream,
            bool keeprt )
```

Load the database of sequences from a plain text stream.

Parameters

stream	reference to an input file stream
keeprt	keep retention time as a part of the compound name

16.84.2.5 operator[]()

```
cSequence & cSequenceDatabase::operator[] (
            int position )
```

Overloaded operator [].

Parameters

position	of the sequence in the vector of sequences

Return values

cSequence	reference to a sequence

16.84.2.6 push_back()

```
void cSequenceDatabase::push_back (
            cSequence & sequence )
```

Push a new sequence into the vector of sequences.

Parameters

sequence	an inserted sequence

16.84.2.7 size()

```
int cSequenceDatabase::size ( )
```

Get the number of sequences in the database.

Return values

int	number of sequences

16.84.2.8 store()

```
void cSequenceDatabase::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

16.84.2.9 storeToPlainTextStream()

```
void cSequenceDatabase::storeToPlainTextStream (
            ofstream & stream,
            bool keeprt )
```

Store the database of sequences into a plain text stream.

Parameters

stream	reference to an output file stream
keeprt	keep retention time as a part of the compound name

The documentation for this class was generated from the following files:

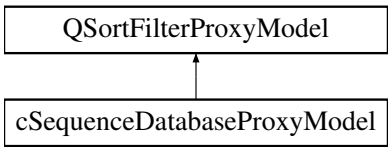
- core/cSequenceDatabase.h
- core/cSequenceDatabase.cpp

16.85 cSequenceDatabaseProxyModel Class Reference

A proxy model for editor of database of sequences.

`#include <cSequenceDatabaseProxyModel.h>`

Inheritance diagram for cSequenceDatabaseProxyModel:



Public Member Functions

- cSequenceDatabaseProxyModel (QObject ∗parent=0)

  The constructor.
- void initialize (QComboBox ∗rowsfilteroperator, QComboBox ∗rowsfiltercombobox1, QComboBox ∗rowsfiltercomparatorcombobox1, QLineEdit ∗rowsfilterline1, QComboBox ∗rowsfiltercombobox2, QComboBox ∗rowsfiltercomparatorcombobox2, QLineEdit ∗rowsfilterline2)

  Initialize the model.
- void setWholeWord (bool wholeword)

  Set whole word option.

Protected Member Functions

- bool filterAcceptsRow (int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE

  Reimplementation of filterAcceptsRow.
- bool lessThan (const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE

  Reimplementation of lessThan.

16.85.1 Detailed Description

A proxy model for editor of database of sequences.

16.85.2 Constructor & Destructor Documentation

16.85.2.1 cSequenceDatabaseProxyModel()

```
cSequenceDatabaseProxyModel::cSequenceDatabaseProxyModel (
            QObject * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.85.3 Member Function Documentation

16.85.3.1 filterAcceptsRow()

```
bool cSequenceDatabaseProxyModel::filterAcceptsRow (
            int sourceRow,
            const QModelIndex & sourceParent ) const  [protected]
```

Reimplementation of filterAcceptsRow.

Parameters

sourceRow	int
sourceParent	QModelIndex

Return values

bool	bool

16.85.3.2 initialize()

```
void cSequenceDatabaseProxyModel::initialize (
            QComboBox * rowsfilteroperator,
            QComboBox * rowsfiltercombobox1,
            QComboBox * rowsfiltercomparatorcombobox1,
            QLineEdit * rowsfilterline1,
            QComboBox * rowsfiltercombobox2,
            QComboBox * rowsfiltercomparatorcombobox2,
            QLineEdit * rowsfilterline2 )
```

Initialize the model.

Parameters

rowsfilteroperator	reference to the combobox with operator selection
rowsfiltercombobox1	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox1	reference to the combobox with selection of comparator
rowsfilterline1	reference to a text field
rowsfiltercombobox2	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox2	reference to the combobox with selection of comparator

Parameters

rowsfilterline2	reference to a text field

16.85.3.3 lessThan()

```
bool cSequenceDatabaseProxyModel::lessThan (
            const QModelIndex & left,
            const QModelIndex & right ) const  [protected]
```

Reimplementation of lessThan.

Parameters

left	QModelIndex
right	QModelIndex

Return values

bool	bool

16.85.3.4 setWholeWord()

```
void cSequenceDatabaseProxyModel::setWholeWord (
            bool wholeword )
```

Set whole word option.

Parameters

wholeword	true if whole words are matched, false otherwise

The documentation for this class was generated from the following files:

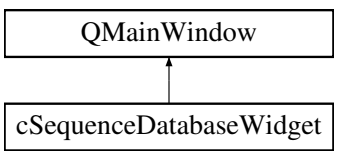
- gui/cSequenceDatabaseProxyModel.h
- gui/cSequenceDatabaseProxyModel.cpp

16.86 cSequenceDatabaseWidget Class Reference

An editor of sequences database.

`#include <cSequenceDatabaseWidget.h>`

Inheritance diagram for cSequenceDatabaseWidget:



Public Member Functions

- cSequenceDatabaseWidget (cGlobalPreferences ∗globalpreferences, QWidget ∗parent=(QWidget ∗) 0)

  The constructor.

- ~**cSequenceDatabaseWidget** ()

  The destructor.
- void closeEvent (QCloseEvent *event)

  Handle the window close event.
- void importSequence (int peptidetypeindex, QString sequence)

  Import a new sequence.
- void applyGlobalPreferences (cGlobalPreferences *globalpreferences)

  Apply new global preferences.

Protected Member Functions

- void keyPressEvent (QKeyEvent *event)

  Handle a key press event.

16.86.1 Detailed Description

An editor of sequences database.

16.86.2 Constructor & Destructor Documentation

16.86.2.1 cSequenceDatabaseWidget()

```
cSequenceDatabaseWidget::cSequenceDatabaseWidget (
            cGlobalPreferences * globalpreferences,
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.86.3 Member Function Documentation

16.86.3.1 applyGlobalPreferences()

```
void cSequenceDatabaseWidget::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.86.3.2 closeEvent()

```
void cSequenceDatabaseWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.86.3.3 importSequence()

```
void cSequenceDatabaseWidget::importSequence (
            int peptidetypeindex,
            QString sequence )
```

Import a new sequence.

Parameters

peptidetypeindex	an index of current peptide type
sequence	sequence

16.86.3.4 keyPressEvent()

```
void cSequenceDatabaseWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

The documentation for this class was generated from the following files:

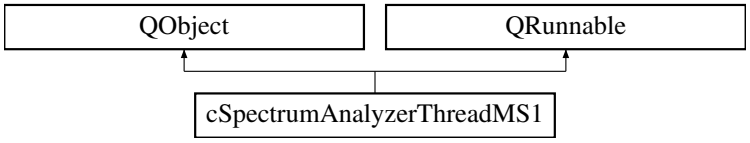
- gui/cSequenceDatabaseWidget.h
- gui/cSequenceDatabaseWidget.cpp

16.87 cSpectrumAnalyzerThreadMS1 Class Reference

The class representing a thread for an initial check of LC-MS and MSI spectra.

`#include <cSpectrumAnalyzerThreadMS1.h>`

Inheritance diagram for cSpectrumAnalyzerThreadMS1:



Public Member Functions

- void initialize (cMainThread *mainthread, cParameters *parameters, int peaklistseriesvectorid, cTheoreticalSpectrum *poolspectrum, vector< vector< int > > *hintsindex, bool *terminatecomputation)

  Initialize the spectrum analyzer.

Protected Member Functions

- void **run** ()

  The main method of the thread.

16.87.1 Detailed Description

The class representing a thread for an initial check of LC-MS and MSI spectra.

16.87.2 Member Function Documentation

16.87.2.1 initialize()

```
void cSpectrumAnalyzerThreadMS1::initialize (
            cMainThread * mainthread,
            cParameters * parameters,
            int peaklistseriesvectorid,
            cTheoreticalSpectrum * poolspectrum,
            vector< vector< int > > * hintsindex,
            bool * terminatecomputation )
```

Initialize the spectrum analyzer.

Parameters

mainthread	pointer to the main thread
parameters	pointer to program parameters
peaklistseriesvectorid	id of a peaklistseriesvector
poolspectrum	input pool of theoretical peaks
hintsindex	output vector of ids of experimental spectra in which a theoretical peak was found; for every theoretical peak
terminatecomputation	pointer to a variable determining that the thread must be stopped

The documentation for this class was generated from the following files:

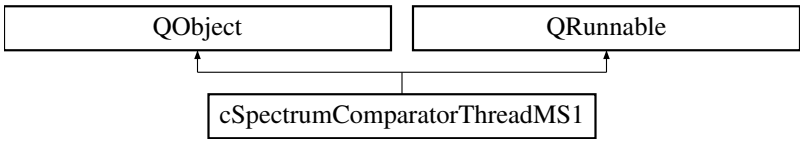
- parallel/cSpectrumAnalyzerThreadMS1.h
- parallel/cSpectrumAnalyzerThreadMS1.cpp

16.88 cSpectrumComparatorThreadMS1 Class Reference

The class representing a thread for comparison of a theoretical spectrum with a peak list (MS1 level).

`#include <cSpectrumComparatorThreadMS1.h>`

Inheritance diagram for cSpectrumComparatorThreadMS1:



Public Member Functions

- void initialize (cMainThread ∗mainthread, cParameters ∗parameters, int peaklistseriesvectorid, cTheoreticalSpectrum ∗poolspectrum, vector< vector< int > > ∗hintsindex, cTheoreticalSpectrumList ∗theoreticalspectrumlist, vector< cPeaksList > ∗unmatchedpeaks, bool ∗terminatecomputation)

  Initialize the spectrum comparator.

Protected Member Functions

- void **run** ()

  The main method of the thread.

16.88.1 Detailed Description

The class representing a thread for comparison of a theoretical spectrum with a peak list (MS1 level).

16.88.2 Member Function Documentation

16.88.2.1 initialize()

```
void cSpectrumComparatorThreadMS1::initialize (
            cMainThread * mainthread,
            cParameters * parameters,
            int peaklistseriesvectorid,
            cTheoreticalSpectrum * poolspectrum,
            vector< vector< int > > * hintsindex,
            cTheoreticalSpectrumList * theoreticalspectrumlist,
            vector< cPeaksList > * unmatchedpeaks,
            bool * terminatecomputation )
```

Initialize the spectrum comparator.

Parameters

mainthread	pointer to the main thread
parameters	pointer to program parameters
peaklistseriesvectorid	id of a peaklistseriesvector
poolspectrum	input pool of theoretical peaks
hintsindex	input vector of ids of experimental spectra in which a theoretical peak was found; for every theoretical peak
theoreticalspectrumlist	output list of matched peaks
unmatchedpeaks	output list of unmatched peaks
terminatecomputation	pointer to a variable determining that the thread must be stopped

The documentation for this class was generated from the following files:

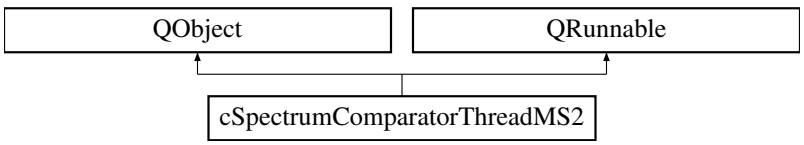
- parallel/cSpectrumComparatorThreadMS1.h
- parallel/cSpectrumComparatorThreadMS1.cpp

16.89 cSpectrumComparatorThreadMS2 Class Reference

The class representing a thread for comparison of a theoretical spectrum with a peak list (MS2 level).

`#include <cSpectrumComparatorThreadMS2.h>`

Inheritance diagram for cSpectrumComparatorThreadMS2:



Public Member Functions

- void initialize (cCandidate &candidate, cPeaksList &peaklist, cBricksDatabase *bricksdatabasewithcombinations, cTheoreticalSpectrumList *theoreticalspectrumlist, cParameters *parameters, regex *rxsequencetag, regex *rxsearchedsequence, double currentworstscore, bool *terminatecomputation)

  Initialize the spectrum comparator.

Protected Member Functions

- void **run** ()

  The main method of the thread.

16.89.1 Detailed Description

The class representing a thread for comparison of a theoretical spectrum with a peak list (MS2 level).

16.89.2 Member Function Documentation

16.89.2.1 initialize()

```
void cSpectrumComparatorThreadMS2::initialize (
            cCandidate & candidate,
            cPeaksList & peaklist,
            cBricksDatabase * bricksdatabasewithcombinations,
            cTheoreticalSpectrumList * theoreticalspectrumlist,
            cParameters * parameters,
            regex * rxsequencetag,
            regex * rxsearchedsequence,
            double currentworstscore,
            bool * terminatecomputation )
```

Initialize the spectrum comparator.

Parameters

candidate	reference to a peptide sequence candidate
peaklist	reference to a peak list
bricksdatabasewithcombinations	pointer to a database of bricks with combinations of bricks
theoreticalspectrumlist	pointer to the result list of theoretical spectra
parameters	pointer to program parameters
rxsequencetag	pointer to a regular expression of a sequence tag
rxsearchedsequence	pointer to a regular expression of a searched sequence
currentworstscore	worst score in the resulting set of theoretical spectra
terminatecomputation	reference to a variable determining that the thread must be stopped

The documentation for this class was generated from the following files:

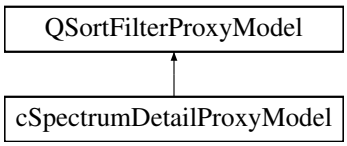
- parallel/cSpectrumComparatorThreadMS2.h
- parallel/cSpectrumComparatorThreadMS2.cpp

16.90 cSpectrumDetailProxyModel Class Reference

A table proxy model.

`#include <cSpectrumDetailProxyModel.h>`

Inheritance diagram for cSpectrumDetailProxyModel:



Public Member Functions

- cSpectrumDetailProxyModel (QObject ∗parent=0)

  The constructor.
- void setFlags (bool hidematched, bool hideunmatched, bool hidescrambled, bool hideionseries, bool hideneutralloss, bool hiderotations, bool hidetrotations)

  Set the flags for filtering.

Protected Member Functions

- bool filterAcceptsRow (int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE

  Reimplementation of filterAcceptsRow.
- bool lessThan (const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE

  Reimplementation of lessThan.

16.90.1 Detailed Description

A table proxy model.

16.90.2 Constructor & Destructor Documentation

16.90.2.1 cSpectrumDetailProxyModel()

```
cSpectrumDetailProxyModel::cSpectrumDetailProxyModel (
            QObject * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.90.3 Member Function Documentation

16.90.3.1 filterAcceptsRow()

```
bool cSpectrumDetailProxyModel::filterAcceptsRow (
            int sourceRow,
            const QModelIndex & sourceParent ) const  [protected]
```

Reimplementation of filterAcceptsRow.

Parameters

sourceRow	int
sourceParent	QModelIndex

Return values

bool	bool

16.90.3.2 lessThan()

```
bool cSpectrumDetailProxyModel::lessThan (
            const QModelIndex & left,
            const QModelIndex & right ) const  [protected]
```

Reimplementation of lessThan.

Parameters

left	QModelIndex
right	QModelIndex

Return values

bool	bool

16.90.3.3 setFlags()

```
void cSpectrumDetailProxyModel::setFlags (
            bool hidematched,
            bool hideunmatched,
            bool hidescrambled,
            bool hideionseries,
            bool hideneutralloss,
            bool hiderotations,
            bool hidetrotations )
```

Set the flags for filtering.

Parameters

hidematched	hide matched peaks
hideunmatched	hide unmatched peaks
hidescrambled	hide scrambled peaks
hideionseries	hide ion series
hideneutralloss	hide neutral losses
hiderotations	hide rotations of a cyclic peptide or branch-cyclic peptide or cyclic polyketide
hidetrotations	hide trotations of a branched or branch-cyclic peptide

The documentation for this class was generated from the following files:

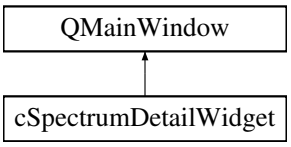
- gui/cSpectrumDetailProxyModel.h
- gui/cSpectrumDetailProxyModel.cpp

16.91 cSpectrumDetailWidget Class Reference

The class representing a widget for visualisation of a detail of a spectrum (opened after doubleclick on a row when results are reported).

`#include <cSpectrumDetailWidget.h>`

Inheritance diagram for cSpectrumDetailWidget:



Signals

- void emitMZInterval (double minmz, double maxmz)

  The signal is emitted when the range of m/z ratios has been changed.
- void rawDataStateChangedSignal (bool state)

  The signal is emitted when raw data state was changed.
- void absoluteIntensityStateChangedSignal (bool state)

  The signal is emitted when the absolute intensity state was changed.
- void lastActiveDetail (int rowid)

  The signal is emitted when the widget has been moved.

Public Member Functions

- **cSpectrumDetailWidget** ()

  The constructor.
- cSpectrumDetailWidget (const cSpectrumDetailWidget &sd)

  The copy constructor.
- cSpectrumDetailWidget & operator= (const cSpectrumDetailWidget &sd)

  Overloaded operator=.
- ~**cSpectrumDetailWidget** ()

  The destructor.
- void closeEvent (QCloseEvent *event)

  Handle the window close event.
- void initialize (int rowid, string scantitle, int activefileid, cGlobalPreferences *globalpreferences, cParameters *parameters, cTheoreticalSpectrum &theoreticalspectrum, QWidget *parent)

  Initialize the widget.
- cTheoreticalSpectrum & getTheoreticalSpectrum ()

  Get the theoretical spectrum.
- string getDetailsAsHTMLString ()

  Get details about a theoretical spectrum as a HTML string.
- string getPeaksTableAsHTMLString (bool unmatchedtheoreticalpeaks, bool unmatchedexperimentalpeaks, bool exportisomers)

  Get a table of peaks as a HTML string.
- string getPartialPeaksTableAsHTMLString (int id, bool exportisomers)

  Get a partial table of peaks as a HTML string.
- void prepareToShow (QAction *actionShowIsomers, cPeakListSeries *rawdata, vector< cImzMLItem > *imzmlprofilemetadata, bool profilemz64precision, bool profileintensity64precision, bool usedefaultsize)

  Prepare the widget to show.
- void findAll (const QString &str, QTextDocument::FindFlags opt=0, bool errormessage=true)

  Find all occurrences of str and highlight them.
- void exportImage (bool exportspectrum)

  Export spectrum/peptide image.
- void zoomToPeak (double value)

  Zoom to a peak determined by the value.
- set< string > & getLocalNeutralLosses ()

  Get a set of local neutral losses.

- void setProfileSpectrumEnabled (bool enable)

  Set the state of profile spectrum visualization.
- bool hasProfileSpectrumEnabled ()

  Check if the profile spectrum visualization is enabled.
- void **disableProfileMode** ()

  Permanently disable the visualization in profile mode.
- void setAbsoluteIntensityEnabled (bool enable)

  Set the state of absolute intensity mode.
- bool hasAbsoluteIntensityEnabled ()

  Check if the absolute intensity mode is enabled.
- void setHideMatchedPeaksEnabled (bool enable)

  Set the state of hide matched peaks mode.
- bool hasHideMatchedPeaksEnabled ()

  Check if the hide matched peaks mode is enabled.
- void setHideUnmatchedPeaksEnabled (bool enable)

  Set the state of hide unmatched peaks mode.
- bool hasHideUnmatchedPeaksEnabled ()

  Check if the hide unmatched peaks mode is enabled.
- void setHideScrambledPeaksEnabled (bool enable)

  Set the state of hide scrambled peaks mode.
- bool hasHideScrambledPeaksEnabled ()

  Check if the hide scrambled peaks mode is enabled.
- void setHidePeakLabelsEnabled (bool enable)

  Set the state of hide peak labels mode.
- bool hasHidePeakLabelsEnabled ()

  Check if the hide peak labels mode is enabled.
- void setMouseMzSelectionEnabled (bool enable)

  Set the state of mouse m/z selection tool.
- bool hasMouseMzSelectionEnabled ()

  Check if the mouse m/z selection tool is enabled.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.
- void moveEvent (QMoveEvent ∗event)

  Handle a move event.

16.91.1 Detailed Description

The class representing a widget for visualisation of a detail of a spectrum (opened after doubleclick on a row when results are reported).

16.91.2 Constructor & Destructor Documentation

16.91.2.1 cSpectrumDetailWidget()

```
cSpectrumDetailWidget::cSpectrumDetailWidget (
            const cSpectrumDetailWidget & sd )
```

The copy constructor.

Parameters

sd	reference to a cSpectrumDetailWidget

16.91.3 Member Function Documentation

16.91.3.1 absoluteIntensityStateChangedSignal

```
void cSpectrumDetailWidget::absoluteIntensityStateChangedSignal (
            bool state )  [signal]
```

The signal is emitted when the absolute intensity state was changed.

Parameters

state	current state

16.91.3.2 closeEvent()

```
void cSpectrumDetailWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.91.3.3 emitMZInterval

```
void cSpectrumDetailWidget::emitMZInterval (
            double minmz,
            double maxmz )  [signal]
```

The signal is emitted when the range of m/z ratios has been changed.

Parameters

minmz	a minimum threshold of m/z ratio
maxmz	a maximum threshold of m/z ratio

16.91.3.4 exportImage()

```
void cSpectrumDetailWidget::exportImage (
            bool exportspectrum )
```

Export spectrum/peptide image.

Parameters

exportspectrum	if true, a spectrum image is exported; if false, a peptide image is exported

16.91.3.5 findAll()

```
void cSpectrumDetailWidget::findAll (
            const QString & str,
            QTextDocument::FindFlags opt = 0,
            bool errormessage = true )
```

Find all occurrences of *str* and highlight them.

Parameters

str	search string
opt	search options
errormessage	if true, an error message in a popup window is shown when no results are matched

16.91.3.6 getDetailsAsHTMLString()

```
string cSpectrumDetailWidget::getDetailsAsHTMLString ( )
```

Get details about a theoretical spectrum as a HTML string.

Return values

string	details about a theoretical spectrum

16.91.3.7 getLocalNeutralLosses()

```
set< string > & cSpectrumDetailWidget::getLocalNeutralLosses ( )
```

Get a set of local neutral losses.

Return values

set<int>	get a set of local neutral losses

16.91.3.8 getPartialPeaksTableAsHTMLString()

```
string cSpectrumDetailWidget::getPartialPeaksTableAsHTMLString (
            int id,
            bool exportisomers )
```

Get a partial table of peaks as a HTML string.

Parameters

id	identifier of a spectrum
exportisomers	if true then isomers of building blocks are exported

Return values

string	a partial table of peaks as a HTML string

16.91.3.9 getPeaksTableAsHTMLString()

```
string cSpectrumDetailWidget::getPeaksTableAsHTMLString (
            bool unmatchedtheoreticalpeaks,
            bool unmatchedexperimentalpeaks,
            bool exportisomers )
```

Get a table of peaks as a HTML string.

Parameters

unmatchedtheoreticalpeaks	if true then unmatched theoretical peaks are included
unmatchedexperimentalpeaks	if true then unmatched experimental peaks are included
exportisomers	if true then isomers of building blocks are exported

Return values

string	a table of peaks as a HTML string

16.91.3.10 getTheoreticalSpectrum()

```
cTheoreticalSpectrum & cSpectrumDetailWidget::getTheoreticalSpectrum ( )
```

Get the theoretical spectrum.

Return values

cTheoreticalSpectrum	reference to a theoretical spectrum

16.91.3.11 hasAbsoluteIntensityEnabled()

```
bool cSpectrumDetailWidget::hasAbsoluteIntensityEnabled ( )
```

Check if the absolute intensity mode is enabled.

Return values

bool	true if the absolute intensity mode is enabled

16.91.3.12 hasHideMatchedPeaksEnabled()

```
bool cSpectrumDetailWidget::hasHideMatchedPeaksEnabled ( )
```

Check if the hide matched peaks mode is enabled.

Return values

bool	true if the hide matched peaks mode is enabled

16.91.3.13 hasHidePeakLabelsEnabled()

`bool cSpectrumDetailWidget::hasHidePeakLabelsEnabled ( )`
Check if the hide peak labels mode is enabled.

Return values

bool	true if the hide peak labels mode is enabled

16.91.3.14 hasHideScrambledPeaksEnabled()

`bool cSpectrumDetailWidget::hasHideScrambledPeaksEnabled ( )`
Check if the hide scrambled peaks mode is enabled.

Return values

bool	true if the hide scrambled peaks mode is enabled

16.91.3.15 hasHideUnmatchedPeaksEnabled()

`bool cSpectrumDetailWidget::hasHideUnmatchedPeaksEnabled ( )`
Check if the hide unmatched peaks mode is enabled.

Return values

bool	true if the hide unmatched peaks mode is enabled

16.91.3.16 hasMouseMzSelectionEnabled()

`bool cSpectrumDetailWidget::hasMouseMzSelectionEnabled ( )`
Check if the mouse m/z selection tool is enabled.

Return values

bool	true if the mouse m/z selection tool is enabled

16.91.3.17 hasProfileSpectrumEnabled()

`bool cSpectrumDetailWidget::hasProfileSpectrumEnabled ( )`
Check if the profile spectrum visualization is enabled.

Return values

bool	true if the profile spectrum is enabled

16.91.3.18 initialize()

`void cSpectrumDetailWidget::initialize (`

```
            int rowid,
            string scantitle,
            int activefileid,
            cGlobalPreferences * globalpreferences,
            cParameters * parameters,
            cTheoreticalSpectrum & theoreticalspectrum,
            QWidget * parent )
```

Initialize the widget.

Parameters

rowid	id of a row
scantitle	scan title
activefileid	id of an active file
globalpreferences	global preferences of the application
parameters	a pointer to parameters
theoreticalspectrum	a reference to a theoretical spectrum
parent	pointer to a parent widget

16.91.3.19 keyPressEvent()

```
void cSpectrumDetailWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.91.3.20 lastActiveDetail

```
void cSpectrumDetailWidget::lastActiveDetail (
            int rowid )  [signal]
```

The signal is emitted when the widget has been moved.

Parameters

rowid	rowid of the widget

16.91.3.21 moveEvent()

```
void cSpectrumDetailWidget::moveEvent (
            QMoveEvent * event )  [protected]
```

Handle a move event.

Parameters

event	pointer to QMoveEvent

16.91.3.22 operator=()

```
cSpectrumDetailWidget & cSpectrumDetailWidget::operator= (
            const cSpectrumDetailWidget & sd )
```

Overloaded operator=.

Parameters

sd	reference to a cSpectrumDetailWidget

16.91.3.23 prepareToShow()

```
void cSpectrumDetailWidget::prepareToShow (
            QAction * actionShowIsomers,
            cPeakListSeries * rawdata,
            vector< cImzMLItem > * imzmlprofilemetadata,
            bool profilemz64precision,
            bool profileintensity64precision,
            bool usedefaultsize )
```

Prepare the widget to show.

Parameters

actionShowIsomers	reference to QAction to show isomers of building blocks
rawdata	a pointer to a structure handling raw data
imzmlprofilemetadata	imzml profile metadata
profilemz64precision	true if 64 mz precision is used, false if 32 bit mz precision is used
profileintensity64precision	true if 64 intensity precision is used, false if 32 bit intensity precision is used
usedefaultsize	if true, the default size is used

16.91.3.24 rawDataStateChangedSignal

```
void cSpectrumDetailWidget::rawDataStateChangedSignal (
            bool state )  [signal]
```

The signal is emitted when raw data state was changed.

Parameters

state	current state

16.91.3.25 setAbsoluteIntensityEnabled()

```
void cSpectrumDetailWidget::setAbsoluteIntensityEnabled (
            bool enable )
```

Set the state of absolute intensity mode.

Parameters

enable	state of absolute intensity mode

16.91.3.26 setHideMatchedPeaksEnabled()

```
void cSpectrumDetailWidget::setHideMatchedPeaksEnabled (
            bool enable )
```

Set the state of hide matched peaks mode.

Parameters

enable	state of hide matched peaks mode

16.91.3.27 setHidePeakLabelsEnabled()

```
void cSpectrumDetailWidget::setHidePeakLabelsEnabled (
            bool enable )
```

Set the state of hide peak labels mode.

Parameters

enable	state of hide peak labels mode

16.91.3.28 setHideScrambledPeaksEnabled()

```
void cSpectrumDetailWidget::setHideScrambledPeaksEnabled (
            bool enable )
```

Set the state of hide scrambled peaks mode.

Parameters

enable	state of hide scrambled peaks mode

16.91.3.29 setHideUnmatchedPeaksEnabled()

```
void cSpectrumDetailWidget::setHideUnmatchedPeaksEnabled (
            bool enable )
```

Set the state of hide unmatched peaks mode.

Parameters

enable	state of hide unmatched peaks mode

16.91.3.30 setMouseMzSelectionEnabled()

```
void cSpectrumDetailWidget::setMouseMzSelectionEnabled (
            bool enable )
```

Set the state of mouse m/z selection tool.

Parameters

enable	state of mouse m/z selection tool

16.91.3.31 setProfileSpectrumEnabled()

```
void cSpectrumDetailWidget::setProfileSpectrumEnabled (
            bool enable )
```

Set the state of profile spectrum visualization.

Parameters

enable	state of profile spectrum visualization

16.91.3.32 zoomToPeak()

```
void cSpectrumDetailWidget::zoomToPeak (
            double value )
```

Zoom to a peak determined by the value.

Parameters

value	mz ratio of a peak

The documentation for this class was generated from the following files:

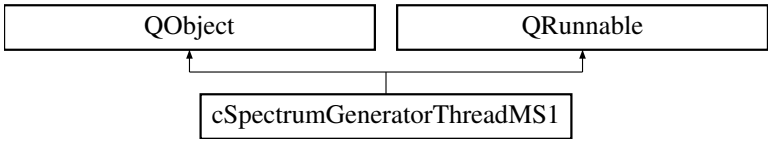
- gui/cSpectrumDetailWidget.h
- gui/cSpectrumDetailWidget.cpp

16.92 cSpectrumGeneratorThreadMS1 Class Reference

The class representing a thread for generating a theoretical spectrum (MS1 level).

`#include <cSpectrumGeneratorThreadMS1.h>`

Inheritance diagram for cSpectrumGeneratorThreadMS1:



Public Member Functions

- void initialize (cMainThread *mainthread, cParameters *parameters, int sequencestart, int sequencestop, cTheoreticalSpectrum *poolspectrum, bool *terminatecomputation)

  Initialize the spectrum generator.

Protected Member Functions

- void **run** ()

  The main method of the thread.

16.92.1 Detailed Description

The class representing a thread for generating a theoretical spectrum (MS1 level).

16.92.2 Member Function Documentation

16.92.2.1 initialize()

```
void cSpectrumGeneratorThreadMS1::initialize (
            cMainThread * mainthread,
            cParameters * parameters,
            int sequencestart,
            int sequencestop,
            cTheoreticalSpectrum * poolspectrum,
            bool * terminatecomputation )
```

Initialize the spectrum generator.

Parameters

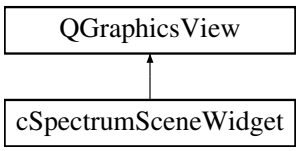
mainthread	pointer to the main thread
parameters	pointer to program parameters
sequencestart	index of the first item to be processed
sequencestop	index of the last item to be processed (excluding the last item)
poolspectrum	output pool of theoretical peaks
terminatecomputation	pointer to a variable determining that the thread must be stopped

The documentation for this class was generated from the following files:

- parallel/cSpectrumGeneratorThreadMS1.h
- parallel/cSpectrumGeneratorThreadMS1.cpp

16.93 cSpectrumSceneWidget Class Reference

The widget representing a 2D mass spectrum.
`#include <cSpectrumSceneWidget.h>`
Inheritance diagram for cSpectrumSceneWidget:



Signals

- void updateMZInterval (double minmz, double maxmz)

  The signal is emitted when the range of m/z ratios has been changed.

Public Member Functions

- cSpectrumSceneWidget (QWidget *parent=0)

  The constructor.
- ~**cSpectrumSceneWidget** ()

  The destructor.
- void initialize (cParameters *parameters, cTheoreticalSpectrum *theoreticalspectrum, cPeakListSeries *rawdata, int rowid, int activefileid)

  Initialize the widget.

- void exportToPDF (QString filename, bool postscript)

  Export spectrum scene into a PDF or a PS file.
- void exportToSVG (QString filename)

  Export spectrum scene into a SVG file.
- void exportToPNG (QString filename)

  Export spectrum scene into a PNG file.
- void hideUnmatchedPeaks (bool state)

  Hide/Show unmatched peaks.
- void hideMatchedPeaks (bool state)

  Hide/Show matched peaks.
- void hideScrambledPeaks (bool state)

  Hide/Show scrambled peaks.
- void hidePeakLabels (bool state)

  Hide/Show labels of peaks.

Protected Member Functions

- void wheelEvent (QWheelEvent ∗event)

  Handle the mouse wheel event.
- void mouseMoveEvent (QMouseEvent ∗event)

  Handle the mouse move event.
- void mouseReleaseEvent (QMouseEvent ∗event)

  Handle the mouse release event.
- void mousePressEvent (QMouseEvent ∗event)

  Handle the mouse press event.
- void showEvent (QShowEvent ∗event)

  Handle the widget show event.
- void resizeEvent (QResizeEvent ∗event)

  Handle the widget resize event.

16.93.1 Detailed Description

The widget representing a 2D mass spectrum.

16.93.2 Constructor & Destructor Documentation

16.93.2.1 cSpectrumSceneWidget()

```
cSpectrumSceneWidget::cSpectrumSceneWidget (
            QWidget * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.93.3 Member Function Documentation

16.93.3.1 exportToPDF()

```
void cSpectrumSceneWidget::exportToPDF (
            QString filename,
            bool postscript )
```

Export spectrum scene into a PDF or a PS file.

Parameters

filename	filename
postscript	if true then PS file is generated instead of PDF

16.93.3.2 exportToPNG()

```
void cSpectrumSceneWidget::exportToPNG (
            QString filename )
```

Export spectrum scene into a PNG file.

Parameters

filename	filename

16.93.3.3 exportToSVG()

```
void cSpectrumSceneWidget::exportToSVG (
            QString filename )
```

Export spectrum scene into a SVG file.

Parameters

filename	filename

16.93.3.4 hideMatchedPeaks()

```
void cSpectrumSceneWidget::hideMatchedPeaks (
            bool state )
```

Hide/Show matched peaks.

Parameters

state	the peaks are hidden when *state* is true

16.93.3.5 hidePeakLabels()

```
void cSpectrumSceneWidget::hidePeakLabels (
            bool state )
```

Hide/Show labels of peaks.

Parameters

state	the peak labels are hidden when *state* is true

16.93.3.6 hideScrambledPeaks()

```
void cSpectrumSceneWidget::hideScrambledPeaks (
            bool state )
```

Hide/Show scrambled peaks.

Parameters

state	the peaks are hidden when *state* is true

16.93.3.7 hideUnmatchedPeaks()

```
void cSpectrumSceneWidget::hideUnmatchedPeaks (
            bool state )
```

Hide/Show unmatched peaks.

Parameters

state	the peaks are hidden when *state* is true

16.93.3.8 initialize()

```
void cSpectrumSceneWidget::initialize (
            cParameters * parameters,
            cTheoreticalSpectrum * theoreticalspectrum,
            cPeakListSeries * rawdata,
            int rowid,
            int activefileid )
```

Initialize the widget.

Parameters

parameters	a pointer to parameters
theoreticalspectrum	a pointer to a theoretical spectrum
rawdata	a pointer to a structure handling raw data
rowid	id of a row
activefileid	id of an active file

16.93.3.9 mouseMoveEvent()

```
void cSpectrumSceneWidget::mouseMoveEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse move event.

Parameters

event	pointer to QMouseEvent

16.93.3.10 mousePressEvent()

```
void cSpectrumSceneWidget::mousePressEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse press event.

Parameters

event	pointer to QMouseEvent

16.93.3.11 mouseReleaseEvent()

```
void cSpectrumSceneWidget::mouseReleaseEvent (
            QMouseEvent * event )  [protected]
```

Handle the mouse release event.

Parameters

event	pointer to QMouseEvent

16.93.3.12 resizeEvent()

```
void cSpectrumSceneWidget::resizeEvent (
            QResizeEvent * event )  [protected]
```

Handle the widget resize event.

Parameters

event	pointer to QResizeEvent

16.93.3.13 showEvent()

```
void cSpectrumSceneWidget::showEvent (
            QShowEvent * event )  [protected]
```

Handle the widget show event.

Parameters

event	pointer to QShowEvent

16.93.3.14 updateMZInterval

```
void cSpectrumSceneWidget::updateMZInterval (
            double minmz,
            double maxmz )  [signal]
```

The signal is emitted when the range of m/z ratios has been changed.

Parameters

minmz	a minimum threshold of m/z ratio
maxmz	a maximum threshold of m/z ratio

16.93.3.15 wheelEvent()

```
void cSpectrumSceneWidget::wheelEvent (
            QWheelEvent * event )  [protected]
```

Handle the mouse wheel event.

Parameters

event	pointer to QWheelEvent

The documentation for this class was generated from the following files:

- gui/cSpectrumSceneWidget.h
- gui/cSpectrumSceneWidget.cpp

16.94 cSummaryFormula Class Reference

The representation of a summary formula.

`#include <cSummaryFormula.h>`

Public Member Functions

- **cSummaryFormula** ()

  The constructor.
- cSummaryFormula (string &formula, bool partial=false)

  The constructor.
- void **clear** ()

  Clear the structure.
- void setFormula (const string &formula, bool partial=false)

  Set the formula.
- string getSummary ()

  Get the summary formula as a string.
- string getFancySummary (int charge)

  Get the fancy summary formula as a string.
- bool isPartial ()

  Check if the formula is incomplete or not.
- void setPartial (bool partial)

  Set the formula as partial.
- void addFormula (string &formula, bool remove=false)

  Add a formula to the existing one.
- void setFromMap (map< string, int > &atoms, bool partial)

  Set the string formula from a map of atoms.
- bool isValid (string &errormessage)

  Check the syntax of a chemical formula.
- bool isEmpty ()

  Check if the formula is empty.
- bool hasAllElementsPositive ()

  Check if the formula has the numbers of all elements positive.
- double getMass ()

  Get mass of the formula.
- cPeaksList getIsotopePattern (double fwhm, int charge, bool positive, int maxchangedatoms, bool writedescription)

Get a theoretical isotope pattern of the formula.

- void store (ofstream &os)

  Store the structure into an output stream.

- void load (ifstream &is)

  Load the structure from an input stream.

- bool equals (cSummaryFormula &secondformula)

  Check if the object equals to another object.

16.94.1 Detailed Description

The representation of a summary formula.

16.94.2 Constructor & Destructor Documentation

16.94.2.1 cSummaryFormula()

```
cSummaryFormula::cSummaryFormula (
            string & formula,
            bool partial = false )
```

The constructor.

Parameters

formula	a chemical formula
partial	true when the formula is incomplete

16.94.3 Member Function Documentation

16.94.3.1 addFormula()

```
void cSummaryFormula::addFormula (
            string & formula,
            bool remove = false )
```

Add a formula to the existing one.

Parameters

formula	a chemical formula
remove	if true then the formula is removed; if false then the formula is added

16.94.3.2 equals()

```
bool cSummaryFormula::equals (
            cSummaryFormula & secondformula )
```

Check if the object equals to another object.

Parameters

secondformula	object for comparison

Return values

bool	true if the objects are equals

16.94.3.3 getFancySummary()

```
string cSummaryFormula::getFancySummary (
            int charge )
```

Get the fancy summary formula as a string.

Parameters

charge	peak charge

Return values

string	formula as a string

16.94.3.4 getIsotopePattern()

```
cPeaksList cSummaryFormula::getIsotopePattern (
            double fwhm,
            int charge,
            bool positive,
            int maxchangedatoms,
            bool writedescription )
```

Get a theoretical isotope pattern of the formula.

Parameters

fwhm	FWHM
charge	charge
positive	true if the charge is positive charge, false if the charge is negative
maxchangedatoms	maximum number of atoms of an element which are replaced with isotopic variants
writedescription	if true then string descriptions of peaks are filled

Return values

cPeaksList	isotopic pattern of the formula

16.94.3.5 getMass()

```
double cSummaryFormula::getMass ( )
```

Get mass of the formula.

Return values

double	mass of the formula

16.94.3.6 getSummary()

```
string cSummaryFormula::getSummary ( )
```

Get the summary formula as a string.

Return values

string	formula as a string

16.94.3.7 hasAllElementsPositive()

```
bool cSummaryFormula::hasAllElementsPositive ( )
```

Check if the formula has the numbers of all elements positive.

Return values

bool	true if the formula has the numbers of all elements positive

16.94.3.8 isEmpty()

```
bool cSummaryFormula::isEmpty ( )
```

Check if the formula is empty.

Return values

bool	true if the formula is empty; false otherwise

16.94.3.9 isPartial()

```
bool cSummaryFormula::isPartial ( )
```

Check if the formula is incomplete or not.

Return values

bool	true when the formula is incomplete, false when the formula is complete

16.94.3.10 isValid()

```
bool cSummaryFormula::isValid (
            string & errormessage )
```

Check the syntax of a chemical formula.

Parameters

errormessage	an error message if formula is invalid

Return values

bool	true when formula is valid

16.94.3.11 load()

```
void cSummaryFormula::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.94.3.12 setFormula()

```
void cSummaryFormula::setFormula (
            const string & formula,
            bool partial = false )
```

Set the formula.

Parameters

formula	a chemical formula
partial	true when the formula is incomplete

16.94.3.13 setFromMap()

```
void cSummaryFormula::setFromMap (
            map< string, int > & atoms,
            bool partial )
```

Set the string formula from a map of atoms.

Parameters

atoms	an input map of atoms
partial	true if the formula is incomplete

16.94.3.14 setPartial()

```
void cSummaryFormula::setPartial (
            bool partial )
```

Set the formula as partial.

Parameters

partial	true if the formula is incomplete

16.94.3.15 store()

```
void cSummaryFormula::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this class was generated from the following files:

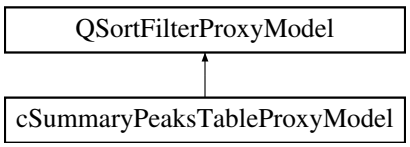
- core/cSummaryFormula.h
- core/cSummaryFormula.cpp

16.95 cSummaryPeaksTableProxyModel Class Reference

A table proxy model.

`#include <cSummaryPeaksTableProxyModel.h>`

Inheritance diagram for cSummaryPeaksTableProxyModel:



Public Member Functions

- cSummaryPeaksTableProxyModel (QObject *parent=0)

  The constructor.
- void initialize (eModeType mode, ePeakListFileFormat peaklistfileformat, bool generateisotopepattern, QComboBox *rowsfilteroperator, QComboBox *rowsfiltercombobox1, QComboBox *rowsfiltercomparatorcombobox1, QLineEdit *rowsfilterline1, QComboBox *rowsfiltercombobox2, QComboBox *rowsfiltercomparatorcombobox2, QLineEdit *rowsfilterline2)

  Initialize the model.
- void setRegion (int xmin, int xmax, int ymin, int ymax)

  Set the region for filtering.
- void setWholeWord (bool wholeword)

  Set whole word option.

Public Attributes

- int **xmin**

  The minimum X coordinate.
- int **xmax**

  The maximum X coordinate.
- int **ymin**

  The minimum Y coordinate.
- int **ymax**

  The maximum Y coordinate.

Protected Member Functions

- bool filterAcceptsRow (int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE

  Reimplementation of filterAcceptsRow.
- bool lessThan (const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE

  Reimplementation of lessThan.

16.95.1 Detailed Description

A table proxy model.

16.95.2 Constructor & Destructor Documentation

16.95.2.1 cSummaryPeaksTableProxyModel()

```
cSummaryPeaksTableProxyModel::cSummaryPeaksTableProxyModel (
            QObject * parent = 0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.95.3 Member Function Documentation

16.95.3.1 filterAcceptsRow()

```
bool cSummaryPeaksTableProxyModel::filterAcceptsRow (
            int sourceRow,
            const QModelIndex & sourceParent ) const  [protected]
```

Reimplementation of filterAcceptsRow.

Parameters

sourceRow	int
sourceParent	QModelIndex

Return values

bool	bool

16.95.3.2 initialize()

```
void cSummaryPeaksTableProxyModel::initialize (
            eModeType mode,
            ePeakListFileFormat peaklistfileformat,
            bool generateisotopepattern,
            QComboBox * rowsfilteroperator,
            QComboBox * rowsfiltercombobox1,
            QComboBox * rowsfiltercomparatorcombobox1,
```

```
            QLineEdit * rowsfilterline1,
            QComboBox * rowsfiltercombobox2,
            QComboBox * rowsfiltercomparatorcombobox2,
            QLineEdit * rowsfilterline2 )
```

Initialize the model.

Parameters

mode	program mode
peaklistfileformat	peaklist format
generateisotopepattern	true if full isotope patterns are generated; false otherwise
rowsfilteroperator	reference to the combobox with operator selection
rowsfiltercombobox1	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox1	reference to the combobox with selection of comparator
rowsfilterline1	reference to a text field
rowsfiltercombobox2	reference to the combobox with selection of columns
rowsfiltercomparatorcombobox2	reference to the combobox with selection of comparator
rowsfilterline2	reference to a text field

16.95.3.3 lessThan()

```
bool cSummaryPeaksTableProxyModel::lessThan (
            const QModelIndex & left,
            const QModelIndex & right ) const  [protected]
```

Reimplementation of lessThan.

Parameters

left	QModelIndex
right	QModelIndex

Return values

bool	bool

16.95.3.4 setRegion()

```
void cSummaryPeaksTableProxyModel::setRegion (
            int xmin,
            int xmax,
            int ymin,
            int ymax )
```

Set the region for filtering.

Parameters

xmin	minimum x coordinate
xmax	maximum x coordinate
ymin	minimum y coordinate
ymax	maximum y coordinate

16.95.3.5 setWholeWord()

```
void cSummaryPeaksTableProxyModel::setWholeWord (
            bool wholeword )
```

Set whole word option.

Parameters

wholeword	true if whole words are matched, false otherwise

The documentation for this class was generated from the following files:

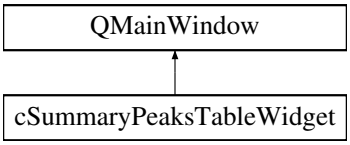
- gui/cSummaryPeaksTableProxyModel.h
- gui/cSummaryPeaksTableProxyModel.cpp

16.96 cSummaryPeaksTableWidget Class Reference

Summary table of matched peaks.

`#include <cSummaryPeaksTableWidget.h>`

Inheritance diagram for cSummaryPeaksTableWidget:



Signals

- void **tableCancelled** ()

  The table was not generated because the process was cancelled.
- void summaryPeaksTableRowDoubleClicked (int rowid, double experimentalmz)

  A row was double clicked.
- void **resetRegion** ()

  Reset the region in image window.
- void sendFilterOptionsToImageWindow (vector< cCoordinateInfo > coordinateinfo, bool operatortype, string columnname1, string comparatorname1, string filterstring1, string columnname2, string comparatorname2, string filterstring2, bool casesensitive, bool wholeword)

  Send the vector of aggregated informations about coordinates to image window.
- void sendFilterOptionsToChromatogram (cPeaksList eic)

  Send the EIC chromatogram to chromatogram window.

Public Member Functions

- cSummaryPeaksTableWidget (cGlobalPreferences *globalpreferences, QWidget *parent=(QWidget *) 0)

  The constructor.
- ~**cSummaryPeaksTableWidget** ()

  The destructor.
- void closeEvent (QCloseEvent *event)

  Handle the window close event.
- bool prepareToShow (QStandardItemModel *resultsstandardmodel, cMainWindowProxyModel *resultsproxymodel, cParameters *parameters, cTheoreticalSpectrumList *listoftheoreticalspectra, int activefileid, bool showisomers)

Prepare the widget to show.

- void **deleteTable** ()

  Delete the table content.

- void updateFilterBySelectedRegion (int xmin, int xmax, int ymin, int ymax)

  Update the filter using an image region.

- void applyGlobalPreferences (cGlobalPreferences ∗globalpreferences)

  Apply new global preferences.

- cPubChemSearchWidget ∗ getPubChemSearchWidget ()

  Get the reference to cPubChemSearchWidget.

- void **filterTablerows** ()

  Filter table rows.

- void filterCompound (string name, string iontype, int datatypeview, string mzstr)

  Filter a specific compound.

Protected Member Functions

- void keyPressEvent (QKeyEvent ∗event)

  Handle a key press event.

16.96.1 Detailed Description

Summary table of matched peaks.

16.96.2 Constructor & Destructor Documentation

16.96.2.1 cSummaryPeaksTableWidget()

```
cSummaryPeaksTableWidget::cSummaryPeaksTableWidget (
            cGlobalPreferences * globalpreferences,
            QWidget * parent = (QWidget *)0 )
```

The constructor.

Parameters

globalpreferences	global preferences of the application
parent	pointer to a parent widget

16.96.3 Member Function Documentation

16.96.3.1 applyGlobalPreferences()

```
void cSummaryPeaksTableWidget::applyGlobalPreferences (
            cGlobalPreferences * globalpreferences )
```

Apply new global preferences.

Parameters

globalpreferences	global preferences of the application

16.96.3.2 closeEvent()

```
void cSummaryPeaksTableWidget::closeEvent (
            QCloseEvent * event )
```

Handle the window close event.

Parameters

event	pointer to QCloseEvent

16.96.3.3 filterCompound()

```
void cSummaryPeaksTableWidget::filterCompound (
            string name,
            string iontype,
            int datatypeview,
            string mzstr )
```

Filter a specific compound.

Parameters

name	compound name
iontype	ion type
datatypeview	type of data shown
mzstr	theoretical m/z value converted to a string

16.96.3.4 getPubChemSearchWidget()

```
cPubChemSearchWidget * cSummaryPeaksTableWidget::getPubChemSearchWidget ( )
```

Get the reference to cPubChemSearchWidget.

Return values

cPubChemSearchWidget	reference to cPubChemSearchWidget

16.96.3.5 keyPressEvent()

```
void cSummaryPeaksTableWidget::keyPressEvent (
            QKeyEvent * event )  [protected]
```

Handle a key press event.

Parameters

event	pointer to QKeyEvent

16.96.3.6 prepareToShow()

```
bool cSummaryPeaksTableWidget::prepareToShow (
            QStandardItemModel * resultsstandardmodel,
            cMainWindowProxyModel * resultsproxymodel,
```

```
            cParameters * parameters,
            cTheoreticalSpectrumList * listoftheoreticalspectra,
            int activefileid,
            bool showisomers )
```

Prepare the widget to show.

Parameters

resultsstandardmodel	standard model of the tableview in the main application window
resultsproxymodel	proxy model of the tableview in the main application window
parameters	parameters of the application
listoftheoreticalspectra	lists of theoretical spectra
activefileid	id of the active file
showisomers	true if isomers of blocks are reported; false otherwise

Return values

bool	true if the table was successfully prepared, false otherwise

16.96.3.7 sendFilterOptionsToChromatogram

```
void cSummaryPeaksTableWidget::sendFilterOptionsToChromatogram (
            cPeaksList eic ) [signal]
```

Send the EIC chromatogram to chromatogram window.

Parameters

eic	eic chromatogram

16.96.3.8 sendFilterOptionsToImageWindow

```
void cSummaryPeaksTableWidget::sendFilterOptionsToImageWindow (
            vector< cCoordinateInfo > coordinateinfo,
            bool operatortype,
            string columnname1,
            string comparatorname1,
            string filterstring1,
            string columnname2,
            string comparatorname2,
            string filterstring2,
            bool casesensitive,
            bool wholeword ) [signal]
```

Send the vector of aggregated informations about coordinates to image window.

Parameters

coordinateinfo	a vector of aggregated informations about coordinates
operatortype	the type of operator (0 = or; 1 = and)
columnname1	name of column which was compared
comparatorname1	name of used comparator
filterstring1	a string used to filter the points

Parameters

columnname2	name of column which was compared
comparatorname2	name of used comparator
filterstring2	a string used to filter the points
casesensitive	true if the string was used as a casesensitive, false otherwise
wholeword	true if whole words only are compared, false otherwise

16.96.3.9 summaryPeaksTableRowDoubleClicked

```
void cSummaryPeaksTableWidget::summaryPeaksTableRowDoubleClicked (
            int rowid,
            double experimentalmz ) [signal]
```

A row was double clicked.

Parameters

rowid	id of a row in the main window
experimentalmz	experimental mz of a peak

16.96.3.10 updateFilterBySelectedRegion()

```
void cSummaryPeaksTableWidget::updateFilterBySelectedRegion (
            int xmin,
            int xmax,
            int ymin,
            int ymax )
```

Update the filter using an image region.

Parameters

xmin	minimum x coordinate
xmax	maximum x coordinate
ymin	minimum y coordinate
ymax	maximum y coordinate

The documentation for this class was generated from the following files:

- gui/cSummaryPeaksTableWidget.h
- gui/cSummaryPeaksTableWidget.cpp

16.97 cSummaryTableKeyMS Struct Reference

Key to reduce the summary table when exporting statistics for MS and MSI data.

`#include <cSummaryPeaksTableWidget.h>`

Public Member Functions

- void **clear** ()

  Clear the structure.

Public Attributes

- string **id**

  Identification ID subkey.
- string **iontype**

  Ion type subkey.
- string **theoreticalmz**

  Theoretical m/z subkey.
- string **summaryformula**

  Summary formula subkey.
- string **name**

  Compound name subkey.
- string **reference**

  Reference subkey.
- string **score**

  Score subkey.
- string **fdr**

  FDR subkey.

16.97.1 Detailed Description

Key to reduce the summary table when exporting statistics for MS and MSI data.
The documentation for this struct was generated from the following file:

- gui/cSummaryPeaksTableWidget.h

16.98 cSummaryTableKeyMSMS Struct Reference

Key to reduce the summary table when exporting statistics for MS/MS data.
`#include <cSummaryPeaksTableWidget.h>`

Public Member Functions

- void **clear** ()

  Clear the structure.

Public Attributes

- string **theoreticalmz**

  Theoretical m/z subkey.
- string **thint**

  Theoretical intensity subkey.
- string **experimentalmz**

  Experimental m/z subkey.
- string **relint**

  Relative intensity subkey.
- string **absint**

  Absolute intensity subkey.
- string **ppmerror**

  ppm error subkey.
- string **summary**

  summary formula subkey.

16.98.1 Detailed Description

Key to reduce the summary table when exporting statistics for MS/MS data.
The documentation for this struct was generated from the following file:

- gui/cSummaryPeaksTableWidget.h

16.99 cSummaryTableKeyMSMS_comp Struct Reference

Comparison object for cSummaryTableKeyMSMS.
`#include <cSummaryPeaksTableWidget.h>`

Public Member Functions

- bool operator() (const cSummaryTableKeyMSMS &left, const cSummaryTableKeyMSMS &right) const

  Operator to define the order of two items.

16.99.1 Detailed Description

Comparison object for cSummaryTableKeyMSMS.

16.99.2 Member Function Documentation

16.99.2.1 operator()()

```
bool cSummaryTableKeyMSMS_comp::operator() (
            const cSummaryTableKeyMSMS & left,
            const cSummaryTableKeyMSMS & right ) const  [inline]
```

Operator to define the order of two items.

Parameters

left	first key
right	second key

Return values

bool	true if the first object is less than the second object

The documentation for this struct was generated from the following file:

- gui/cSummaryPeaksTableWidget.h

16.100 cTablePosition Struct Reference

Position in QTableWidget.
`#include <cSpectrumDetailWidget.h>`

Public Attributes

- int **row**

  A number of a row.
- int **column**

  A number of a column.

16.100.1 Detailed Description

Position in QTableWidget.
The documentation for this struct was generated from the following file:

- gui/cSpectrumDetailWidget.h

16.101 cTheoreticalSpectrum Class Reference

The class representing a theoretical mass spectrum.
`#include <cTheoreticalSpectrum.h>`

Public Member Functions

- **cTheoreticalSpectrum** ()

  The constructor.
- cTheoreticalSpectrum (cParameters *parameters, cCandidate &candidate)

  The constructor.
- void clear (bool clearpeaklist=true)

  Clear the spectrum.
- cCandidate & getCandidate ()

  Get the peptide spectrum candidate.
- void setCandidate (cCandidate &candidate)

  Set a peptide spectrum candidate.
- int getNumberOfPeaks ()

  Get the number of peaks in the spectrum.
- int compareBranched (cPeaksList &sortedpeaklist, cBricksDatabase &bricksdatabasewithcombinations, bool writedescription, regex &sequencetag, regex &searchedsequence, cPeaksList &unmatchedpeaksinmatchedpatterns, unordered_map< string, int > *isotopeformuladesctoid)

  Compare the theoretical spectrum of a branched peptide with an experimental spectrum.
- int compareLinear (cPeaksList &sortedpeaklist, cBricksDatabase &bricksdatabasewithcombinations, bool writedescription, regex &sequencetag, regex &searchedsequence, cPeaksList &unmatchedpeaksinmatchedpatterns, unordered_map< string, int > *isotopeformuladesctoid)

  Compare the theoretical spectrum of a linear peptide with an experimental spectrum.
- int compareCyclic (cPeaksList &sortedpeaklist, cBricksDatabase &bricksdatabasewithcombinations, bool writedescription, regex &sequencetag, regex &searchedsequence, cPeaksList &unmatchedpeaksinmatchedpatterns, unordered_map< string, int > *isotopeformuladesctoid)

  Compare the theoretical spectrum of a cyclic peptide with an experimental spectrum.
- int compareBranchCyclic (cPeaksList &sortedpeaklist, cBricksDatabase &bricksdatabasewithcombinations, bool writedescription, regex &sequencetag, regex &searchedsequence, cPeaksList &unmatchedpeaksinmatchedpatterns, unordered_map< string, int > *isotopeformuladesctoid)

  Compare the theoretical spectrum of a branch-cyclic peptide with an experimental spectrum.
- int compareLinearPolyketide (cPeaksList &sortedpeaklist, cBricksDatabase &bricksdatabasewithcombinations, bool writedescription, regex &sequencetag, regex &searchedsequence, cPeaksList &unmatchedpeaksinmatchedpatterns, unordered_map< string, int > *isotopeformuladesctoid)

  Compare the theoretical spectrum of a linear polyketide with an experimental spectrum.
- int compareCyclicPolyketide (cPeaksList &sortedpeaklist, cBricksDatabase &bricksdatabasewithcombinations, bool writedescription, regex &sequencetag, regex &searchedsequence, cPeaksList &unmatchedpeaksinmatchedpatterns, unordered_map< string, int > *isotopeformuladesctoid)

  Compare the theoretical spectrum of a cyclic polyketide with an experimental spectrum.
- int compareOther (cPeaksList &sortedpeaklist, bool writedescription, cPeaksList &unmatchedpeaksinmatchedpatterns, unordered_map< string, int > *isotopeformuladesctoid)

  Compare the theoretical spectrum of a metabolite with an experimental spectrum.
- void generateMSSpectrum (int sequencestart, int sequencestop, bool &terminatecomputation, bool writedescription)

Generate a simple mass spectrum.

- void generateFineMSSpectrum (int sequencestart, int sequencestop, bool &terminatecomputation)

  Generate a simple mass spectrum with fine isotopic patterns.
- void getHintsIndex (int id, int peaklistseriesvectorid, cTheoreticalSpectrum &tsfull, cPeaksList &unmatchedpeaksinmatchedpatterns, vector< vector< int > > &hintsindex, bool lcmsrt, bool skipcomparison)

  Get a map of search hints to identify features in LC-MS data.
- void compareMSSpectrum (int id, int peaklistseriesvectorid, cTheoreticalSpectrum &tsfull, cPeaksList &unmatchedpeaksinmatchedpatterns, vector< vector< int > > &hintsindex, bool lcmsrt, bool skipcomparison)

  Compare theoretical peaks with an experimental spectrum.
- void compareAverageMSSpectrum (cPeaksList &averagespectrum, cTheoreticalSpectrum &tsfull, cPeaksList &unmatchedpeaksinmatchedpatterns)

  Compare theoretical peaks with an average spectrum.
- void finalizeMSSpectrum (cPeaksList &unmatchedpeaksinmatchedpatterns, bool writedescription)

  Finalize MS spectrum after comparison.
- int getNumberOfMatchedPeaks () const

  Get the number of matched peaks between an experimental and a theoretical spectrum.
- int getNumberOfScrambledPeaks ()

  Get the number of matched scrambled peaks between an experimental and a theoretical spectrum.
- int getNumberOfMatchedPeaks (eFragmentIonType iontype, int neutrallosstype) const

  Get the number of matched peaks between an experimental and a theoretical spectrum of a specified fragment ion type.
- int getNumberOfMatchedPeaksB () const

  Get the number of matched B ions between an experimental and a theoretical spectrum.
- int getNumberOfMatchedPeaksY () const

  Get the number of matched Y ions between an experimental and a theoretical spectrum.
- int getNumberOfMatchedPeaksYB () const

  Get the number of matched Y and B ions between an experimental and a theoretical spectrum.
- double getRatioOfMatchedPeaks ()

  Get a ratio of matched peaks.
- double getWeightedRatioOfMatchedPeaks () const

  Get weighted ratio of matched peaks.
- double getCosineSimilarity () const

  Get cosine similarity.
- void generateNTerminalFragmentIons (int maxcharge, int &peaklistrealsize, vector< int > &intcomposition, eFragmentIonType fragmentiontype, int neutrallosstype, cBricksDatabase &bricksdatabase, bool writedescription, int rotationid, vector< splitSite > &splittingsites, vector< cFragmentIonType > &searchedmodifications, ePeptideType peptidetype, bool regularblocksorder, TRotationInfo *trotation=0, eResidueLossType leftresiduelosstype=h2o_loss, bool hasfirstblockartificial=false)

  Generate a N-terminal fragment ion series.
- void generateCTerminalFragmentIons (int maxcharge, int &peaklistrealsize, vector< int > &intcomposition, eFragmentIonType fragmentiontype, int neutrallosstype, cBricksDatabase &bricksdatabase, bool writedescription, int rotationid, vector< splitSite > &splittingsites, vector< cFragmentIonType > &searchedmodifications, ePeptideType peptidetype, bool regularblocksorder, TRotationInfo *trotation=0, eResidueLossType rightresiduelosstype=h2o_loss, bool haslastblockartificial=false)

  Generate a C-terminal fragment ion series.
- void sortByMass (int limit=-1)

  Sort peaks by mass.
- void resizePeakList (int size)

  Resize the list of peaks.
- double getSumOfRelativeIntensities () const

  Get the sum of relative intensities of matched peaks.
- cPeak & operator[ ] (int position)

Overloaded operator [].

- cPeaksList & getExperimentalSpectrum ()

  Get an experimental spectrum which has been compared with the theoretical spectrum.
- void setExperimentalSpectrum (cPeaksList &experimentalspectrum)

  Set the experimental spectrum.
- set< int > & getExperimentalMatches (int peakid)

  Get ids of theoretical peaks which hit an experimental peak.
- string getCoverageBySeries ()

  Get coverage of series of fragment ions.
- int getUnmatchedExperimentalPeaksCount ()

  Get the number of unmatched experimental peaks.
- cPeaksList * getTheoreticalPeaks ()

  Get a list of theoretical peaks.
- bool isValid ()

  Check if the candidate sequence corresponds to the searched sequence.
- int getValidPosition ()

  Get a position of a rotation of a cyclic peptide sequence corresponding to searched sequence.
- int getReverseValidPosition ()

  Get a reverted position of a rotation of a cyclic peptide sequence corresponding to searched sequence.
- void setValidSequence (regex &searchedsequence)

  Set a flag determining if the candidate sequence corresponds to the searched sequence.
- vector< string > & getLabelsOfRotations ()

  Get the labels of cyclic rotations.
- int getNumberOfCompletedSeries ()

  Get a number of series which identify the peptide.
- void setNumberOfCompletedSeries (int numberofcompletedseries)

  Set a number of series which identify the peptide.
- void setPathId (int pathid)

  Set path id.
- int getPathId () const

  Get path id.
- void setParameters (cParameters *parameters)

  Set a pointer to the parameters of the application.
- double getTargetPatternScore (int groupid)

  Get target pattern score.
- double getTargetPatternFDR (int groupid)

  Get target pattern FDR.
- map< int, double > & getTargetScores ()

  Get map of target scores.
- map< int, double > & getDecoyScores ()

  Get map of decoy scores.
- void setFDRs (vector< double > &targetscoresvector, vector< double > &fdrs, cPeaksList &unmatchedpeaksinmatchedpatterns)

  Set false discovery rates to peak groups.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is, int fileversionpart1, int fileversionpart2, int fileversionpart3)

  Load the structure from an input stream.
- bool equals (cTheoreticalSpectrum &secondtheoreticalspectrum)

  Check if the object equals to another object.
- void attach (cTheoreticalSpectrum &secondtheoreticalspectrum)

  Attach another spectrum.

16.101.1 Detailed Description

The class representing a theoretical mass spectrum.

16.101.2 Constructor & Destructor Documentation

16.101.2.1 cTheoreticalSpectrum()

```
cTheoreticalSpectrum::cTheoreticalSpectrum (
            cParameters * parameters,
            cCandidate & candidate )
```

The constructor.

Parameters

parameters	a pointer to the parameters of the application
candidate	reference to a peptide sequence candidate

16.101.3 Member Function Documentation

16.101.3.1 attach()

```
void cTheoreticalSpectrum::attach (
            cTheoreticalSpectrum & secondtheoreticalspectrum )
```

Attach another spectrum.

Parameters

secondtheoreticalspectrum	spectrum to be attached

16.101.3.2 clear()

```
void cTheoreticalSpectrum::clear (
            bool clearpeaklist = true )
```

Clear the spectrum.

Parameters

clearpeaklist	if true then all variables including the list of theoretical peaks are cleared else the list of theoretical peaks is not cleared

16.101.3.3 compareAverageMSSpectrum()

```
void cTheoreticalSpectrum::compareAverageMSSpectrum (
            cPeaksList & averagespectrum,
            cTheoreticalSpectrum & tsfull,
            cPeaksList & unmatchedpeaksinmatchedpatterns )
```

Compare theoretical peaks with an average spectrum.

Parameters

averagespectrum	average experimental spectrum
tsfull	theoretical spectrum with descriptions of peaks
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns

16.101.3.4 compareBranchCyclic()

```
int cTheoreticalSpectrum::compareBranchCyclic (
            cPeaksList & sortedpeaklist,
            cBricksDatabase & bricksdatabasewithcombinations,
            bool writedescription,
            regex & sequencetag,
            regex & searchedsequence,
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            unordered_map< string, int > * isotopeformuladesctoid )
```

Compare the theoretical spectrum of a branch-cyclic peptide with an experimental spectrum.

Parameters

sortedpeaklist	reference to an experimental peaklist
bricksdatabasewithcombinations	reference to a database of bricks with combinations of bricks
writedescription	if true then string descriptions of peaks are filled
sequencetag	reference to a regex of a sequence tag
searchedsequence	reference to a regex of a searched sequence
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
isotopeformuladesctoid	a map of isotope descriptions

Return values

int	number theoretical peaks generated; -2 when the sequence tag does not match the peptide sequence candidate

16.101.3.5 compareBranched()

```
int cTheoreticalSpectrum::compareBranched (
            cPeaksList & sortedpeaklist,
            cBricksDatabase & bricksdatabasewithcombinations,
            bool writedescription,
            regex & sequencetag,
            regex & searchedsequence,
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            unordered_map< string, int > * isotopeformuladesctoid )
```

Compare the theoretical spectrum of a branched peptide with an experimental spectrum.

Parameters

sortedpeaklist	reference to an experimental peaklist
bricksdatabasewithcombinations	reference to a database of bricks with combinations of bricks
writedescription	if true then string descriptions of peaks are filled
sequencetag	reference to a regex of a sequence tag

Parameters

searchedsequence	reference to a regex of a searched sequence
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
isotopeformuladesctoid	a map of isotope descriptions

Return values

int	number theoretical peaks generated; -2 when the sequence tag does not match the peptide sequence candidate

16.101.3.6 compareCyclic()

```
int cTheoreticalSpectrum::compareCyclic (
            cPeaksList & sortedpeaklist,
            cBricksDatabase & bricksdatabasewithcombinations,
            bool writedescription,
            regex & sequencetag,
            regex & searchedsequence,
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            unordered_map< string, int > * isotopeformuladesctoid )
```

Compare the theoretical spectrum of a cyclic peptide with an experimental spectrum.

Parameters

sortedpeaklist	reference to an experimental peaklist
bricksdatabasewithcombinations	reference to a database of bricks with combinations of bricks
writedescription	if true then string descriptions of peaks are filled
sequencetag	reference to a regex of a sequence tag
searchedsequence	reference to a regex of a searched sequence
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
isotopeformuladesctoid	a map of isotope descriptions

Return values

int	number theoretical peaks generated; -2 when the sequence tag does not match the peptide sequence candidate

16.101.3.7 compareCyclicPolyketide()

```
int cTheoreticalSpectrum::compareCyclicPolyketide (
            cPeaksList & sortedpeaklist,
            cBricksDatabase & bricksdatabasewithcombinations,
            bool writedescription,
            regex & sequencetag,
            regex & searchedsequence,
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            unordered_map< string, int > * isotopeformuladesctoid )
```

Compare the theoretical spectrum of a cyclic polyketide with an experimental spectrum.

Parameters

sortedpeaklist	reference to an experimental peaklist
bricksdatabasewithcombinations	reference to a database of bricks with combinations of bricks
writedescription	if true then string descriptions of peaks are filled
sequencetag	reference to a regex of a sequence tag
searchedsequence	reference to a regex of a searched sequence
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
isotopeformuladesctoid	a map of isotope descriptions

Return values

int	number theoretical peaks generated; -2 when the sequence tag does not match the peptide sequence candidate

16.101.3.8 compareLinear()

```
int cTheoreticalSpectrum::compareLinear (
            cPeaksList & sortedpeaklist,
            cBricksDatabase & bricksdatabasewithcombinations,
            bool writedescription,
            regex & sequencetag,
            regex & searchedsequence,
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            unordered_map< string, int > * isotopeformuladesctoid )
```

Compare the theoretical spectrum of a linear peptide with an experimental spectrum.

Parameters

sortedpeaklist	reference to an experimental peaklist
bricksdatabasewithcombinations	reference to a database of bricks with combinations of bricks
writedescription	if true then string descriptions of peaks are filled
sequencetag	reference to a regex of a sequence tag
searchedsequence	reference to a regex of a searched sequence
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
isotopeformuladesctoid	a map of isotope descriptions

Return values

int	number theoretical peaks generated; -2 when the sequence tag does not match the peptide sequence candidate

16.101.3.9 compareLinearPolyketide()

```
int cTheoreticalSpectrum::compareLinearPolyketide (
            cPeaksList & sortedpeaklist,
            cBricksDatabase & bricksdatabasewithcombinations,
            bool writedescription,
            regex & sequencetag,
            regex & searchedsequence,
```

```
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            unordered_map< string, int > * isotopeformuladesctoid )
```

Compare the theoretical spectrum of a linear polyketide with an experimental spectrum.

Parameters

sortedpeaklist	reference to an experimental peaklist
bricksdatabasewithcombinations	reference to a database of bricks with combinations of bricks
writedescription	if true then string descriptions of peaks are filled
sequencetag	reference to a regex of a sequence tag
searchedsequence	reference to a regex of a searched sequence
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
isotopeformuladesctoid	a map of isotope descriptions

Return values

int	number theoretical peaks generated; -2 when the sequence tag does not match the peptide sequence candidate

16.101.3.10 compareMSSpectrum()

```
void cTheoreticalSpectrum::compareMSSpectrum (
            int id,
            int peaklistseriesvectorid,
            cTheoreticalSpectrum & tsfull,
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            vector< vector< int > > & hintsindex,
            bool lcmsrt,
            bool skipcomparison )
```

Compare theoretical peaks with an experimental spectrum.

Parameters

id	identifier of an experimental spectrum
peaklistseriesvectorid	id of a peaklistseriesvector
tsfull	theoretical spectrum with descriptions of peaks
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
hintsindex	index of experimental spectra for every matched theoretical peak
lcmsrt	true if LC-MS data are processed and retention time is available
skipcomparison	if true, the peak comparison is skipped

16.101.3.11 compareOther()

```
int cTheoreticalSpectrum::compareOther (
            cPeaksList & sortedpeaklist,
            bool writedescription,
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            unordered_map< string, int > * isotopeformuladesctoid )
```

Compare the theoretical spectrum of a metabolite with an experimental spectrum.

Parameters

sortedpeaklist	reference to an experimental peaklist
writedescription	if true then string descriptions of peaks are filled
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
isotopeformuladesctoid	a map of isotope descriptions

Return values

int	number theoretical peaks generated

16.101.3.12 equals()

```
bool cTheoreticalSpectrum::equals (
            cTheoreticalSpectrum & secondtheoreticalspectrum )
```

Check if the object equals to another object.

Parameters

secondtheoreticalspectrum	object for comparison

Return values

bool	true if the objects are equals

16.101.3.13 finalizeMSSpectrum()

```
void cTheoreticalSpectrum::finalizeMSSpectrum (
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            bool writedescription )
```

Finalize MS spectrum after comparison.

Parameters

unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
writedescription	if true then string descriptions of fragment ions are generated

16.101.3.14 generateCTerminalFragmentIons()

```
void cTheoreticalSpectrum::generateCTerminalFragmentIons (
            int maxcharge,
            int & peaklistrealsize,
            vector< int > & intcomposition,
            eFragmentIonType fragmentiontype,
            int neutrallosstype,
            cBricksDatabase & bricksdatabase,
            bool writedescription,
            int rotationid,
            vector< splitSite > & splittingsites,
```

```
            vector< cFragmentIonType > & searchedmodifications,
            ePeptideType peptidetype,
            bool regularblocksorder,
            TRotationInfo * trotation = 0,
            eResidueLossType rightresiduelosstype = h2o_loss,
            bool haslastblockartificial = false )
```

Generate a C-terminal fragment ion series.

Parameters

maxcharge	a charge of precursor ion
peaklistrealsize	real size of the peak list
intcomposition	reference to a vector of ids of bricks as integers
fragmentiontype	fragment ion type to be generated
neutrallosstype	neutral loss type to be generated
bricksdatabase	reference to a database of building blocks
writedescription	if true then string descriptions of fragment ions are generated
rotationid	id of a cyclic peptide
splittingsites	reference to a vector of splitting sites of a cyclic peptide
searchedmodifications	reference to a vector of searched modifications
peptidetype	the type of searched peptide
regularblocksorder	check regular order of ketide blocks
trotation	a pointer to a T-permutation of a branched peptide
rightresiduelosstype	a residue type of the rightmost building block
haslastblockartificial	true when the last block is artificial, false otherwise

16.101.3.15 generateFineMSSpectrum()

```
void cTheoreticalSpectrum::generateFineMSSpectrum (
            int sequencestart,
            int sequencestop,
            bool & terminatecomputation )
```

Generate a simple mass spectrum with fine isotopic patterns.

Parameters

sequencestart	index of the first item to be processed
sequencestop	index of the last item to be processed (excluding the last item)
terminatecomputation	reference to a variable determining that the computation must be stopped

16.101.3.16 generateMSSpectrum()

```
void cTheoreticalSpectrum::generateMSSpectrum (
            int sequencestart,
            int sequencestop,
            bool & terminatecomputation,
            bool writedescription )
```

Generate a simple mass spectrum.

Parameters

sequencestart	index of the first item to be processed
sequencestop	index of the last item to be processed (excluding the last item)
terminatecomputation	reference to a variable determining that the computation must be stopped
writedescription	if true then string descriptions of peaks are filled

16.101.3.17 generateNTerminalFragmentIons()

```
void cTheoreticalSpectrum::generateNTerminalFragmentIons (
            int maxcharge,
            int & peaklistrealsize,
            vector< int > & intcomposition,
            eFragmentIonType fragmentiontype,
            int neutrallosstype,
            cBricksDatabase & bricksdatabase,
            bool writedescription,
            int rotationid,
            vector< splitSite > & splittingsites,
            vector< cFragmentIonType > & searchedmodifications,
            ePeptideType peptidetype,
            bool regularblocksorder,
            TRotationInfo * trotation = 0,
            eResidueLossType leftresiduelosstype = h2o_loss,
            bool hasfirstblockartificial = false )
```

Generate a N-terminal fragment ion series.

Parameters

maxcharge	a charge of precursor ion
peaklistrealsize	real size of the peak list
intcomposition	reference to a vector of ids of bricks as integers
fragmentiontype	fragment ion type to be generated
neutrallosstype	neutral loss type to be generated
bricksdatabase	reference to a database of building blocks
writedescription	if true then string descriptions of fragment ions are generated
rotationid	id of a cyclic peptide
splittingsites	reference to a vector of splitting sites of a cyclic peptide
searchedmodifications	reference to a vector of searched modifications
peptidetype	the type of searched peptide
regularblocksorder	check regular order of ketide blocks
trotation	a pointer to a T-permutation of a branched peptide
leftresiduelosstype	a residue type of the leftmost building block
hasfirstblockartificial	true when the first block is artificial, false otherwise

16.101.3.18 getCandidate()

```
cCandidate & cTheoreticalSpectrum::getCandidate ( )
```

Get the peptide spectrum candidate.

Return values

cCandidate	reference to the peptide sequence candidate

16.101.3.19 getCosineSimilarity()

```
double cTheoreticalSpectrum::getCosineSimilarity ( ) const
```

Get cosine similarity.

Return values

double	cosine similarity

16.101.3.20 getCoverageBySeries()

```
string cTheoreticalSpectrum::getCoverageBySeries ( )
```

Get coverage of series of fragment ions.

Return values

string	coverage of the series of fragment ions

16.101.3.21 getDecoyScores()

```
map< int, double > & cTheoreticalSpectrum::getDecoyScores ( )
```

Get map of decoy scores.

Return values

map<int,double>	map of decoy scores

16.101.3.22 getExperimentalMatches()

```
set< int > & cTheoreticalSpectrum::getExperimentalMatches (
            int peakid )
```

Get ids of theoretical peaks which hit an experimental peak.

Return values

set<int>	set of ids of theoretical peaks

16.101.3.23 getExperimentalSpectrum()

```
cPeaksList & cTheoreticalSpectrum::getExperimentalSpectrum ( )
```

Get an experimental spectrum which has been compared with the theoretical spectrum.

Return values

cPeaksList	reference to an experimental spectrum

16.101.3.24 getHintsIndex()

```
void cTheoreticalSpectrum::getHintsIndex (
            int id,
            int peaklistseriesvectorid,
            cTheoreticalSpectrum & tsfull,
            cPeaksList & unmatchedpeaksinmatchedpatterns,
            vector< vector< int > > & hintsindex,
            bool lcmsrt,
            bool skipcomparison )
```

Get a map of search hints to identify features in LC-MS data.

Parameters

id	identifier of an experimental spectrum
peaklistseriesvectorid	id of a peaklistseriesvector
tsfull	theoretical spectrum
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns
hintsindex	index of experimental spectra for every matched theoretical peak
lcmsrt	true if LC-MS data are processed and retention time is available
skipcomparison	if true, the peak comparison is skipped

16.101.3.25 getLabelsOfRotations()

```
vector< string > & cTheoreticalSpectrum::getLabelsOfRotations ( )
```

Get the labels of cyclic rotations.

Return values

vector<string>	a vector of labels

16.101.3.26 getNumberOfCompletedSeries()

```
int cTheoreticalSpectrum::getNumberOfCompletedSeries ( )
```

Get a number of series which identify the peptide.

Return values

int	number of completed series

16.101.3.27 getNumberOfMatchedPeaks() [1/2]

```
int cTheoreticalSpectrum::getNumberOfMatchedPeaks ( ) const
```

Get the number of matched peaks between an experimental and a theoretical spectrum.

Return values

int	number of matched peaks

16.101.3.28 getNumberOfMatchedPeaks() [2/2]

```
int cTheoreticalSpectrum::getNumberOfMatchedPeaks (
            eFragmentIonType iontype,
            int neutrallosstype ) const
```

Get the number of matched peaks between an experimental and a theoretical spectrum of a specified fragment ion type.

Parameters

iontype	a fragment ion type
neutrallosstype	a neutral loss type

Return values

int	number of matched peaks

16.101.3.29 getNumberOfMatchedPeaksB()

```
int cTheoreticalSpectrum::getNumberOfMatchedPeaksB ( ) const
```

Get the number of matched B ions between an experimental and a theoretical spectrum.

Return values

int	number of matched peaks

16.101.3.30 getNumberOfMatchedPeaksY()

```
int cTheoreticalSpectrum::getNumberOfMatchedPeaksY ( ) const
```

Get the number of matched Y ions between an experimental and a theoretical spectrum.

Return values

int	number of matched peaks

16.101.3.31 getNumberOfMatchedPeaksYB()

```
int cTheoreticalSpectrum::getNumberOfMatchedPeaksYB ( ) const
```

Get the number of matched Y and B ions between an experimental and a theoretical spectrum.

Return values

int	number of matched peaks

16.101.3.32 getNumberOfPeaks()

`int cTheoreticalSpectrum::getNumberOfPeaks ( )`
Get the number of peaks in the spectrum.

Return values

int	the number of peaks

16.101.3.33 getNumberOfScrambledPeaks()

`int cTheoreticalSpectrum::getNumberOfScrambledPeaks ( )`
Get the number of matched scrambled peaks between an experimental and a theoretical spectrum.

Return values

int	number of matched scrambled peaks

16.101.3.34 getPathId()

`int cTheoreticalSpectrum::getPathId ( ) const`
Get path id.

Return values

int	id of a path

16.101.3.35 getRatioOfMatchedPeaks()

`double cTheoreticalSpectrum::getRatioOfMatchedPeaks ( )`
Get a ratio of matched peaks.

Return values

double	ratio of matched peaks

16.101.3.36 getReverseValidPosition()

`int cTheoreticalSpectrum::getReverseValidPosition ( )`
Get a reverted position of a rotation of a cyclic peptide sequence corresponding to searched sequence.

Return values

int	reverted position of sequence

16.101.3.37 getSumOfRelativeIntensities()

```
double cTheoreticalSpectrum::getSumOfRelativeIntensities ( ) const
```

Get the sum of relative intensities of matched peaks.

Return values

double	sum of relative intensities of matched peaks

16.101.3.38 getTargetPatternFDR()

```
double cTheoreticalSpectrum::getTargetPatternFDR (
            int groupid )
```

Get target pattern FDR.

Parameters

groupid	id of a group of peaks forming a target pattern

Return values

double	target pattern FDR

16.101.3.39 getTargetPatternScore()

```
double cTheoreticalSpectrum::getTargetPatternScore (
            int groupid )
```

Get target pattern score.

Parameters

groupid	id of a group of peaks forming a target pattern

Return values

double	target pattern score

16.101.3.40 getTargetScores()

```
map< int, double > & cTheoreticalSpectrum::getTargetScores ( )
```

Get map of target scores.

Return values

map<int,double>	map of target scores

16.101.3.41 getTheoreticalPeaks()

```
cPeaksList * cTheoreticalSpectrum::getTheoreticalPeaks ( )
```

Get a list of theoretical peaks.

Return values

cPeaksList	a pointer to a list of theoretical peaks

16.101.3.42 getUnmatchedExperimentalPeaksCount()

```
int cTheoreticalSpectrum::getUnmatchedExperimentalPeaksCount ( )
```

Get the number of unmatched experimental peaks.

Return values

number	of unmatched experimental peaks

16.101.3.43 getValidPosition()

```
int cTheoreticalSpectrum::getValidPosition ( )
```

Get a position of a rotation of a cyclic peptide sequence corresponding to searched sequence.

Return values

int	position of sequence

16.101.3.44 getWeightedRatioOfMatchedPeaks()

```
double cTheoreticalSpectrum::getWeightedRatioOfMatchedPeaks ( ) const
```

Get weighted ratio of matched peaks.

Return values

double	weighted ratio of matched peaks

16.101.3.45 isValid()

```
bool cTheoreticalSpectrum::isValid ( )
```

Check if the candidate sequence corresponds to the searched sequence.

Return values

bool	if true then the theoretical spectrum corresponds to a searched sequence

16.101.3.46 load()

```
void cTheoreticalSpectrum::load (
```

```
            ifstream & is,
            int fileversionpart1,
            int fileversionpart2,
            int fileversionpart3 )
```

Load the structure from an input stream.

Parameters

is	an input stream
fileversionpart1	first number of .res the file version
fileversionpart2	second number of .res the file version
fileversionpart3	third number of .res the file version

16.101.3.47 operator[]()

```
cPeak & cTheoreticalSpectrum::operator[] (
            int position )
```

Overloaded operator [].

Parameters

position	position of a peak in the peak list

Return values

cPeak	reference to the peak

16.101.3.48 resizePeakList()

```
void cTheoreticalSpectrum::resizePeakList (
            int size )
```

Resize the list of peaks.

Parameters

size	the number of peaks

16.101.3.49 setCandidate()

```
void cTheoreticalSpectrum::setCandidate (
            cCandidate & candidate )
```

Set a peptide spectrum candidate.

Parameters

candidate	reference to the peptide sequence candidate

16.101.3.50 setExperimentalSpectrum()

```
void cTheoreticalSpectrum::setExperimentalSpectrum (
            cPeaksList & experimentalspectrum )
```

Set the experimental spectrum.

Parameters

experimentalspectrum	an experimental spectrum

16.101.3.51 setFDRs()

```
void cTheoreticalSpectrum::setFDRs (
            vector< double > & targetscoresvector,
            vector< double > & fdrs,
            cPeaksList & unmatchedpeaksinmatchedpatterns )
```

Set false discovery rates to peak groups.

Parameters

targetscoresvector	vector of target scores
fdrs	vector of FDRs
unmatchedpeaksinmatchedpatterns	unmatched peaks in matched isotope patterns

16.101.3.52 setNumberOfCompletedSeries()

```
void cTheoreticalSpectrum::setNumberOfCompletedSeries (
            int numberofcompletedseries )
```

Set a number of series which identify the peptide.

Parameters

numberofcompletedseries	number of completed series

16.101.3.53 setParameters()

```
void cTheoreticalSpectrum::setParameters (
            cParameters * parameters )
```

Set a pointer to the parameters of the application.

Parameters

parameters	the pointer to the parameters of the application

16.101.3.54 setPathId()

```
void cTheoreticalSpectrum::setPathId (
            int pathid )
```

Set path id.

Parameters

pathid	id of a path

16.101.3.55 setValidSequence()

```
void cTheoreticalSpectrum::setValidSequence (
            regex & searchedsequence )
```

Set a flag determining if the candidate sequence corresponds to the searched sequence.

Parameters

searchedsequence	searched sequence

16.101.3.56 sortByMass()

```
void cTheoreticalSpectrum::sortByMass (
            int limit = -1 )
```

Sort peaks by mass.

Parameters

limit	sort first *limit* peaks only; if *limit* == -1 then all peaks are sorted

16.101.3.57 store()

```
void cTheoreticalSpectrum::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this class was generated from the following files:

- core/cTheoreticalSpectrum.h
- core/cTheoreticalSpectrum.cpp

16.102 cTheoreticalSpectrumList Class Reference

The class representing lists of theoretical spectra.

```
#include <cTheoreticalSpectrumList.h>
```

Public Member Functions

- **cTheoreticalSpectrumList** ()

  The constructor.
- void **clear** ()

  Clear the class.

- void clear (int fileid)

  Clear a specified file id.
- void initialize (cMainThread &os, cParameters &parameters, cDeNovoGraph *graph)

  Initialize the list.
- void add (int fileid, cTheoreticalSpectrum &theoreticalspectrum)

  Add a spectrum to the list.
- cTheoreticalSpectrum & get (int fileid, int order)

  Get a reference to a spectrum.
- int size ()

  Get the number of files.
- int size (int fileid)

  Get the number of spectra in a file.
- void resize (int size)

  Resize the vector of files.
- void resize (int fileid, int size)

  Resize a file id vector.
- int parallelCompareAndStore (int fileid, cCandidateSet &candidates, bool &terminatecomputation)

  Parallel comparison of theoretical spectra with an experimental spectrum.
- double updatekNNList (cTheoreticalSpectrum &theoreticalspectrum, int theoreticalpeaksrealsize, double score, regex *rxsearchedsequence)

  Update the kNN list of peptide sequence candidates (thread-safe).
- void sortAndFitSize (int fileid)

  Sort the list according to a score defined by cParameters::scoretype and crop the list according cParameters::hitsreported.

16.102.1 Detailed Description

The class representing lists of theoretical spectra.

16.102.2 Member Function Documentation

16.102.2.1 add()

```
void cTheoreticalSpectrumList::add (
            int fileid,
            cTheoreticalSpectrum & theoreticalspectrum )
```

Add a spectrum to the list.

Parameters

fileid	file id
theoreticalspectrum	reference to a spectrum

16.102.2.2 clear()

```
void cTheoreticalSpectrumList::clear (
            int fileid )
```

Clear a specified file id.

Parameters

fileid	file id

16.102.2.3 get()

```
cTheoreticalSpectrum & cTheoreticalSpectrumList::get (
            int fileid,
            int order )
```

Get a reference to a spectrum.

Parameters

fileid	file id
order	id of a spectrum

Return values

cTheoreticalSpectrum	reference to a spectrum

16.102.2.4 initialize()

```
void cTheoreticalSpectrumList::initialize (
            cMainThread & os,
            cParameters & parameters,
            cDeNovoGraph * graph )
```

Initialize the list.

Parameters

os	reference to the main thread of the application
parameters	reference to the parameters of the application
graph	pointer to the de novo graph

16.102.2.5 parallelCompareAndStore()

```
int cTheoreticalSpectrumList::parallelCompareAndStore (
            int fileid,
            cCandidateSet & candidates,
            bool & terminatecomputation )
```

Parallel comparison of theoretical spectra with an experimental spectrum.

Parameters

fileid	file id
candidates	reference to a set of peptide sequence candidates
terminatecomputation	reference to a variable determining that the computation must be stopped

Return values

int	0 when successfully finished; -1 when an error occurred

16.102.2.6 resize() [1/2]

```
void cTheoreticalSpectrumList::resize (
            int fileid,
            int size )
```

Resize a file id vector.

Parameters

fileid	file id
size	size

16.102.2.7 resize() [2/2]

```
void cTheoreticalSpectrumList::resize (
            int size )
```

Resize the vector of files.

Parameters

size	size

16.102.2.8 size() [1/2]

```
int cTheoreticalSpectrumList::size ( )
```

Get the number of files.

Return values

int	number of files

16.102.2.9 size() [2/2]

```
int cTheoreticalSpectrumList::size (
            int fileid )
```

Get the number of spectra in a file.

Parameters

fileid	file id

Return values

int	number of spectra

16.102.2.10 sortAndFitSize()

```
void cTheoreticalSpectrumList::sortAndFitSize (
            int fileid )
```

Sort the list according to a score defined by cParameters::scoretype and crop the list according cParameters::hitsreported.

Parameters

fileid	file id

16.102.2.11 updatekNNList()

```
double cTheoreticalSpectrumList::updatekNNList (
            cTheoreticalSpectrum & theoreticalspectrum,
            int theoreticalpeaksrealsize,
            double score,
            regex * rxsearchedsequence )
```

Update the kNN list of peptide sequence candidates (thread-safe).

Parameters

theoreticalspectrum	reference to the newly added spectrum
theoreticalpeaksrealsize	the number of peaks in the newly added spectrum (required because of a performance improvement)
score	score of the candidate to be inserted
rxsearchedsequence	regular expression corresponding to searched sequence

Return values

double	current value of the worst score in the list of candidates

The documentation for this class was generated from the following files:

- core/cTheoreticalSpectrumList.h
- core/cTheoreticalSpectrumList.cpp

16.103 cTileInfo Struct Reference

Description of an image tile.

`#include <cImageWindowTabOptical.h>`

Public Member Functions

- **cTileInfo** ()

  The constructor.
- void **print** ()

  Print the metadata.

Public Attributes

- int **fieldx**

  Tile logical x coordinate.
- int **fieldy**

  Tile logical y coordinate.
- double **posx**

  Tile position x [m].
- double **posy**

  Tile position y [m].

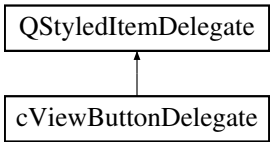
16.103.1 Detailed Description

Description of an image tile.
The documentation for this struct was generated from the following file:

- gui/cImageWindowTabOptical.h

16.104 cViewButtonDelegate Class Reference

A delegate to create a reference view button.
`#include <cViewButtonDelegate.h>`
Inheritance diagram for cViewButtonDelegate:



Public Member Functions

- cViewButtonDelegate (QObject ∗parent=(QObject ∗) 0)

  The constructor.
- QString displayText (const QVariant &value, const QLocale &locale) const

  Reimplementation of displayText.
- void paint (QPainter ∗painter, const QStyleOptionViewItem &option, const QModelIndex &index) const

  Reimplementation of paint.

Protected Member Functions

- bool editorEvent (QEvent ∗event, QAbstractItemModel ∗model, const QStyleOptionViewItem &option, const QModelIndex &index)

  Reimplementation of editorEvent.

16.104.1 Detailed Description

A delegate to create a reference view button.

16.104.2 Constructor & Destructor Documentation

16.104.2.1 cViewButtonDelegate()

```
cViewButtonDelegate::cViewButtonDelegate (
            QObject * parent = (QObject *)0 )
```

The constructor.

Parameters

parent	pointer to a parent widget

16.104.3 Member Function Documentation

16.104.3.1 displayText()

```
QString cViewButtonDelegate::displayText (
            const QVariant & value,
            const QLocale & locale ) const
```

Reimplementation of displayText.

Parameters

value	a value
locale	a locale

Return values

QString	the value as a string

16.104.3.2 editorEvent()

```
bool cViewButtonDelegate::editorEvent (
            QEvent * event,
            QAbstractItemModel * model,
            const QStyleOptionViewItem & option,
            const QModelIndex & index )  [protected]
```

Reimplementation of editorEvent.

Parameters

event	QEvent
model	QAbstractItemModel
option	QStyleOptionViewItem
index	QModelIndex

16.104.3.3 paint()

```
void cViewButtonDelegate::paint (
            QPainter * painter,
            const QStyleOptionViewItem & option,
            const QModelIndex & index ) const
```

Reimplementation of paint.

Parameters

painter	QPainter
option	QStyleOptionViewItem
index	QModelIndex

The documentation for this class was generated from the following files:

- gui/cViewButtonDelegate.h
- gui/cViewButtonDelegate.cpp

16.105 hash_cCandidate Struct Reference

The structure defining a hash function of cCandidate.
`#include <cCandidate.h>`

Public Member Functions

- size_t operator() (const cCandidate &candidate) const

  Hash a peptide sequence candidate.

16.105.1 Detailed Description

The structure defining a hash function of cCandidate.

16.105.2 Member Function Documentation

16.105.2.1 operator()()

```
size_t hash_cCandidate::operator() (
            const cCandidate & candidate ) const  [inline]
```

Hash a peptide sequence candidate.

Parameters

candidate	reference to a peptide sequence candidate

Return values

size_t	hashed candidate

The documentation for this struct was generated from the following file:

- core/cCandidate.h

16.106 hash_cEdge Struct Reference

The structure defining a hash function of cEdge.
`#include <cDeNovoGraphNode.h>`

Public Member Functions

- size_t operator() (const cEdge &e) const

  Hash an edge.

16.106.1 Detailed Description

The structure defining a hash function of cEdge.

16.106.2 Member Function Documentation

16.106.2.1 operator()()

```
size_t hash_cEdge::operator() (
            const cEdge & e ) const  [inline]
```

Hash an edge.

Parameters

e	reference to an edge

Return values

size_t	hashed edge

The documentation for this struct was generated from the following file:

- core/cDeNovoGraphNode.h

16.107 hash_cIonLabel Struct Reference

The structure defining a hash function of cIonLabel.
`#include <cLinearWidget.h>`

Public Member Functions

- size_t operator() (const cIonLabel &label) const

  Hash a ion label.

16.107.1 Detailed Description

The structure defining a hash function of cIonLabel.

16.107.2 Member Function Documentation

16.107.2.1 operator()()

```
size_t hash_cIonLabel::operator() (
            const cIonLabel & label ) const  [inline]
```

Hash a ion label.

Parameters

label	reference to a ion label

Return values

size_t	hashed ion label

The documentation for this struct was generated from the following file:

- gui/cLinearWidget.h

16.108 hintStructure Struct Reference

An auxiliary structure for search hints.
`#include <cParameters.h>`

Public Attributes

- double **mz**

  m/z value.
- int **id**

  Spectrum ID.

16.108.1 Detailed Description

An auxiliary structure for search hints.
The documentation for this struct was generated from the following file:

- core/cParameters.h

16.109 isotope Struct Reference

Representation of an isotope.
`#include <cFragmentIons.h>`

Public Member Functions

- **isotope** ()

  The default constructor.
- void **clear** ()

  Clear the structure.

Public Attributes

- double **mass**

  Monoisotopic mass.
- double **probability**

  Probability of natural abundance.

16.109.1 Detailed Description

Representation of an isotope.
The documentation for this struct was generated from the following file:

- core/cFragmentIons.h

16.110 layerInfo Struct Reference

Paramaters of a layer.
`#include <cImageWindowWidget.h>`

Public Member Functions

- **layerInfo** ()

  The constructor.
- **layerInfo** (const layerInfo &layer)

  The copy constructor.
- void **clear** ()

  Clear the structure.
- ∼**layerInfo** ()

  The destructor.

Public Attributes

- bool **checked**

  True if the layer is visible.
- int **alpha**

  Opacity of the layer.
- int **zvalue**

  Z-value of the layer.
- bool **ispixmapdefined**

  True if pixmap is defined.
- bool **fliphorizontally**

  True if the pixmap is flipped horizontally.
- bool **flipvertically**

  True if the pixmap is flipped vertically.
- double **x**

  X position.
- double **y**

  Y position.
- double **width**

  Width.
- double **height**

  Height.
- double **angle**

  Rotation angle [degrees].
- qreal **lastx**

  Last X position.
- qreal **lasty**

  Last Y position.
- int **navigationlayer**

  Navigation layer.
- QPixmap ∗ **pixmap**

  Pixmap.

16.110.1 Detailed Description

Paramaters of a layer.
The documentation for this struct was generated from the following file:

- gui/cImageWindowWidget.h

16.111 matchedSeries Struct Reference

An auxiliary structure for visualisation of matched series of fragment ions.
`#include <cTheoreticalSpectrum.h>`

Public Member Functions

- **matchedSeries** ()

  The constructor.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is)

  Load the structure from an input stream.

Public Attributes

- vector< int > **series**

  A vector of matched peaks in a series.
- string **name**

  A name of a series of fragment ions.

16.111.1 Detailed Description

An auxiliary structure for visualisation of matched series of fragment ions.

16.111.2 Member Function Documentation

16.111.2.1 load()

```
void matchedSeries::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.111.2.2 store()

```
void matchedSeries::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this struct was generated from the following files:

- core/cTheoreticalSpectrum.h
- core/cTheoreticalSpectrum.cpp

16.112 microscopyLayerInfo Struct Reference

Paramaters of a microscopy layer.
`#include <cImageWindow.h>`

Public Member Functions

- **microscopyLayerInfo** ()
  The constructor.
- **microscopyLayerInfo** (const microscopyLayerInfo &layer)
  The copy constructor.
- ~**microscopyLayerInfo** ()
  The destructor.
- void **clear** ()
  Clear the structure.

Public Attributes

- int **type**
  Type (0 = FEI SEM, 1 = LEICA LIF).
- int **navigationlayer**
  Index of navigation layer.
- QString **sourcename**
  Source file name.
- QString **imagename**
  Image name.
- bool **flipx**
  Flip horizontally.
- bool **flipy**
  Flip vertically.
- double **x**
  X position.
- double **y**
  Y position.
- double **width**
  Width.
- double **height**
  Height.
- double **angle**
  Rotation angle [degrees].
- double **defaultwidth**
  Default width.
- double **defaultheight**
  Default height.
- double **stagex**
  Stage X.
- double **stagey**
  Stage Y.
- QImage * **image**
  QImage.
- bool **reduced**
  True if the image size has been reduced.

- int **thresholdmin**

  Noise threshold.
- int **thresholdmax**

  Signal threshold.

16.112.1 Detailed Description

Paramaters of a microscopy layer.
The documentation for this struct was generated from the following file:

- gui/cImageWindow.h

16.113 neutralLoss Struct Reference

The structure for a detailed definition of a neutral loss.
`#include <cFragmentIons.h>`

Public Member Functions

- **neutralLoss** ()

  The default constructor.
- void **clear** ()

  Clear the structure.
- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is)

  Load the structure from an input stream.

Public Attributes

- string **summary**

  Summary formula of the loss.
- map< string, int > **summarymap**

  A map of atoms corresponding to the summary formula.
- double **massdifference**

  Mass difference of the loss.

16.113.1 Detailed Description

The structure for a detailed definition of a neutral loss.

16.113.2 Member Function Documentation

16.113.2.1 load()

```
void neutralLoss::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.113.2.2 store()

```
void neutralLoss::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this struct was generated from the following files:

- core/cFragmentIons.h
- core/cFragmentIons.cpp

16.114 nodeEdge Struct Reference

An auxiliary structure to store relationship between a node and an edge.
`#include <cDeNovoGraphNode.h>`

Public Member Functions

- void store (ofstream &os)

  Store the structure into an output stream.
- void load (ifstream &is)

  Load the structure from an input stream.

Public Attributes

- int **nodeid**

  Id of a node.
- int **edgeid**

  Id of an edge.

16.114.1 Detailed Description

An auxiliary structure to store relationship between a node and an edge.

16.114.2 Member Function Documentation

16.114.2.1 load()

```
void nodeEdge::load (
            ifstream & is )
```

Load the structure from an input stream.

Parameters

is	an input stream

16.114.2.2 store()

```
void nodeEdge::store (
            ofstream & os )
```

Store the structure into an output stream.

Parameters

os	an output stream

The documentation for this struct was generated from the following files:

- core/cDeNovoGraphNode.h
- core/cDeNovoGraphNode.cpp

16.115 splitSite Struct Reference

The structure representing a splitting site of a cyclic peptide.
`#include <cTheoreticalSpectrum.h>`

Public Attributes

- int **first**

  Order of the first amino acid where a cyclic peptide is split.
- int **second**

  Order of the second amino acid where a cyclic peptide is split.

16.115.1 Detailed Description

The structure representing a splitting site of a cyclic peptide.
The documentation for this struct was generated from the following file:

- core/cTheoreticalSpectrum.h

16.116 TRotationInfo Struct Reference

An auxiliary structure representing a branch permutation of a branched peptide.
`#include <cCandidate.h>`

Public Member Functions

- **TRotationInfo** ()

  The constructor.
- void **clear** ()

  Clear the structure.
- bool startsWithBracket ()

  Test whether tcomposition starts with a bracket.
- bool endsWithBracket ()

  Test whether tcomposition ends with a bracket.

Public Attributes

- int **id**

  Id of the rotation.
- vector< int > **bricks**

Vector of ids of bricks.

- int **middlebranchstart**

  Position of a brick where a branch of a branched peptide starts.
- int **middlebranchend**

  Position of a brick where a branch of a branched peptide ends.
- int **startmodifID**

  Id of a terminal modification (at the beginning of a spectrum).
- int **endmodifID**

  Id of a terminal modification (at the end of a spectrum).
- int **middlemodifID**

  Id of a terminal modification of a branched peptide (in the middle of a spectrum).
- string **tcomposition**

  Ids of bricks separated by '-', the branch is specified by '(' and ')'; example: 1-2-3(4-5)6-7.

16.116.1 Detailed Description

An auxiliary structure representing a branch permutation of a branched peptide.

16.116.2 Member Function Documentation

16.116.2.1 endsWithBracket()

```
bool TRotationInfo::endsWithBracket ( )  [inline]
```

Test whether *tcomposition* ends with a bracket.

Return values

bool	true when *tcomposition* ends with a bracket

16.116.2.2 startsWithBracket()

```
bool TRotationInfo::startsWithBracket ( )  [inline]
```

Test whether *tcomposition* starts with a bracket.

Return values

bool	true when *tcomposition* starts with a bracket

The documentation for this struct was generated from the following file:

- core/cCandidate.h

Chapter 17

File Documentation

17.1 core/cBrick.h File Reference

The representation of a brick.

```
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <algorithm>
#include <map>
#include "core/utilities.h"
```

Classes

- class cBrick

  The class representing a building block (brick).

Enumerations

- enum eResidueLossType { **h2o_loss** = 0 , **h2_loss** }

  Residue loss types.

Functions

- int getNumberOfBricks (const string &composition)

  Get the number of bricks in a composition.

17.1.1 Detailed Description

The representation of a brick.

17.1.2 Function Documentation

17.1.2.1 getNumberOfBricks()

```
int getNumberOfBricks (
            const string & composition )
```

Get the number of bricks in a composition.

Parameters

composition	string containing ids of bricks separated by '-'

Return values

number	number of bricks

17.2 cBrick.h

Go to the documentation of this file.

```
1
7 #ifndef _CBRICK_H
8 #define _CBRICK_H
9
10 #include <iostream>
11 #include <fstream>
12 #include <string>
13 #include <vector>
14 #include <algorithm>
15 #include <map>
16 #include "core/utilities.h"
17
18 using namespace std;
19
20
24 enum eResidueLossType {
25     h2o_loss = 0,
26     h2_loss
27 };
28
29
35 int getNumberOfBricks(const string& composition);
36
37
41 class cBrick {
42     string name;
43     vector<string> acronyms;
44     vector<string> references;
45     string summary;
46     map<string, int> summarymap;
47     double mass;
48     string composition;
49     bool artificial;
50     eResidueLossType residuelosstype;
51     string losses;
52     vector<int> lossids;
53
54 public:
55
59     cBrick();
60
61
65     void clear();
66
67
72     bool empty();
73
74
79     string& getName();
80
81
86     void setName(const string& name);
87
88
93     string& getSummary();
94
95
100     map<string, int>& getSummaryMap();
101
102
107     vector<string>& getAcronyms();
108
109
114     string& getLosses();
115
116
121     vector<int>& getLossIDs();
```

```
122
123
128     vector<string>& getReferences();
129
130
135     string& getComposition();
136
137
142     string getReverseComposition();
143
144
149     double getMass() const;
150
151
156     void setMass(double mass);
157
158
164     void setComposition(const string& composition, bool normalize);
165
166
171     void setSummary(const string& summary);
172
173
177     void createSummaryMap();
178
179
184     void setAcronyms(const string& acronyms);
185
186
191     void setLosses(const string& str);
192
193
198     void setLossIDs(vector<int>& lossids);
199
200
205     void setReferences(const string& references);
206
207
211     void normalizeComposition();
212
213
218     void explodeToIntComposition(vector<int>& bricks);
219
220
225     void explodeToStringComposition(vector<string>& bricks);
226
227
232     string getAcronymsAsString();
233
234
239     string getReferencesAsString();
240
241
246     string getFirstAcronymAsString();
247
248
253     string getAcronymsWithReferencesAsHTMLString();
254
255
260     void setArtificial(bool artificial);
261
262
267     bool isArtificial();
268
269
274     void setResidueLossType(eResidueLossType residuelosstype);
275
276
281     eResidueLossType getResidueLossType();
282
283
288     void store(ofstream& os);
289
290
295     void load(ifstream& is);
296
297 };
298
299
300 #endif
301
```

17.3 core/cBricksDatabase.h File Reference

The database of building blocks.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>
#include <iomanip>
#include "core/utilities.h"
#include "core/cBrick.h"
#include "core/cSummaryFormula.h"
```

Classes

- class cBricksDatabase

  The class representing a database of building blocks (bricks).

Functions

- bool compareBrickMasses (const cBrick &a, const cBrick &b)

  Compare masses of two bricks.
- void getNameOfCompositionFromIntVector (string &composition, vector< int > &combarray)

  Convert vector of ids of bricks into a string.
- string & getReversedNameOfCompositionFromStringVector (string &composition, vector< string > &string-composition)

  Convert vector of ids of bricks into a string having reversed ids of bricks.
- void generateBricksPermutations (vector< string > &bricks, vector< string > &currentpermutation, vector< string > &permutations)

  Generate permutations of bricks.

17.3.1 Detailed Description

The database of building blocks.

17.3.2 Function Documentation

17.3.2.1 compareBrickMasses()

```
bool compareBrickMasses (
            const cBrick & a,
            const cBrick & b )
```

Compare masses of two bricks.

Parameters

a	first brick
b	second brick

Return values

bool	true if the mass of *a* is less than the mass of *b*

17.3.2.2 generateBricksPermutations()

```
void generateBricksPermutations (
            vector< string > & bricks,
            vector< string > & currentpermutation,
            vector< string > & permutations )
```

Generate permutations of bricks.

Parameters

bricks	reference to an input vector of strings (ids of bricks)
currentpermutation	reference to an empty vector of strings (auxiliary parameter)
permutations	reference to an empty output vector of strings where the result will be stored

17.3.2.3 getNameOfCompositionFromIntVector()

```
void getNameOfCompositionFromIntVector (
            string & composition,
            vector< int > & combarray )
```

Convert vector of ids of bricks into a string.

Parameters

composition	reference to a string where ids of bricks separated by '-' will be stored
combarray	reference to an input integer vector of ids of bricks

17.3.2.4 getReversedNameOfCompositionFromStringVector()

```
string & getReversedNameOfCompositionFromStringVector (
            string & composition,
            vector< string > & stringcomposition )
```

Convert vector of ids of bricks into a string having reversed ids of bricks.

Parameters

composition	reference to a string where ids of bricks separated by '-' will be stored
stringcomposition	reference to an input string vector of ids of bricks

Return values

string	reference to a string where ids of bricks separated by '-' will be stored

17.4 cBricksDatabase.h

Go to the documentation of this file.

```
1
7 #ifndef _CBRICKSDATABASE_H
8 #define _CBRICKSDATABASE_H
9
```

```
10 #include <iostream>
11 #include <fstream>
12 #include <vector>
13 #include <string>
14 #include <algorithm>
15 #include <iomanip>
16
17 #include "core/utilities.h"
18 #include "core/cBrick.h"
19 #include "core/cSummaryFormula.h"
20
21
22 using namespace std;
23
24
25 class cMainThread;
26
27
34 bool compareBrickMasses(const cBrick& a, const cBrick& b);
35
36
42 void getNameOfCompositionFromIntVector(string& composition, vector<int>& combarray);
43
44
51 string& getReversedNameOfCompositionFromStringVector(string& composition, vector<string>&
       stringcomposition);
52
53
60 void generateBricksPermutations(vector<string>& bricks, vector<string>& currentpermutation,
       vector<string>& permutations);
61
62
66 class cBricksDatabase {
67
68     vector<cBrick> bricks;
69
70     map<string, int> lossorders;
71
72     vector<double> lossmasses;
73     vector<string> losssummaries;
74     vector< map<string, int> > lossmaps;
75
76     void addLossToMap(string& lossstr, vector<int>& lossids);
77
78     double getMassFromCombCounts(vector<int>& combcounts, vector<double>& combmasses);
79
80
81 public:
82
83
93     bool nextCombination(vector<int>& combarray, int numberofbasicbricks, int maximumbricksincombination,
       double maximumcumulativemass, double neutralprecursormass);
94
95
108     bool nextCombinationFast(vector<int>& combarray, vector<int>& combcounts, vector<double>&
       combmasses, double& mass, int numberofbasicbricks, int maximumbricksincombination, double
       maximumcumulativemass, double neutralprecursormass);
109
110
124     bool nextCombinationFastLimited(vector<int>& combarray, vector<int>& combcounts, vector<int>&
       comblimits, vector<double>& combmasses, double& mass, int numberofbasicbricks, int
       maximumbricksincombination, double maximumcumulativemass, double neutralprecursormass);
125
126
130     cBricksDatabase();
131
132
141     int loadFromPlainTextStream(ifstream &stream, string& errormessage, bool ignoreerrors, bool
       skiph2blocks);
142
143
148     void storeToPlainTextStream(ofstream &stream);
149
150
155     int size();
156
157
161     void print(/*ostream &stream*/);
162
163
167     void sortbyMass();
168
169
175     cBrick& operator[](int position);
176
177
184     double getMassOfComposition(vector<int>& combarray, int numberofbasicbricks);
```

```
185
186
192     string getRealName(string& composition);
193
194
201     string getAcronymName(string& composition, bool useHTMLReferences);
202
203
209     string getTagName(string& composition);
210
211
218     string getAcronymNameOfTPeptide(const string& tcomposition, bool useHTMLReferences);
219
220
226     string getTagNameOfTPeptide(string& tcomposition);
227
228
232     void clear();
233
234
241     bool replaceAcronymsByIDs(string& sequence, string& errormessage);
242
243
248     void push_back(cBrick& brick);
249
250
254     void removeLastBrick();
255
256
262     string calculateSummaryFromComposition(int brickid);
263
264
269     void store(ofstream& os);
270
271
276     void load(ifstream& is);
277
278
285     bool checkKetideBlocks(cBrick& brickseries, bool regularblocksorder);
286
287
293     double getMassOfNeutralLosses(vector<int>& lossids);
294
295
301     string getSummaryFormulaOfNeutralLosses(vector<int>& lossids);
302
303
309     void getMapOfNeutralLosses(vector<int>& lossids, map<string, int>& summarymap);
310
311 };
312
313
314 #endif
```

17.5 core/cCandidate.h File Reference

The representation of a peptide sequence candidate.

```
#include <vector>
#include <unordered_set>
#include <string>
#include "core/cParameters.h"
#include "core/cBricksDatabase.h"
#include "core/cDeNovoGraphNode.h"
#include "core/cSummaryFormula.h"
```

Classes

- struct TRotationInfo

  An auxiliary structure representing a branch permutation of a branched peptide.
- class cCandidate

  The class representing a peptide sequence candidate.
- struct hash_cCandidate

  The structure defining a hash function of cCandidate.

Functions

- bool operator== (cCandidate const &a, cCandidate const &b)

  Overloaded operator ==.

17.5.1 Detailed Description

The representation of a peptide sequence candidate.

17.5.2 Function Documentation

17.5.2.1 operator==()

```
bool operator== (
            cCandidate const & a,
            cCandidate const & b )
```

Overloaded operator ==.

Parameters

a	first candidate
b	second candidate

Return values

bool	true when candidates are equal

17.6 cCandidate.h

Go to the documentation of this file.

```
1
7 #ifndef _CCANDIDATE_H
8 #define _CCANDIDATE_H
9
10 #include <vector>
11 #include <unordered_set>
12 #include <string>
13
14 #include "core/cParameters.h"
15 #include "core/cBricksDatabase.h"
16 #include "core/cDeNovoGraphNode.h"
17 #include "core/cSummaryFormula.h"
18
19 class cCandidateSet;
20 class cDeNovoGraph;
21
22
23 using namespace std;
24 using namespace boost;
25
26
30 struct TRotationInfo {
31
35     int id;
36
37
41     vector<int> bricks;
42
43
47     int middlebranchstart;
48
49
53     int middlebranchend;
54
55
```

```
59      int startmodifID;
60
61
65      int endmodifID;
66
67
71      int middlemodifID;
72
73
77      string tcomposition;
78
79
83      TRotationInfo() {
84          clear();
85      }
86
87
91      void clear() {
92          id = 0;
93          bricks.clear();
94          middlebranchstart = 0;
95          middlebranchend = 0;
96          startmodifID = 0;
97          endmodifID = 0;
98          middlemodifID = 0;
99          tcomposition = "";
100     }
101
102
107     bool startsWithBracket() {
108         if (tcomposition.size() == 0) {
109             return false;
110         }
111         return tcomposition[0] == '(';
112     }
113
118     bool endsWithBracket() {
119         if (tcomposition.size() == 0) {
120             return false;
121         }
122         return tcomposition.back() == ')';
123     }
124
125 };
126
127
131 class cCandidate {
132
133     int startmodifID;
134     int endmodifID;
135     int middlemodifID;
136     int branchstart;
137     int branchend;
138     vector<nodeEdge> path;
139     eFragmentIonType startiontype;
140     string name;
141     cSummaryFormula summary;
142
143     vector<string> composition;
144     string internalcomposition;
145     int numberofinternalbricks;
146
147     string realpeptidename;
148     string acronympeptidename;
149     vector<string> acronyms;
150     vector<string> backboneacronyms;
151     vector<string> branchacronyms;
152     string stringpath;
153
154     void updateInternalComposition();
155
156     void getPermutationsIter(cCandidateSet& permutations, vector<string>& currentcandidate, int
      position, bool* terminatecomputation);
157
158     void attachSubBranchCandidates(cCandidate& candidate, cCandidateSet& result, ePeptideType
      peptidetype, bool* terminatecomputation);
159
160     void attachAllBranches(cCandidate& candidate, cCandidateSet& result, ePeptideType peptidetype, bool*
      terminatecomputation);
161
162     void getPartialRotations(const string& composition, vector<string>& rotations);
163
164     void getPartialBranchCyclicRotations(const string& composition, vector<cCandidate>&
      branchcyclicrotations, int branchstart, int branchend);
165
166 public:
167
```

```
171     cCandidate();
172
173
184     cCandidate(vector<string>& composition, vector<nodeEdge>& path, eFragmentIonType startiontype, int
    startmodifID, int endmodifID, int middlemodifID, int middlepos);
185
186
198     cCandidate(vector<string>& composition, vector<nodeEdge>& path, eFragmentIonType startiontype, int
    startmodifID, int endmodifID, int middlemodifID, int branchstart, int branchend);
199
200
211     void setCandidate(vector<string>& composition, vector<nodeEdge>& path, eFragmentIonType
    startiontype, int startmodifID, int endmodifID, int middlemodifID, int middlepos);
212
213
225     void setCandidate(vector<string>& composition, vector<nodeEdge>& path, eFragmentIonType
    startiontype, int startmodifID, int endmodifID, int middlemodifID, int branchstart, int branchend);
226
227
231     void clear();
232
233
238     string& getComposition();
239
240
244     void revertComposition();
245
246
253     void prepareBranchedCandidates(cCandidateSet& result, ePeptideType peptidetype, bool*
    terminatecomputation);
254
255
260     string getTComposition();
261
262
268     string getRevertedTComposition(bool usebrackets);
269
270
276     string getAcronymsTComposition(cBricksDatabase& brickdatabase);
277
278
284     string getRealNameTComposition(cBricksDatabase& brickdatabase);
285
286
291     int getBranchSize();
292
293
298     int getBranchStart();
299
300
305     int getBranchEnd();
306
307
312     void getPermutationsOfBranches(vector<TRotationInfo>& tpermutations);
313
314
321     double calculatePrecursorMassFromBricks(cBricksDatabase& brickdatabasewithcombinations, cParameters*
    parameters);
322
323
329     bool isEqualTo(cCandidate& candidate);
330
331
337     void getPermutations(cCandidateSet& permutations, bool* terminatecomputation);
338
339
344     int getStartModifID();
345
346
351     void setStartModifID(int id);
352
353
358     int getEndModifID();
359
360
365     void setEndModifID(int id);
366
367
372     int getMiddleModifID();
373
374
379     void setMiddleModifID(int id);
380
381
387     bool compare(cCandidate& candidate);
388
389
```

```
394     vector<nodeEdge>& getPath();
395
396
402     bool hasEqualTPermutations(cCandidate& candidate);
403
404
410     bool hasOnlyArtificialBricks(cBricksDatabase& brickdatabasewithcombinations);
411
412
418     bool hasFirstBrickArtificial(cBricksDatabase& brickdatabasewithcombinations);
419
420
426     bool hasLastBrickArtificial(cBricksDatabase& brickdatabasewithcombinations);
427
428
434     bool hasLastBrickInvalid(cBricksDatabase& brickdatabasewithcombinations);
435
436
442     void getRotations(vector<string>& rotations, bool includerevertedrotations);
443
444
450     void getBranchCyclicRotations(vector<cCandidate>& branchcyclicrotations, bool
    includerevertedrotations);
451
452
460     cSummaryFormula calculateSummaryFormulaFromBricks(cParameters& parameters, ePeptideType peptidetype,
    double precursormass = 0);
461
462
467     cSummaryFormula& getSummaryFormula();
468
469
474     void setSummaryFormula(cSummaryFormula& summary);
475
476
481     void store(ofstream& os);
482
483
488     void load(ifstream& is);
489
490
495     void setName(string& name);
496
497
502     string& getName();
503
504
509     void setAcronyms(cBricksDatabase& bricksdatabase);
510
511
516     void setBackboneAcronyms(cBricksDatabase& bricksdatabase);
517
518
523     void setBranchAcronyms(cBricksDatabase& bricksdatabase);
524
525
530     vector<string>& getAcronyms();
531
532
537     vector<string>& getBackboneAcronyms();
538
539
544     vector<string>& getBranchAcronyms();
545
546
552     void setPath(cDeNovoGraph& graph, cParameters* parameters);
553
554
559     string& getPathAsString();
560
561
567     void setRealPeptideName(cBricksDatabase& bricksdatabase, ePeptideType peptidetype);
568
569
575     void setAcronymPeptideNameWithHTMLReferences(cBricksDatabase& bricksdatabase, ePeptideType
    peptidetype);
576
577
582     string& getRealPeptideName();
583
584
589     string& getAcronymPeptideNameWithHTMLReferences();
590
591
596     void setStartIonType(eFragmentIonType iontype);
597
598
```

```
603     eFragmentIonType getStartIonType();
604
605
613     bool checkKetideSequence(cBricksDatabase& bricksdatabase, ePeptideType peptidetype, bool
    regularblocksorder);
614
615
621     eResidueLossType getLeftResidueType(cBricksDatabase& bricksdatabase);
622
623
629     eResidueLossType getRightResidueType(cBricksDatabase& bricksdatabase);
630
631
639     bool checkKetideBlocks(cBricksDatabase& bricksdatabase, ePeptideType peptidetype, bool
    regularblocksorder);
640
641
642 };
643
644
648 struct hash_cCandidate {
649
655     size_t operator()(const cCandidate& candidate)const {
656         return std::hash<string>()(((cCandidate&)candidate).getComposition());
657     }
658
659 };
660
661
668 bool operator == (cCandidate const& a, cCandidate const& b);
669
670
671 #endif
672
```

17.7 core/cCandidateSet.h File Reference

The representation of a set of peptide sequence candidates.

```
#include <QMutex>
#include "core/cCandidate.h"
```

Classes

- class cCandidateSet

  The class representing a set of peptide sequence candidates.

17.7.1 Detailed Description

The representation of a set of peptide sequence candidates.

17.8 cCandidateSet.h

Go to the documentation of this file.

```
1
7 #ifndef _CCANDIDATESET_H
8 #define _CCANDIDATESET_H
9
10 #include <QMutex>
11
12 #include "core/cCandidate.h"
13
14
18 class cCandidateSet :  public QMutex {
19
20     unordered_set<cCandidate, hash_cCandidate> candidates;
21
22 public:
23
27     cCandidateSet();
28
29
34     int size();
35
```

```
36
41     unordered_set<cCandidate, hash_cCandidate>& getSet();
42
43 };
44
45
46 #endif
47
```

17.9 core/cDeNovoGraph.h File Reference

The implementation of the de novo graph.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include "core/cParameters.h"
#include "core/cDeNovoGraphNode.h"
#include "core/cPeakListSeries.h"
#include "core/cBricksDatabase.h"
#include "core/cTheoreticalSpectrumList.h"
#include "parallel/cGraphReaderThread.h"
```

Classes

- class cDeNovoGraph

  The class implementing the de novo graph.

Functions

- bool compareNodes (cDeNovoGraphNode &node1, cDeNovoGraphNode &node2)

  Compare two nodes.

Variables

- const long long unsigned **maximumcandidates** = 1000000000

  Maximum number of candidates to be read from the graph.

17.9.1 Detailed Description

The implementation of the de novo graph.

17.9.2 Function Documentation

17.9.2.1 compareNodes()

```
bool compareNodes (
            cDeNovoGraphNode & node1,
            cDeNovoGraphNode & node2 )
```

Compare two nodes.

Parameters

node1	reference to the first node
node2	reference to the second node

Return values

bool	true when the mz ratio of the first node is smaller than the mz ratio of the second node

17.10 cDeNovoGraph.h

Go to the documentation of this file.

```
1
7 #ifndef _CDENOVOGRAPH_H
8 #define _CDENOVOGRAPH_H
9
10 #include <iostream>
11 #include <fstream>
12 #include <vector>
13 #include <string>
14
15 #include "core/cParameters.h"
16 #include "core/cDeNovoGraphNode.h"
17 #include "core/cPeakListSeries.h"
18 #include "core/cBricksDatabase.h"
19 #include "core/cTheoreticalSpectrumList.h"
20 #include "parallel/cGraphReaderThread.h"
21
22 class cMainThread;
23
24
25 using namespace std;
26
27
31 const long long unsigned maximumcandidates = 1000000000;
32
33
40 bool compareNodes(cDeNovoGraphNode& node1, cDeNovoGraphNode& node2);
41
42
46 class cDeNovoGraph {
47
48     cParameters* parameters;
49     vector<cDeNovoGraphNode> graph;
50     cPeaksList sortedpeaklist;
51     cBricksDatabase bricksdatabasewithcombinations;
52     int lastsystemnode;
53     cMainThread* os;
54     int startnode;
55
56     cGraphReaderThread* graphreaderthread;
57
58     bool findPath(int sourcenodeid, int edgeid, int targetnodeid, string& composition, string brickspath,
      int maximumbricksincombination, vector<nodeEdge>& path, bool& terminatecomputation);
59
60     // create vectors of edges from temporary unordered_sets
61     void createVectorsOfEdges();
62
63     // check whether a path leads to the precursor
64     bool finishInPrecursor(int currentnode, double cummass);
65
66
67 public:
68
72     cDeNovoGraph();
73
74
80     void initialize(cMainThread& os, cParameters& parameters);
81
82
88     int createGraph(bool& terminatecomputation);
89
90
96     int removeEdgesFormingPathsNotFinishingInPrecursor(bool& terminatecomputation);
97
98
104     int removeEdgesFormingPathsNotStartingFromFirstNode(bool& terminatecomputation);
105
106
112     int connectEdgesFormingPathsNotFinishingInPrecursor(bool& terminatecomputation);
113
114
120     int connectEdgesFormingPathsNotStartingFromFirstNode(bool& terminatecomputation);
121
122
128     int removePathsWhichCanBeSubstitutedByLongerPath(bool& terminatecomputation);
```

```
129
130
138     void startGraphReader(cCandidateSet& candidates, long long unsigned& count, int scanmode, bool&
        terminatecomputation);
139
140
145     int getNumberOfEdges();
146
147
155     void printPaths(cMainThread* os, cDeNovoGraphNode& node, vector<string>& path, vector<string>&
        composition);
156
157
162     string printGraph();
163
164
170     cDeNovoGraphNode& operator[](int position);
171
172
176     void sortEdgesByPPMError();
177
178
182     void sortEdgesByTargetNodeIntensity();
183
184
189     cBricksDatabase* getBrickDatabaseWithCombinations();
190
191 };
192
193
194 #endif
```

17.11 core/cDeNovoGraphNode.h File Reference

The representation of a node in the de novo graph.

```
#include <unordered_set>
#include <vector>
#include <map>
#include <fstream>
#include "core/cFragmentIons.h"
```

Classes

- struct nodeEdge

  An auxiliary structure to store relationship between a node and an edge.
- struct cEdge

  The structure representing an outgoing edge from a node.
- struct hash_cEdge

  The structure defining a hash function of cEdge.
- class cDeNovoGraphNode

  The class representing a node in the de novo graph.

Functions

- bool operator== (nodeEdge const &a, nodeEdge const &b)

  Overloaded operator ==.
- bool operator== (cEdge const &a, cEdge const &b)

  Overloaded operator ==.
- bool compareEdgesByPPMError (const cEdge &a, const cEdge &b)

  Compare edges by ppm error.
- bool compareEdgesByTargetNodeIntensityDesc (const cEdge &a, const cEdge &b)

  Compare edges by target node intensity.

17.11.1 Detailed Description

The representation of a node in the de novo graph.

17.11.2 Function Documentation

17.11.2.1 compareEdgesByPPMError()

```
bool compareEdgesByPPMError (
            const cEdge & a,
            const cEdge & b )
```

Compare edges by ppm error.

Parameters

a	first edge
b	second edge

Return values

bool	true when *a* has smaller ppm error than *b*

17.11.2.2 compareEdgesByTargetNodeIntensityDesc()

```
bool compareEdgesByTargetNodeIntensityDesc (
            const cEdge & a,
            const cEdge & b )
```

Compare edges by target node intensity.

Parameters

a	first edge
b	second edge

Return values

bool	true when the target node of *a* has bigger intensity than the target node of *b*

17.11.2.3 operator==() [1/2]

```
bool operator== (
            cEdge const & a,
            cEdge const & b )
```

Overloaded operator ==.

Parameters

a	first edge
b	second edge

Return values

bool	true when edges are equal

17.11.2.4 operator==() [2/2]

```
bool operator== (
            nodeEdge const & a,
            nodeEdge const & b )
```

Overloaded operator ==.

Parameters

a	first nodeEdge
b	second nodeEdge

Return values

bool	true when nodeEdges are equal

17.12 cDeNovoGraphNode.h

Go to the documentation of this file.

```
1
7 #ifndef _CDENOVOGRAPHNODE_H
8 #define _CDENOVOGRAPHNODE_H
9
10 #include <unordered_set>
11 #include <vector>
12 #include <map>
13 #include <fstream>
14
15 #include "core/cFragmentIons.h"
16
17 using namespace std;
18
19
23 struct nodeEdge {
24
28     int nodeid;
29
30
34     int edgeid;
35
36
41     void store(ofstream& os);
42
43
48     void load(ifstream& is);
49
50 };
51
52
59 bool operator == (nodeEdge const& a, nodeEdge const& b);
60
61
65 struct cEdge {
66
70     int targetnode;
71
72
76     int brickid;
77
78
82     double ppmerror;
83
84
88     double massdifference;
```

```
89
90
94      int sourcecharge;
95
96
100     int targetcharge;
101
102
106     double targetintensity;
107
108
112     string composition;
113
114
118     string summary;
119
120
124     int startmodifID;
125
126
130     int middlemodifID;
131
132
136     int endmodifID;
137
138
142     eFragmentIonType sourceion;
143
144
148     eFragmentIonType targetion;
149
150
154     cEdge() {
155         clear();
156     }
157
158
162     void clear() {
163         targetnode = 0;
164         brickid = 0;
165         ppmerror = 0;
166         massdifference = 0;
167         sourcecharge = 1;
168         targetcharge = 1;
169         targetintensity = 0;
170         composition = "";
171         summary = "";
172         startmodifID = 0;
173         middlemodifID = 0;
174         endmodifID = 0;
175         sourceion = fragmentIonTypeEnd;
176         targetion = fragmentIonTypeEnd;
177     }
178
179 };
180
181
185 struct hash_cEdge {
186
192     size_t operator()(const cEdge& e)const {
193         return std::hash<int>()(e.targetnode) ^ std::hash<int>()(e.brickid) ^
      std::hash<int>()((int)(e.sourceion)) ^ std::hash<int>()((int)(e.targetion));
194     }
195
196 };
197
198
205 bool operator == (cEdge const& a, cEdge const& b);
206
207
214 bool compareEdgesByPPMError(const cEdge& a, const cEdge& b);
215
216
223 bool compareEdgesByTargetNodeIntensityDesc(const cEdge& a, const cEdge& b);
224
225
229 class cDeNovoGraphNode {
230
231     double mzratio;
232     double intensity;
233
234     unordered_set<cEdge, hash_cEdge> tempedges;
235     vector<cEdge> edges;
236     map<eFragmentIonType, bool> ionannotation;
237
238 public:
239
```

```
240
244     cDeNovoGraphNode();
245
246
250     void clear();
251
252
257     double getMZRatio();
258
259
264     void setMZRatio(double mzratio);
265
266
271     double getIntensity();
272
273
278     void setIntensity(double intensity);
279
280
285     void insertTempEdge(cEdge& e);
286
287
292     void insertEdge(cEdge& e);
293
294
298     void sortEdgesByPPMError();
299
300
304     void sortEdgesByTargetNodeIntensity();
305
306
310     void createVectorOfEdges();
311
312
317     int size();
318
319
325     cEdge& operator[](int position);
326
327
332     void removeEdge(int position);
333
334
338     void clearEdges();
339
340
345     void addIonAnnotation(eFragmentIonType iontype);
346
347
353     bool checkIonAnnotation(eFragmentIonType iontype);
354
355 };
356
357
358 #endif
```

17.13 core/cFragmentIons.h File Reference

Basic definitions of fragment ions occurring in the mass spectra.

```
#include <string>
#include <iostream>
#include <iomanip>
#include <fstream>
#include <map>
#include <vector>
#include <QMetaType>
#include "core/utilities.h"
```

Classes

- class cPeriodicTableMap

  The class representing a periodic table.

- struct isotope

  Representation of an isotope.

- class cIsotopeMap

  The class representing an isotope map.
- struct cFragmentIonType

  The struct for definition of a fragment ion type.
- class cFragmentIons

  The detailed definitions of all supported fragment ion types.
- struct neutralLoss

  The structure for a detailed definition of a neutral loss.
- class cDefaultNeutralLosses

  The class representing a vector of default neutral losses.

Enumerations

- enum eFragmentIonType {
  a_ion = 1 , **b_ion** , **c_ion** , **x_ion** ,
  y_ion , **z_ion** , **precursor_ion** , **cyclic_precursor_ion** ,
  l0h_ion , **l1h_ion** , **l2h_ion** , **r1h_ion** ,
  r2h_ion , **l1oh_ion** , **l2oh_ion** , **r1oh_ion** ,
  r2oh_ion , **linear_polyketide_precursor_ion_h_h** , **linear_polyketide_precursor_ion_h_oh** , **linear_↩polyketide_precursor_ion_oh_oh** ,
  cyclic_polyketide_precursor_ion , **fragmentIonTypeEnd** }

  The types of supported fragment ions.

Functions

- **Q_DECLARE_METATYPE** (ePeptideType)

  Register ePeptideType by Qt.
- double charge (double neutralmass, int charge)

  Get mzratio from neutral mass.
- double uncharge (double mzratio, int charge)

  Get neutral mass from mzratio.
- **Q_DECLARE_METATYPE** (vector< eFragmentIonType >)

  Register vector<eFragmentIonType> by Qt.
- **Q_DECLARE_METATYPE** (vector< cFragmentIonType >)

  Register vector<cFragmentIonType> by Qt.
- **Q_DECLARE_METATYPE** (cFragmentIons)

  Register cFragmentIons by Qt.
- void initializeFragmentIonsForDeNovoGraphOfCyclicPeptides (vector< eFragmentIonType > &fragmentions)

  Initialize fragment ion types for the de novo graph of cyclic peptides.
- void initializeFragmentIonsForDeNovoGraphOfLinearPeptides (vector< eFragmentIonType > &fragmentions)

  Initialize fragment ion types for the de novo graph of linear peptides.
- void initializeFragmentIonsForDeNovoGraphOfTPeptides (vector< eFragmentIonType > &fragmentions)

  Initialize fragment ion types for the de novo graph of branched peptides.
- void initializeFragmentIonsForDeNovoGraphOfBranchCyclicPeptides (vector< eFragmentIonType > &fragmentions)

  Initialize fragment ion types for the de novo graph of branch-cyclic peptides.
- void initializeFragmentIonsForDeNovoGraphOfLinearPolyketide (vector< eFragmentIonType > &fragmentions)

  Initialize fragment ion types for the de novo graph of a linear polyketide.
- void initializeFragmentIonsForDeNovoGraphOfCyclicPolyketide (vector< eFragmentIonType > &fragmentions)

  Initialize fragment ion types for the de novo graph of a cyclic polyketide.

- int loadModificationsFromPlainTextStream (ifstream &stream, vector< cFragmentIonType > &modifications, string &errormessage, bool ignoreerrors)

  Load modifications from a plain text stream.
- void storeModificationsToPlainTextStream (ofstream &stream, vector< cFragmentIonType > &modifications)

  Store modifications into a plain text stream.

Variables

- const double **H** = 1.0078250321
- const double **D** = 2.014102
- const double **T** = 3.016049
- const double **He** = 4.002606
- const double **Li** = 7.016004
- const double **Be** = 9.012182
- const double **B** = 11.009306
- const double **C** = 12.0
- const double **N** = 14.0030740052
- const double **O** = 15.9949146221
- const double **F** = 18.998404
- const double **Ne** = 19.992439
- const double **Na** = 22.989771
- const double **Mg** = 23.985043
- const double **Al** = 26.981539
- const double **Si** = 27.976927
- const double **P** = 30.97376151
- const double **S** = 31.97207069
- const double **Cl** = 34.968853
- const double **Ar** = 39.962383
- const double **K** = 38.963707
- const double **Ca** = 39.962589
- const double **Sc** = 44.95591
- const double **Ti** = 47.947948
- const double **V** = 50.943962
- const double **Cr** = 51.940514
- const double **Mn** = 54.938049
- const double **Fe** = 55.93494
- const double **Co** = 58.933201
- const double **Ni** = 57.935349
- const double **Cu** = 62.9296
- const double **Zn** = 63.929146
- const double **Ga** = 68.925583
- const double **Ge** = 73.921181
- const double **As** = 74.921600
- const double **Se** = 79.916519
- const double **Br** = 78.918327
- const double **Kr** = 83.911507
- const double **Rb** = 84.911789
- const double **Sr** = 87.905617
- const double **Y** = 88.905846
- const double **Zr** = 89.904701
- const double **Nb** = 92.906403
- const double **Mo** = 97.905411
- const double **Tc** = 97.907219
- const double **Ru** = 101.904388
- const double **Rh** = 102.905502

- const double **Pd** = 105.903481
- const double **Ag** = 106.905090
- const double **Cd** = 113.903358
- const double **In** = 114.903877
- const double **Sn** = 119.902199
- const double **Sb** = 120.903824
- const double **Te** = 129.906219
- const double **I** = 126.904457
- const double **Xe** = 131.904160
- const double **Cs** = 132.905441
- const double **Ba** = 137.905243
- const double **La** = 138.906342
- const double **Ce** = 139.905441
- const double **Pr** = 140.907593
- const double **Nd** = 141.907700
- const double **Pm** = 144.912750
- const double **Sm** = 151.919693
- const double **Eu** = 152.921204
- const double **Gd** = 157.924103
- const double **Tb** = 158.925293
- const double **Dy** = 163.929199
- const double **Ho** = 164.930298
- const double **Er** = 165.930298
- const double **Tm** = 168.934204
- const double **Yb** = 173.938904
- const double **Lu** = 174.940796
- const double **Hf** = 179.946503
- const double **Ta** = 180.947998
- const double **W** = 183.950897
- const double **Re** = 186.955597
- const double **Os** = 191.961502
- const double **Ir** = 192.962906
- const double **Pt** = 194.964798
- const double **Au** = 196.966507
- const double **Hg** = 201.970596
- const double **Tl** = 204.974396
- const double **Pb** = 207.976593
- const double **Bi** = 208.980392
- const double **Po** = 208.982422
- const double **At** = 209.987137
- const double **Rn** = 222.017563
- const double **Fr** = 209.996399
- const double **Ra** = 226.025406
- const double **Ac** = 227.027740
- const double **Th** = 232.038055
- const double **Pa** = 231.035904
- const double **U** = 238.050781
- const double **Np** = 237.048172
- const double **Pu** = 242.058746
- const double **Am** = 243.061371
- const double **Cm** = 247.070343
- const double **Bk** = 247.070297
- const double **Cf** = 251.079575
- const double **Es** = 252.082977
- const double **Fm** = 257.095093

- const double **Md** = 258.098419
- const double **No** = 259.101044
- const double **Lr** = 262.109802
- const double **R** = 0
- const double **X** = 0
- const double **e** = H - 1.00727645
- const double **H1** = H
- const double **H2** = D
- const double **He3** = 3.016029
- const double **He4** = He
- const double **Li6** = 6.015122
- const double **Li7** = Li
- const double **Be9** = Be
- const double **B10** = 10.012937
- const double **B11** = B
- const double **C12** = C
- const double **C13** = 13.003355
- const double **N14** = N
- const double **N15** = 15.000109
- const double **O16** = O
- const double **O17** = 16.999132
- const double **O18** = 17.999160
- const double **F19** = F
- const double **Ne20** = Ne
- const double **Ne21** = 20.993847
- const double **Ne22** = 21.991386
- const double **Na23** = Na
- const double **Mg24** = Mg
- const double **Mg25** = 24.985837
- const double **Mg26** = 25.982593
- const double **Al27** = Al
- const double **Si28** = Si
- const double **Si29** = 28.97649
- const double **Si30** = 29.973770
- const double **P31** = P
- const double **S32** = S
- const double **S33** = 32.971458
- const double **S34** = 33.967867
- const double **S36** = 35.967081
- const double **Cl35** = Cl
- const double **Cl37** = 36.965903
- const double **Ar36** = 35.967546
- const double **Ar38** = 37.962732
- const double **Ar40** = Ar
- const double **K39** = K
- const double **K40** = 39.963999
- const double **K41** = 40.961825
- const double **Ca40** = Ca
- const double **Ca42** = 41.958618
- const double **Ca43** = 42.958767
- const double **Ca44** = 43.955481
- const double **Ca46** = 45.953693
- const double **Ca48** = 47.952534
- const double **Sc45** = Sc
- const double **Ti46** = 45.952629

- const double **Ti47** = 46.951764
- const double **Ti48** = Ti
- const double **Ti49** = 48.947871
- const double **Ti50** = 49.944792
- const double **V50** = 49.947163
- const double **V51** = V
- const double **Cr50** = 49.946046
- const double **Cr52** = Cr
- const double **Cr53** = 52.940651
- const double **Cr54** = 53.938885
- const double **Mn55** = Mn
- const double **Fe54** = 53.939612
- const double **Fe56** = Fe
- const double **Fe57** = 56.935399
- const double **Fe58** = 57.933280
- const double **Co59** = Co
- const double **Ni58** = Ni
- const double **Ni60** = 59.930788
- const double **Ni61** = 60.931060
- const double **Ni62** = 61.928346
- const double **Ni64** = 63.927970
- const double **Cu63** = Cu
- const double **Cu65** = 64.927793
- const double **Zn64** = Zn
- const double **Zn66** = 65.926034
- const double **Zn67** = 66.927129
- const double **Zn68** = 67.924846
- const double **Zn70** = 69.925325
- const double **Ga69** = Ga
- const double **Ga71** = 70.9247
- const double **Ge70** = 69.924250
- const double **Ge72** = 71.922076
- const double **Ge73** = 72.923459
- const double **Ge74** = Ge
- const double **Ge76** = 75.921403
- const double **As75** = As
- const double **Se74** = 73.922477
- const double **Se76** = 75.919214
- const double **Se77** = 76.919915
- const double **Se78** = 77.917310
- const double **Se80** = Se
- const double **Se82** = 81.9167
- const double **Br79** = Br
- const double **Br81** = 80.916291
- const double **Kr78** = 77.920386
- const double **Kr80** = 79.916378
- const double **Kr82** = 81.913485
- const double **Kr83** = 82.914136
- const double **Kr84** = Kr
- const double **Kr86** = 85.910610
- const double **Rb85** = Rb
- const double **Rb87** = 86.909183
- const double **Sr84** = 83.913425
- const double **Sr86** = 85.909262
- const double **Sr87** = 86.908879

- const double **Sr88** = Sr
- const double **Y89** = Y
- const double **Zr90** = Zr
- const double **Zr91** = 90.905645
- const double **Zr92** = 91.90504
- const double **Zr94** = 93.906316
- const double **Zr96** = 95.908276
- const double **Nb93** = Nb
- const double **Mo92** = 91.906810
- const double **Mo94** = 93.905088
- const double **Mo95** = 94.905841
- const double **Mo96** = 95.904679
- const double **Mo97** = 96.906021
- const double **Mo98** = Mo
- const double **Mo100** = 99.907477
- const double **Tc98** = Tc
- const double **Ru96** = 95.907598
- const double **Ru98** = 97.905287
- const double **Ru99** = 98.905939
- const double **Ru100** = 99.904220
- const double **Ru101** = 100.905582
- const double **Ru102** = Ru
- const double **Ru104** = 103.90543
- const double **Rh103** = Rh
- const double **Pd102** = 101.905608
- const double **Pd104** = 103.904035
- const double **Pd105** = 104.905084
- const double **Pd106** = Pd
- const double **Pd108** = 107.903894
- const double **Pd110** = 109.905152
- const double **Ag107** = Ag
- const double **Ag109** = 108.904756
- const double **Cd106** = 105.906458
- const double **Cd108** = 107.904183
- const double **Cd110** = 109.903006
- const double **Cd111** = 110.904182
- const double **Cd112** = 111.902757
- const double **Cd113** = 112.904401
- const double **Cd114** = Cd
- const double **Cd116** = 115.904755
- const double **In113** = 112.904061
- const double **In115** = In
- const double **Sn112** = 111.904821
- const double **Sn114** = 113.902782
- const double **Sn115** = 114.903346
- const double **Sn116** = 115.901744
- const double **Sn117** = 116.902954
- const double **Sn118** = 117.901606
- const double **Sn119** = 118.903309
- const double **Sn120** = Sn
- const double **Sn122** = 121.90344
- const double **Sn124** = 123.905275
- const double **Sb121** = Sb
- const double **Sb123** = 122.904216
- const double **Te120** = 119.90402

- const double **Te122** = 121.903047
- const double **Te123** = 122.904273
- const double **Te124** = 123.902819
- const double **Te125** = 124.904425
- const double **Te126** = 125.903306
- const double **Te128** = 127.904461
- const double **Te130** = Te
- const double **I127** = I
- const double **Xe124** = 123.905896
- const double **Xe126** = 125.904269
- const double **Xe128** = 127.90353
- const double **Xe129** = 128.904779
- const double **Xe130** = 129.903508
- const double **Xe131** = 130.905082
- const double **Xe132** = Xe
- const double **Xe134** = 133.905395
- const double **Xe136** = 135.90722
- const double **Cs133** = Cs
- const double **Ba130** = 129.906310
- const double **Ba132** = 131.905056
- const double **Ba134** = 133.904503
- const double **Ba135** = 134.905683
- const double **Ba136** = 135.90457
- const double **Ba137** = 136.905821
- const double **Ba138** = Ba
- const double **La138** = 137.907107
- const double **La139** = La
- const double **Ce136** = 135.907144
- const double **Ce138** = 137.905986
- const double **Ce140** = Ce
- const double **Ce142** = 141.90924
- const double **Pr141** = Pr
- const double **Nd142** = Nd
- const double **Nd143** = 142.90981
- const double **Nd144** = 143.910083
- const double **Nd145** = 144.912569
- const double **Nd146** = 145.913112
- const double **Nd148** = 147.916889
- const double **Nd150** = 149.920887
- const double **Pm145** = Pm
- const double **Sm144** = 143.911995
- const double **Sm147** = 146.914893
- const double **Sm148** = 147.914818
- const double **Sm149** = 148.917180
- const double **Sm150** = 149.917271
- const double **Sm152** = Sm
- const double **Sm154** = 153.922205
- const double **Eu151** = 150.919846
- const double **Eu153** = Eu
- const double **Gd152** = 151.919788
- const double **Gd154** = 153.920862
- const double **Gd155** = 154.922619
- const double **Gd156** = 155.92212
- const double **Gd157** = 156.923957
- const double **Gd158** = Gd

- const double **Gd160** = 159.927051
- const double **Tb159** = Tb
- const double **Dy156** = 155.924278
- const double **Dy158** = 157.924405
- const double **Dy160** = 159.925194
- const double **Dy161** = 160.926930
- const double **Dy162** = 161.926795
- const double **Dy163** = 162.928728
- const double **Dy164** = Dy
- const double **Ho165** = Ho
- const double **Er162** = 161.928775
- const double **Er164** = 163.929197
- const double **Er166** = Er
- const double **Er167** = 166.932045
- const double **Er168** = 167.932368
- const double **Er170** = 169.935460
- const double **Tm169** = Tm
- const double **Yb168** = 167.933894
- const double **Yb170** = 169.934759
- const double **Yb171** = 170.936322
- const double **Yb172** = 171.936378
- const double **Yb173** = 172.938207
- const double **Yb174** = Yb
- const double **Yb176** = 175.942568
- const double **Lu175** = Lu
- const double **Lu176** = 175.942682
- const double **Hf174** = 173.94004
- const double **Hf176** = 175.941402
- const double **Hf177** = 176.94322
- const double **Hf178** = 177.943698
- const double **Hf179** = 178.945815
- const double **Hf180** = Hf
- const double **Ta180** = 179.947466
- const double **Ta181** = Ta
- const double **W180** = 179.946706
- const double **W182** = 181.948206
- const double **W183** = 182.950224
- const double **W184** = W
- const double **W186** = 185.954362
- const double **Re185** = 184.952956
- const double **Re187** = Re
- const double **Os184** = 183.952491
- const double **Os186** = 185.953838
- const double **Os187** = 186.955748
- const double **Os188** = 187.955836
- const double **Os189** = 188.958145
- const double **Os190** = 189.958445
- const double **Os192** = Os
- const double **Ir191** = 190.960591
- const double **Ir193** = Ir
- const double **Pt190** = 189.95993
- const double **Pt192** = 191.961035
- const double **Pt194** = 193.962664
- const double **Pt195** = Pt
- const double **Pt196** = 195.964935

- const double **Pt198** = 197.967876
- const double **Au197** = Au
- const double **Hg196** = 195.965815
- const double **Hg198** = 197.966752
- const double **Hg199** = 198.968262
- const double **Hg200** = 199.968309
- const double **Hg201** = 200.970285
- const double **Hg202** = Hg
- const double **Hg204** = 203.973476
- const double **Tl203** = 202.972329
- const double **Tl205** = Tl
- const double **Pb204** = 203.973029
- const double **Pb206** = 205.974449
- const double **Pb207** = 206.975881
- const double **Pb208** = Pb
- const double **Bi209** = Bi
- const double **Po209** = Po
- const double **At210** = At
- const double **Rn222** = Rn
- const double **Fr223** = Fr
- const double **Ra226** = Ra
- const double **Ac227** = Ac
- const double **Th232** = Th
- const double **Pa231** = Pa
- const double **U234** = 234.040946
- const double **U235** = 235.043923
- const double **U238** = U
- const double **Np237** = Np
- const double **Pu244** = Pu
- const double **Am243** = Am
- const double **Cm247** = Cm
- const double **Bk247** = Bk
- const double **Cf251** = Cf
- const double **Es252** = Es
- const double **Fm257** = Fm
- const double **Md258** = Md
- const double **No259** = No
- const double **Lr262** = Lr
- const double **Pr_H1** = 0.999885
- const double **Pr_H2** = 0.000115
- const double **Pr_He3** = 0.00000137
- const double **Pr_He4** = 0.99999863
- const double **Pr_Li6** = 0.0759
- const double **Pr_Li7** = 0.9241
- const double **Pr_Be9** = 1
- const double **Pr_B10** = 0.00199
- const double **Pr_B11** = 0.00801
- const double **Pr_C12** = 0.9893
- const double **Pr_C13** = 0.0107
- const double **Pr_N14** = 0.99632
- const double **Pr_N15** = 0.00368
- const double **Pr_O16** = 0.99757
- const double **Pr_O17** = 0.00038
- const double **Pr_O18** = 0.00205
- const double **Pr_F19** = 1

- const double **Pr_Ne20** = 0.9048
- const double **Pr_Ne21** = 0.0027
- const double **Pr_Ne22** = 0.0925
- const double **Pr_Na23** = 1
- const double **Pr_Mg24** = 0.7899
- const double **Pr_Mg25** = 0.1
- const double **Pr_Mg26** = 0.1101
- const double **Pr_Al27** = 1
- const double **Pr_Si28** = 0.922297
- const double **Pr_Si29** = 0.046832
- const double **Pr_Si30** = 0.030872
- const double **Pr_P31** = 1
- const double **Pr_S32** = 0.9493
- const double **Pr_S33** = 0.0076
- const double **Pr_S34** = 0.0429
- const double **Pr_S36** = 0.0002
- const double **Pr_Cl35** = 0.7578
- const double **Pr_Cl37** = 0.2422
- const double **Pr_Ar36** = 0.003365
- const double **Pr_Ar38** = 0.000632
- const double **Pr_Ar40** = 0.996003
- const double **Pr_K39** = 0.932581
- const double **Pr_K40** = 0.000117
- const double **Pr_K41** = 0.067302
- const double **Pr_Ca40** = 0.96941
- const double **Pr_Ca42** = 0.00647
- const double **Pr_Ca43** = 0.00135
- const double **Pr_Ca44** = 0.02086
- const double **Pr_Ca46** = 0.00004
- const double **Pr_Ca48** = 0.00187
- const double **Pr_Sc45** = 1
- const double **Pr_Ti46** = 0.0825
- const double **Pr_Ti47** = 0.0744
- const double **Pr_Ti48** = 0.7372
- const double **Pr_Ti49** = 0.0541
- const double **Pr_Ti50** = 0.0518
- const double **Pr_V50** = 0.0025
- const double **Pr_V51** = 0.9975
- const double **Pr_Cr50** = 0.04345
- const double **Pr_Cr52** = 0.83789
- const double **Pr_Cr53** = 0.09501
- const double **Pr_Cr54** = 0.02365
- const double **Pr_Mn55** = 1
- const double **Pr_Fe54** = 0.05845
- const double **Pr_Fe56** = 0.91754
- const double **Pr_Fe57** = 0.02119
- const double **Pr_Fe58** = 0.00282
- const double **Pr_Co59** = 1
- const double **Pr_Ni58** = 0.680769
- const double **Pr_Ni60** = 0.262231
- const double **Pr_Ni61** = 0.011399
- const double **Pr_Ni62** = 0.036345
- const double **Pr_Ni64** = 0.009256
- const double **Pr_Cu63** = 0.6917
- const double **Pr_Cu65** = 0.3083

- const double **Pr_Zn64** = 0.4863
- const double **Pr_Zn66** = 0.2790
- const double **Pr_Zn67** = 0.0410
- const double **Pr_Zn68** = 0.1875
- const double **Pr_Zn70** = 0.0062
- const double **Pr_Ga69** = 0.60108
- const double **Pr_Ga71** = 0.39892
- const double **Pr_Ge70** = 0.2084
- const double **Pr_Ge72** = 0.2754
- const double **Pr_Ge73** = 0.0773
- const double **Pr_Ge74** = 0.3628
- const double **Pr_Ge76** = 0.0761
- const double **Pr_As75** = 1
- const double **Pr_Se74** = 0.0089
- const double **Pr_Se76** = 0.0937
- const double **Pr_Se77** = 0.0763
- const double **Pr_Se78** = 0.2377
- const double **Pr_Se80** = 0.4961
- const double **Pr_Se82** = 0.0873
- const double **Pr_Br79** = 0.5069
- const double **Pr_Br81** = 0.4931
- const double **Pr_Kr78** = 0.0035
- const double **Pr_Kr80** = 0.0228
- const double **Pr_Kr82** = 0.1158
- const double **Pr_Kr83** = 0.1149
- const double **Pr_Kr84** = 0.57
- const double **Pr_Kr86** = 0.173
- const double **Pr_Rb85** = 0.7217
- const double **Pr_Rb87** = 0.2783
- const double **Pr_Sr84** = 0.0056
- const double **Pr_Sr86** = 0.0986
- const double **Pr_Sr87** = 0.07
- const double **Pr_Sr88** = 0.8258
- const double **Pr_Y89** = 1
- const double **Pr_Zr90** = 0.5145
- const double **Pr_Zr91** = 0.1122
- const double **Pr_Zr92** = 0.1715
- const double **Pr_Zr94** = 0.1738
- const double **Pr_Zr96** = 0.028
- const double **Pr_Nb93** = 1
- const double **Pr_Mo92** = 0.1484
- const double **Pr_Mo94** = 0.0925
- const double **Pr_Mo95** = 0.1592
- const double **Pr_Mo96** = 0.1668
- const double **Pr_Mo97** = 0.0955
- const double **Pr_Mo98** = 0.2413
- const double **Pr_Mo100** = 0.0963
- const double **Pr_Tc98** = 1
- const double **Pr_Ru96** = 0.0554
- const double **Pr_Ru98** = 0.0187
- const double **Pr_Ru99** = 0.1276
- const double **Pr_Ru100** = 0.126
- const double **Pr_Ru101** = 0.1706
- const double **Pr_Ru102** = 0.3155
- const double **Pr_Ru104** = 0.1862

- const double **Pr_Rh103** = 1
- const double **Pr_Pd102** = 0.0102
- const double **Pr_Pd104** = 0.1114
- const double **Pr_Pd105** = 0.2233
- const double **Pr_Pd106** = 0.2733
- const double **Pr_Pd108** = 0.2646
- const double **Pr_Pd110** = 0.1172
- const double **Pr_Ag107** = 0.51839
- const double **Pr_Ag109** = 0.48161
- const double **Pr_Cd106** = 0.0125
- const double **Pr_Cd108** = 0.0089
- const double **Pr_Cd110** = 0.1249
- const double **Pr_Cd111** = 0.128
- const double **Pr_Cd112** = 0.2413
- const double **Pr_Cd113** = 0.1222
- const double **Pr_Cd114** = 0.2873
- const double **Pr_Cd116** = 0.0749
- const double **Pr_In113** = 0.0429
- const double **Pr_In115** = 0.9571
- const double **Pr_Sn112** = 0.0097
- const double **Pr_Sn114** = 0.0066
- const double **Pr_Sn115** = 0.0034
- const double **Pr_Sn116** = 0.1454
- const double **Pr_Sn117** = 0.0768
- const double **Pr_Sn118** = 0.2422
- const double **Pr_Sn119** = 0.0859
- const double **Pr_Sn120** = 0.3258
- const double **Pr_Sn122** = 0.0463
- const double **Pr_Sn124** = 0.0579
- const double **Pr_Sb121** = 0.5721
- const double **Pr_Sb123** = 0.4279
- const double **Pr_Te120** = 0.0009
- const double **Pr_Te122** = 0.0255
- const double **Pr_Te123** = 0.0089
- const double **Pr_Te124** = 0.0474
- const double **Pr_Te125** = 0.0707
- const double **Pr_Te126** = 0.1884
- const double **Pr_Te128** = 0.3174
- const double **Pr_Te130** = 0.3408
- const double **Pr_I127** = 1
- const double **Pr_Xe124** = 0.0009
- const double **Pr_Xe126** = 0.0009
- const double **Pr_Xe128** = 0.0192
- const double **Pr_Xe129** = 0.2644
- const double **Pr_Xe130** = 0.0408
- const double **Pr_Xe131** = 0.2118
- const double **Pr_Xe132** = 0.2689
- const double **Pr_Xe134** = 0.1044
- const double **Pr_Xe136** = 0.0887
- const double **Pr_Cs133** = 1
- const double **Pr_Ba130** = 0.00106
- const double **Pr_Ba132** = 0.00101
- const double **Pr_Ba134** = 0.02417
- const double **Pr_Ba135** = 0.06592
- const double **Pr_Ba136** = 0.07854

- const double **Pr_Ba137** = 0.11232
- const double **Pr_Ba138** = 0.71698
- const double **Pr_La138** = 0.0009
- const double **Pr_La139** = 0.9991
- const double **Pr_Ce136** = 0.00185
- const double **Pr_Ce138** = 0.00251
- const double **Pr_Ce140** = 0.8845
- const double **Pr_Ce142** = 0.11114
- const double **Pr_Pr141** = 1
- const double **Pr_Nd142** = 0.272
- const double **Pr_Nd143** = 0.122
- const double **Pr_Nd144** = 0.238
- const double **Pr_Nd145** = 0.083
- const double **Pr_Nd146** = 0.172
- const double **Pr_Nd148** = 0.057
- const double **Pr_Nd150** = 0.056
- const double **Pr_Pm145** = 1
- const double **Pr_Sm144** = 0.0307
- const double **Pr_Sm147** = 0.1499
- const double **Pr_Sm148** = 0.1124
- const double **Pr_Sm149** = 0.1382
- const double **Pr_Sm150** = 0.0738
- const double **Pr_Sm152** = 0.2675
- const double **Pr_Sm154** = 0.2275
- const double **Pr_Eu151** = 0.4781
- const double **Pr_Eu153** = 0.5219
- const double **Pr_Gd152** = 0.002
- const double **Pr_Gd154** = 0.0218
- const double **Pr_Gd155** = 0.148
- const double **Pr_Gd156** = 0.2047
- const double **Pr_Gd157** = 0.1565
- const double **Pr_Gd158** = 0.2484
- const double **Pr_Gd160** = 0.2186
- const double **Pr_Tb159** = 1
- const double **Pr_Dy156** = 0.0006
- const double **Pr_Dy158** = 0.001
- const double **Pr_Dy160** = 0.0234
- const double **Pr_Dy161** = 0.1891
- const double **Pr_Dy162** = 0.2551
- const double **Pr_Dy163** = 0.249
- const double **Pr_Dy164** = 0.2818
- const double **Pr_Ho165** = 1
- const double **Pr_Er162** = 0.0014
- const double **Pr_Er164** = 0.0161
- const double **Pr_Er166** = 0.3361
- const double **Pr_Er167** = 0.2293
- const double **Pr_Er168** = 0.2678
- const double **Pr_Er170** = 0.1493
- const double **Pr_Tm169** = 1
- const double **Pr_Yb168** = 0.0013
- const double **Pr_Yb170** = 0.0304
- const double **Pr_Yb171** = 0.1428
- const double **Pr_Yb172** = 0.2183
- const double **Pr_Yb173** = 0.1613
- const double **Pr_Yb174** = 0.3183

- const double **Pr_Yb176** = 0.1276
- const double **Pr_Lu175** = 0.9741
- const double **Pr_Lu176** = 0.0259
- const double **Pr_Hf174** = 0.0016
- const double **Pr_Hf176** = 0.0526
- const double **Pr_Hf177** = 0.186
- const double **Pr_Hf178** = 0.2728
- const double **Pr_Hf179** = 0.1362
- const double **Pr_Hf180** = 0.3508
- const double **Pr_Ta180** = 0.00012
- const double **Pr_Ta181** = 0.99988
- const double **Pr_W180** = 0.0012
- const double **Pr_W182** = 0.265
- const double **Pr_W183** = 0.1431
- const double **Pr_W184** = 0.3064
- const double **Pr_W186** = 0.2843
- const double **Pr_Re185** = 0.374
- const double **Pr_Re187** = 0.626
- const double **Pr_Os184** = 0.0002
- const double **Pr_Os186** = 0.0159
- const double **Pr_Os187** = 0.0196
- const double **Pr_Os188** = 0.1324
- const double **Pr_Os189** = 0.1615
- const double **Pr_Os190** = 0.2626
- const double **Pr_Os192** = 0.4078
- const double **Pr_Ir191** = 0.373
- const double **Pr_Ir193** = 0.627
- const double **Pr_Pt190** = 0.00014
- const double **Pr_Pt192** = 0.00782
- const double **Pr_Pt194** = 0.32967
- const double **Pr_Pt195** = 0.33832
- const double **Pr_Pt196** = 0.25242
- const double **Pr_Pt198** = 0.07163
- const double **Pr_Au197** = 1
- const double **Pr_Hg196** = 0.0015
- const double **Pr_Hg198** = 0.0997
- const double **Pr_Hg199** = 0.1687
- const double **Pr_Hg200** = 0.231
- const double **Pr_Hg201** = 0.1318
- const double **Pr_Hg202** = 0.2986
- const double **Pr_Hg204** = 0.0687
- const double **Pr_Tl203** = 0.29524
- const double **Pr_Tl205** = 0.70476
- const double **Pr_Pb204** = 0.014
- const double **Pr_Pb206** = 0.241
- const double **Pr_Pb207** = 0.221
- const double **Pr_Pb208** = 0.524
- const double **Pr_Bi209** = 1
- const double **Pr_Po209** = 1
- const double **Pr_At210** = 1
- const double **Pr_Rn222** = 1
- const double **Pr_Fr223** = 1
- const double **Pr_Ra226** = 1
- const double **Pr_Ac227** = 1
- const double **Pr_Th232** = 1

- const double **Pr_Pa231** = 1
- const double **Pr_U234** = 0.000055
- const double **Pr_U235** = 0.0072
- const double **Pr_U238** = 0.992745
- const double **Pr_Np237** = 1
- const double **Pr_Pu244** = 1
- const double **Pr_Am243** = 1
- const double **Pr_Cm247** = 1
- const double **Pr_Bk247** = 1
- const double **Pr_Cf251** = 1
- const double **Pr_Es252** = 1
- const double **Pr_Fm257** = 1
- const double **Pr_Md258** = 1
- const double **Pr_No259** = 1
- const double **Pr_Lr262** = 1
- cPeriodicTableMap **periodictablemap**

  Global periodic table map.
- cIsotopeMap **isotopemap**

  Global isotope map.
- const double **H2O** = 2 ∗ H + O
- const double **NH3** = 3 ∗ H + N
- const double **CO** = C + O
- const double **A_ION** = H - e - C - O
- const double **B_ION** = H - e
- const double **C_ION** = H - e + NH3
- const double **X_ION** = H - e + 2 ∗ O + C
- const double **Y_ION** = H - e + H2O
- const double **Z_ION** = H - e + O - N - H
- const double **PRECURSOR_ION** = H - e + H2O
- const double **PRECURSOR_ION_CYCLIC** = H - e
- const double **L0H_ION** = H - e - 2 ∗ H
- const double **L1H_ION** = H - e
- const double **L2H_ION** = H - e + 2 ∗ H
- const double **R1H_ION** = H - e
- const double **R2H_ION** = H - e + 2 ∗ H
- const double **L1OH_ION** = H - e + O
- const double **L2OH_ION** = H - e + H2O
- const double **R1OH_ION** = H - e + O
- const double **R2OH_ION** = H - e + H2O
- const double **PRECURSOR_ION_LINEAR_POLYKETIDE_H_H** = H - e + 2 ∗ H
- const double **PRECURSOR_ION_LINEAR_POLYKETIDE_H_OH** = H - e + H2O
- const double **PRECURSOR_ION_LINEAR_POLYKETIDE_OH_OH** = H - e + 2 ∗ H + 2 ∗ O
- cDefaultNeutralLosses **defaultneutrallosses**

  Default neutral losses definitions.

17.13.1 Detailed Description

Basic definitions of fragment ions occurring in the mass spectra.

17.13.2 Function Documentation

17.13.2.1 charge()

```
double charge (
            double neutralmass,
            int charge )
```

Get mzratio from neutral mass.

Parameters

neutralmass	neutral mass
charge	charge

Return values

double	mzratio

17.13.2.2 initializeFragmentIonsForDeNovoGraphOfBranchCyclicPeptides()

```
void initializeFragmentIonsForDeNovoGraphOfBranchCyclicPeptides (
            vector< eFragmentIonType > & fragmentions )
```

Initialize fragment ion types for the de novo graph of branch-cyclic peptides.

Parameters

fragmentions	reference to a vector of fragment ion types

17.13.2.3 initializeFragmentIonsForDeNovoGraphOfCyclicPeptides()

```
void initializeFragmentIonsForDeNovoGraphOfCyclicPeptides (
            vector< eFragmentIonType > & fragmentions )
```

Initialize fragment ion types for the de novo graph of cyclic peptides.

Parameters

fragmentions	reference to a vector of fragment ion types

17.13.2.4 initializeFragmentIonsForDeNovoGraphOfCyclicPolyketide()

```
void initializeFragmentIonsForDeNovoGraphOfCyclicPolyketide (
            vector< eFragmentIonType > & fragmentions )
```

Initialize fragment ion types for the de novo graph of a cyclic polyketide.

Parameters

fragmentions	reference to a vector of fragment ion types

17.13.2.5 initializeFragmentIonsForDeNovoGraphOfLinearPeptides()

```
void initializeFragmentIonsForDeNovoGraphOfLinearPeptides (
```

```
            vector< eFragmentIonType > & fragmentions )
```

Initialize fragment ion types for the de novo graph of linear peptides.

Parameters

fragmentions	reference to a vector of fragment ion types

17.13.2.6 initializeFragmentIonsForDeNovoGraphOfLinearPolyketide()

```
void initializeFragmentIonsForDeNovoGraphOfLinearPolyketide (
            vector< eFragmentIonType > & fragmentions )
```

Initialize fragment ion types for the de novo graph of a linear polyketide.

Parameters

fragmentions	reference to a vector of fragment ion types

17.13.2.7 initializeFragmentIonsForDeNovoGraphOfTPeptides()

```
void initializeFragmentIonsForDeNovoGraphOfTPeptides (
            vector< eFragmentIonType > & fragmentions )
```

Initialize fragment ion types for the de novo graph of branched peptides.

Parameters

fragmentions	reference to a vector of fragment ion types

17.13.2.8 loadModificationsFromPlainTextStream()

```
int loadModificationsFromPlainTextStream (
            ifstream & stream,
            vector< cFragmentIonType > & modifications,
            string & errormessage,
            bool ignoreerrors )
```

Load modifications from a plain text stream.

Parameters

stream	reference to an input file stream
modifications	reference to a vector where modifications will be placed
errormessage	reference to a string where an error message might be stored
ignoreerrors	if true errors are ignored

Return values

int	0 when modifications were successfully loaded, -1 when an error occurred (*errormessage* is filled up)

17.13.2.9 storeModificationsToPlainTextStream()

```
void storeModificationsToPlainTextStream (
            ofstream & stream,
            vector< cFragmentIonType > & modifications )
```

Store modifications into a plain text stream.

Parameters

stream	reference to an output file stream
modifications	reference to a vector of modifications which will be stored

17.13.2.10 uncharge()

```
double uncharge (
            double mzratio,
            int charge )
```

Get neutral mass from mzratio.

Parameters

mzratio	mzratio
charge	charge

Return values

double	neutral mass

17.14 cFragmentIons.h

Go to the documentation of this file.

```
1
7 #ifndef _CFRAGMENTIONS_H
8 #define _CFRAGMENTIONS_H
9
10 #include <string>
11 #include <iostream>
12 #include <iomanip>
13 #include <fstream>
14 #include <map>
15 #include <vector>
16 #include <QMetaType>
17 #include "core/utilities.h"
18
19
20 using namespace std;
21
22
23 // monoisotopic masses
24 const double H = 1.0078250321;
25 const double D = 2.014102;
26 const double T = 3.016049;
27 const double He = 4.002606;
28 const double Li = 7.016004;
29 const double Be = 9.012182;
30 const double B = 11.009306;
31 const double C = 12.0;
32 const double N = 14.0030740052;
33 const double O = 15.9949146221;
34 const double F = 18.998404;
35 const double Ne = 19.992439;
36 const double Na = 22.989771;
37 const double Mg = 23.985043;
38 const double Al = 26.981539;
39 const double Si = 27.976927;
```

```
40 const double P = 30.97376151;
41 const double S = 31.97207069;
42 const double Cl = 34.968853;
43 const double Ar = 39.962383;
44 const double K = 38.963707;
45 const double Ca = 39.962589;
46 const double Sc = 44.95591;
47 const double Ti = 47.947948;
48 const double V = 50.943962;
49 const double Cr = 51.940514;
50 const double Mn = 54.938049;
51 const double Fe = 55.93494;
52 const double Co = 58.933201;
53 const double Ni = 57.935349;
54 const double Cu = 62.9296;
55 const double Zn = 63.929146;
56 const double Ga = 68.925583;
57 const double Ge = 73.921181;
58 const double As = 74.921600;
59 const double Se = 79.916519;
60 const double Br = 78.918327;
61 const double Kr = 83.911507;
62 const double Rb = 84.911789;
63 const double Sr = 87.905617;
64 const double Y = 88.905846;
65 const double Zr = 89.904701;
66 const double Nb = 92.906403;
67 const double Mo = 97.905411;
68 const double Tc = 97.907219;
69 const double Ru = 101.904388;
70 const double Rh = 102.905502;
71 const double Pd = 105.903481;
72 const double Ag = 106.905090;
73 const double Cd = 113.903358;
74 const double In = 114.903877;
75 const double Sn = 119.902199;
76 const double Sb = 120.903824;
77 const double Te = 129.906219;
78 const double I = 126.904457;
79 const double Xe = 131.904160;
80 const double Cs = 132.905441;
81 const double Ba = 137.905243;
82 const double La = 138.906342;
83 const double Ce = 139.905441;
84 const double Pr = 140.907593;
85 const double Nd = 141.907700;
86 const double Pm = 144.912750;
87 const double Sm = 151.919693;
88 const double Eu = 152.921204;
89 const double Gd = 157.924103;
90 const double Tb = 158.925293;
91 const double Dy = 163.929199;
92 const double Ho = 164.930298;
93 const double Er = 165.930298;
94 const double Tm = 168.934204;
95 const double Yb = 173.938904;
96 const double Lu = 174.940796;
97 const double Hf = 179.946503;
98 const double Ta = 180.947998;
99 const double W = 183.950897;
100 const double Re = 186.955597;
101 const double Os = 191.961502;
102 const double Ir = 192.962906;
103 const double Pt = 194.964798;
104 const double Au = 196.966507;
105 const double Hg = 201.970596;
106 const double Tl = 204.974396;
107 const double Pb = 207.976593;
108 const double Bi = 208.980392;
109 const double Po = 208.982422;
110 const double At = 209.987137;
111 const double Rn = 222.017563;
112 const double Fr = 209.996399;
113 const double Ra = 226.025406;
114 const double Ac = 227.027740;
115 const double Th = 232.038055;
116 const double Pa = 231.035904;
117 const double U = 238.050781;
118 const double Np = 237.048172;
119 const double Pu = 242.058746;
120 const double Am = 243.061371;
121 const double Cm = 247.070343;
122 const double Bk = 247.070297;
123 const double Cf = 251.079575;
124 const double Es = 252.082977;
125 const double Fm = 257.095093;
126 const double Md = 258.098419;
```

```
127 const double No = 259.101044;
128 const double Lr = 262.109802;
129
130 const double R = 0;
131 const double X = 0;
132 const double e = H - 1.00727645;
133
134
135 // monoisotopic masses of natural isotopes
136 const double H1 = H;
137 const double H2 = D;
138 //const double H3 = T;
139 const double He3 = 3.016029;
140 const double He4 = He;
141 const double Li6 = 6.015122;
142 const double Li7 = Li;
143 const double Be9 = Be;
144 const double B10 = 10.012937;
145 const double B11 = B;
146 const double C12 = C;
147 const double C13 = 13.003355;
148 //const double C14 = 14.003242;
149 const double N14 = N;
150 const double N15 = 15.000109;
151 const double O16 = O;
152 const double O17 = 16.999132;
153 const double O18 = 17.999160;
154 const double F19 = F;
155 const double Ne20 = Ne;
156 const double Ne21 = 20.993847;
157 const double Ne22 = 21.991386;
158 const double Na23 = Na;
159 const double Mg24 = Mg;
160 const double Mg25 = 24.985837;
161 const double Mg26 = 25.982593;
162 const double Al27 = Al;
163 const double Si28 = Si;
164 const double Si29 = 28.97649;
165 const double Si30 = 29.973770;
166 const double P31 = P;
167 const double S32 = S;
168 const double S33 = 32.971458;
169 const double S34 = 33.967867;
170 const double S36 = 35.967081;
171 const double Cl35 = Cl;
172 const double Cl37 = 36.965903;
173 const double Ar36 = 35.967546;
174 const double Ar38 = 37.962732;
175 const double Ar40 = Ar;
176 const double K39 = K;
177 const double K40 = 39.963999;
178 const double K41 = 40.961825;
179 const double Ca40 = Ca;
180 const double Ca42 = 41.958618;
181 const double Ca43 = 42.958767;
182 const double Ca44 = 43.955481;
183 const double Ca46 = 45.953693;
184 const double Ca48 = 47.952534;
185 const double Sc45 = Sc;
186 const double Ti46 = 45.952629;
187 const double Ti47 = 46.951764;
188 const double Ti48 = Ti;
189 const double Ti49 = 48.947871;
190 const double Ti50 = 49.944792;
191 const double V50 = 49.947163;
192 const double V51 = V;
193 const double Cr50 = 49.946046;
194 const double Cr52 = Cr;
195 const double Cr53 = 52.940651;
196 const double Cr54 = 53.938885;
197 const double Mn55 = Mn;
198 const double Fe54 = 53.939612;
199 const double Fe56 = Fe;
200 const double Fe57 = 56.935399;
201 const double Fe58 = 57.933280;
202 const double Co59 = Co;
203 const double Ni58 = Ni;
204 const double Ni60 = 59.930788;
205 const double Ni61 = 60.931060;
206 const double Ni62 = 61.928346;
207 const double Ni64 = 63.927970;
208 const double Cu63 = Cu;
209 const double Cu65 = 64.927793;
210 const double Zn64 = Zn;
211 const double Zn66 = 65.926034;
212 const double Zn67 = 66.927129;
213 const double Zn68 = 67.924846;
```

```
214 const double Zn70 = 69.925325;
215 const double Ga69 = Ga;
216 const double Ga71 = 70.9247;
217 const double Ge70 = 69.924250;
218 const double Ge72 = 71.922076;
219 const double Ge73 = 72.923459;
220 const double Ge74 = Ge;
221 const double Ge76 = 75.921403;
222 const double As75 = As;
223 const double Se74 = 73.922477;
224 const double Se76 = 75.919214;
225 const double Se77 = 76.919915;
226 const double Se78 = 77.917310;
227 const double Se80 = Se;
228 const double Se82 = 81.9167;
229 const double Br79 = Br;
230 const double Br81 = 80.916291;
231 const double Kr78 = 77.920386;
232 const double Kr80 = 79.916378;
233 const double Kr82 = 81.913485;
234 const double Kr83 = 82.914136;
235 const double Kr84 = Kr;
236 const double Kr86 = 85.910610;
237 const double Rb85 = Rb;
238 const double Rb87 = 86.909183;
239 const double Sr84 = 83.913425;
240 const double Sr86 = 85.909262;
241 const double Sr87 = 86.908879;
242 const double Sr88 = Sr;
243 const double Y89 = Y;
244 const double Zr90 = Zr;
245 const double Zr91 = 90.905645;
246 const double Zr92 = 91.90504;
247 const double Zr94 = 93.906316;
248 const double Zr96 = 95.908276;
249 const double Nb93 = Nb;
250 const double Mo92 = 91.906810;
251 const double Mo94 = 93.905088;
252 const double Mo95 = 94.905841;
253 const double Mo96 = 95.904679;
254 const double Mo97 = 96.906021;
255 const double Mo98 = Mo;
256 const double Mo100 = 99.907477;
257 const double Tc98 = Tc;
258 const double Ru96 = 95.907598;
259 const double Ru98 = 97.905287;
260 const double Ru99 = 98.905939;
261 const double Ru100 = 99.904220;
262 const double Ru101 = 100.905582;
263 const double Ru102 = Ru;
264 const double Ru104 = 103.90543;
265 const double Rh103 = Rh;
266 const double Pd102 = 101.905608;
267 const double Pd104 = 103.904035;
268 const double Pd105 = 104.905084;
269 const double Pd106 = Pd;
270 const double Pd108 = 107.903894;
271 const double Pd110 = 109.905152;
272 const double Ag107 = Ag;
273 const double Ag109 = 108.904756;
274 const double Cd106 = 105.906458;
275 const double Cd108 = 107.904183;
276 const double Cd110 = 109.903006;
277 const double Cd111 = 110.904182;
278 const double Cd112 = 111.902757;
279 const double Cd113 = 112.904401;
280 const double Cd114 = Cd;
281 const double Cd116 = 115.904755;
282 const double In113 = 112.904061;
283 const double In115 = In;
284 const double Sn112 = 111.904821;
285 const double Sn114 = 113.902782;
286 const double Sn115 = 114.903346;
287 const double Sn116 = 115.901744;
288 const double Sn117 = 116.902954;
289 const double Sn118 = 117.901606;
290 const double Sn119 = 118.903309;
291 const double Sn120 = Sn;
292 const double Sn122 = 121.90344;
293 const double Sn124 = 123.905275;
294 const double Sb121 = Sb;
295 const double Sb123 = 122.904216;
296 const double Te120 = 119.90402;
297 const double Te122 = 121.903047;
298 const double Te123 = 122.904273;
299 const double Te124 = 123.902819;
300 const double Te125 = 124.904425;
```

```
301 const double Te126 = 125.903306;
302 const double Te128 = 127.904461;
303 const double Te130 = Te;
304 const double I127 = I;
305 const double Xe124 = 123.905896;
306 const double Xe126 = 125.904269;
307 const double Xe128 = 127.90353;
308 const double Xe129 = 128.904779;
309 const double Xe130 = 129.903508;
310 const double Xe131 = 130.905082;
311 const double Xe132 = Xe;
312 const double Xe134 = 133.905395;
313 const double Xe136 = 135.90722;
314 const double Cs133 = Cs;
315 const double Ba130 = 129.906310;
316 const double Ba132 = 131.905056;
317 const double Ba134 = 133.904503;
318 const double Ba135 = 134.905683;
319 const double Ba136 = 135.90457;
320 const double Ba137 = 136.905821;
321 const double Ba138 = Ba;
322 const double La138 = 137.907107;
323 const double La139 = La;
324 const double Ce136 = 135.907144;
325 const double Ce138 = 137.905986;
326 const double Ce140 = Ce;
327 const double Ce142 = 141.90924;
328 const double Pr141 = Pr;
329 const double Nd142 = Nd;
330 const double Nd143 = 142.90981;
331 const double Nd144 = 143.910083;
332 const double Nd145 = 144.912569;
333 const double Nd146 = 145.913112;
334 const double Nd148 = 147.916889;
335 const double Nd150 = 149.920887;
336 const double Pm145 = Pm;
337 const double Sm144 = 143.911995;
338 const double Sm147 = 146.914893;
339 const double Sm148 = 147.914818;
340 const double Sm149 = 148.917180;
341 const double Sm150 = 149.917271;
342 const double Sm152 = Sm;
343 const double Sm154 = 153.922205;
344 const double Eu151 = 150.919846;
345 const double Eu153 = Eu;
346 const double Gd152 = 151.919788;
347 const double Gd154 = 153.920862;
348 const double Gd155 = 154.922619;
349 const double Gd156 = 155.92212;
350 const double Gd157 = 156.923957;
351 const double Gd158 = Gd;
352 const double Gd160 = 159.927051;
353 const double Tb159 = Tb;
354 const double Dy156 = 155.924278;
355 const double Dy158 = 157.924405;
356 const double Dy160 = 159.925194;
357 const double Dy161 = 160.926930;
358 const double Dy162 = 161.926795;
359 const double Dy163 = 162.928728;
360 const double Dy164 = Dy;
361 const double Ho165 = Ho;
362 const double Er162 = 161.928775;
363 const double Er164 = 163.929197;
364 const double Er166 = Er;
365 const double Er167 = 166.932045;
366 const double Er168 = 167.932368;
367 const double Er170 = 169.935460;
368 const double Tm169 = Tm;
369 const double Yb168 = 167.933894;
370 const double Yb170 = 169.934759;
371 const double Yb171 = 170.936322;
372 const double Yb172 = 171.936378;
373 const double Yb173 = 172.938207;
374 const double Yb174 = Yb;
375 const double Yb176 = 175.942568;
376 const double Lu175 = Lu;
377 const double Lu176 = 175.942682;
378 const double Hf174 = 173.94004;
379 const double Hf176 = 175.941402;
380 const double Hf177 = 176.94322;
381 const double Hf178 = 177.943698;
382 const double Hf179 = 178.945815;
383 const double Hf180 = Hf;
384 const double Ta180 = 179.947466;
385 const double Ta181 = Ta;
386 const double W180 = 179.946706;
387 const double W182 = 181.948206;
```

```
388 const double W183 = 182.950224;
389 const double W184 = W;
390 const double W186 = 185.954362;
391 const double Re185 = 184.952956;
392 const double Re187 = Re;
393 const double Os184 = 183.952491;
394 const double Os186 = 185.953838;
395 const double Os187 = 186.955748;
396 const double Os188 = 187.955836;
397 const double Os189 = 188.958145;
398 const double Os190 = 189.958445;
399 const double Os192 = Os;
400 const double Ir191 = 190.960591;
401 const double Ir193 = Ir;
402 const double Pt190 = 189.95993;
403 const double Pt192 = 191.961035;
404 const double Pt194 = 193.962664;
405 const double Pt195 = Pt;
406 const double Pt196 = 195.964935;
407 const double Pt198 = 197.967876;
408 const double Au197 = Au;
409 const double Hg196 = 195.965815;
410 const double Hg198 = 197.966752;
411 const double Hg199 = 198.968262;
412 const double Hg200 = 199.968309;
413 const double Hg201 = 200.970285;
414 const double Hg202 = Hg;
415 const double Hg204 = 203.973476;
416 const double Tl203 = 202.972329;
417 const double Tl205 = Tl;
418 const double Pb204 = 203.973029;
419 const double Pb206 = 205.974449;
420 const double Pb207 = 206.975881;
421 const double Pb208 = Pb;
422 const double Bi209 = Bi;
423 const double Po209 = Po;
424 const double At210 = At;
425 const double Rn222 = Rn;
426 const double Fr223 = Fr;
427 const double Ra226 = Ra;
428 const double Ac227 = Ac;
429 const double Th232 = Th;
430 const double Pa231 = Pa;
431 const double U234 = 234.040946;
432 const double U235 = 235.043923;
433 const double U238 = U;
434 const double Np237 = Np;
435 const double Pu244 = Pu;
436 const double Am243 = Am;
437 const double Cm247 = Cm;
438 const double Bk247 = Bk;
439 const double Cf251 = Cf;
440 const double Es252 = Es;
441 const double Fm257 = Fm;
442 const double Md258 = Md;
443 const double No259 = No;
444 const double Lr262 = Lr;
445
446
447 // probabilities of natural isotope abundances
448 const double Pr_H1 = 0.999885;
449 const double Pr_H2 = 0.000115;
450 const double Pr_He3 = 0.00000137;
451 const double Pr_He4 = 0.99999863;
452 const double Pr_Li6 = 0.0759;
453 const double Pr_Li7 = 0.9241;
454 const double Pr_Be9 = 1;
455 const double Pr_B10 = 0.00199;
456 const double Pr_B11 = 0.00801;
457 const double Pr_C12 = 0.9893;
458 const double Pr_C13 = 0.0107;
459 const double Pr_N14 = 0.99632;
460 const double Pr_N15 = 0.00368;
461 const double Pr_O16 = 0.99757;
462 const double Pr_O17 = 0.00038;
463 const double Pr_O18 = 0.00205;
464 const double Pr_F19 = 1;
465 const double Pr_Ne20 = 0.9048;
466 const double Pr_Ne21 = 0.0027;
467 const double Pr_Ne22 = 0.0925;
468 const double Pr_Na23 = 1;
469 const double Pr_Mg24 = 0.7899;
470 const double Pr_Mg25 = 0.1;
471 const double Pr_Mg26 = 0.1101;
472 const double Pr_Al27 = 1;
473 const double Pr_Si28 = 0.922297;
474 const double Pr_Si29 = 0.046832;
```

```
475 const double Pr_Si30 = 0.030872;
476 const double Pr_P31 = 1;
477 const double Pr_S32 = 0.9493;
478 const double Pr_S33 = 0.0076;
479 const double Pr_S34 = 0.0429;
480 const double Pr_S36 = 0.0002;
481 const double Pr_Cl35 = 0.7578;
482 const double Pr_Cl37 = 0.2422;
483 const double Pr_Ar36 = 0.003365;
484 const double Pr_Ar38 = 0.000632;
485 const double Pr_Ar40 = 0.996003;
486 const double Pr_K39 = 0.932581;
487 const double Pr_K40 = 0.000117;
488 const double Pr_K41 = 0.067302;
489 const double Pr_Ca40 = 0.96941;
490 const double Pr_Ca42 = 0.00647;
491 const double Pr_Ca43 = 0.00135;
492 const double Pr_Ca44 = 0.02086;
493 const double Pr_Ca46 = 0.00004;
494 const double Pr_Ca48 = 0.00187;
495 const double Pr_Sc45 = 1;
496 const double Pr_Ti46 = 0.0825;
497 const double Pr_Ti47 = 0.0744;
498 const double Pr_Ti48 = 0.7372;
499 const double Pr_Ti49 = 0.0541;
500 const double Pr_Ti50 = 0.0518;
501 const double Pr_V50 = 0.0025;
502 const double Pr_V51 = 0.9975;
503 const double Pr_Cr50 = 0.04345;
504 const double Pr_Cr52 = 0.83789;
505 const double Pr_Cr53 = 0.09501;
506 const double Pr_Cr54 = 0.02365;
507 const double Pr_Mn55 = 1;
508 const double Pr_Fe54 = 0.05845;
509 const double Pr_Fe56 = 0.91754;
510 const double Pr_Fe57 = 0.02119;
511 const double Pr_Fe58 = 0.00282;
512 const double Pr_Co59 = 1;
513 const double Pr_Ni58 = 0.680769;
514 const double Pr_Ni60 = 0.262231;
515 const double Pr_Ni61 = 0.011399;
516 const double Pr_Ni62 = 0.036345;
517 const double Pr_Ni64 = 0.009256;
518 const double Pr_Cu63 = 0.6917;
519 const double Pr_Cu65 = 0.3083;
520 const double Pr_Zn64 = 0.4863;
521 const double Pr_Zn66 = 0.2790;
522 const double Pr_Zn67 = 0.0410;
523 const double Pr_Zn68 = 0.1875;
524 const double Pr_Zn70 = 0.0062;
525 const double Pr_Ga69 = 0.60108;
526 const double Pr_Ga71 = 0.39892;
527 const double Pr_Ge70 = 0.2084;
528 const double Pr_Ge72 = 0.2754;
529 const double Pr_Ge73 = 0.0773;
530 const double Pr_Ge74 = 0.3628;
531 const double Pr_Ge76 = 0.0761;
532 const double Pr_As75 = 1;
533 const double Pr_Se74 = 0.0089;
534 const double Pr_Se76 = 0.0937;
535 const double Pr_Se77 = 0.0763;
536 const double Pr_Se78 = 0.2377;
537 const double Pr_Se80 = 0.4961;
538 const double Pr_Se82 = 0.0873;
539 const double Pr_Br79 = 0.5069;
540 const double Pr_Br81 = 0.4931;
541 const double Pr_Kr78 = 0.0035;
542 const double Pr_Kr80 = 0.0228;
543 const double Pr_Kr82 = 0.1158;
544 const double Pr_Kr83 = 0.1149;
545 const double Pr_Kr84 = 0.57;
546 const double Pr_Kr86 = 0.173;
547 const double Pr_Rb85 = 0.7217;
548 const double Pr_Rb87 = 0.2783;
549 const double Pr_Sr84 = 0.0056;
550 const double Pr_Sr86 = 0.0986;
551 const double Pr_Sr87 = 0.07;
552 const double Pr_Sr88 = 0.8258;
553 const double Pr_Y89 = 1;
554 const double Pr_Zr90 = 0.5145;
555 const double Pr_Zr91 = 0.1122;
556 const double Pr_Zr92 = 0.1715;
557 const double Pr_Zr94 = 0.1738;
558 const double Pr_Zr96 = 0.028;
559 const double Pr_Nb93 = 1;
560 const double Pr_Mo92 = 0.1484;
561 const double Pr_Mo94 = 0.0925;
```

```
562 const double Pr_Mo95 = 0.1592;
563 const double Pr_Mo96 = 0.1668;
564 const double Pr_Mo97 = 0.0955;
565 const double Pr_Mo98 = 0.2413;
566 const double Pr_Mo100 = 0.0963;
567 const double Pr_Tc98 = 1;
568 const double Pr_Ru96 = 0.0554;
569 const double Pr_Ru98 = 0.0187;
570 const double Pr_Ru99 = 0.1276;
571 const double Pr_Ru100 = 0.126;
572 const double Pr_Ru101 = 0.1706;
573 const double Pr_Ru102 = 0.3155;
574 const double Pr_Ru104 = 0.1862;
575 const double Pr_Rh103 = 1;
576 const double Pr_Pd102 = 0.0102;
577 const double Pr_Pd104 = 0.1114;
578 const double Pr_Pd105 = 0.2233;
579 const double Pr_Pd106 = 0.2733;
580 const double Pr_Pd108 = 0.2646;
581 const double Pr_Pd110 = 0.1172;
582 const double Pr_Ag107 = 0.51839;
583 const double Pr_Ag109 = 0.48161;
584 const double Pr_Cd106 = 0.0125;
585 const double Pr_Cd108 = 0.0089;
586 const double Pr_Cd110 = 0.1249;
587 const double Pr_Cd111 = 0.128;
588 const double Pr_Cd112 = 0.2413;
589 const double Pr_Cd113 = 0.1222;
590 const double Pr_Cd114 = 0.2873;
591 const double Pr_Cd116 = 0.0749;
592 const double Pr_In113 = 0.0429;
593 const double Pr_In115 = 0.9571;
594 const double Pr_Sn112 = 0.0097;
595 const double Pr_Sn114 = 0.0066;
596 const double Pr_Sn115 = 0.0034;
597 const double Pr_Sn116 = 0.1454;
598 const double Pr_Sn117 = 0.0768;
599 const double Pr_Sn118 = 0.2422;
600 const double Pr_Sn119 = 0.0859;
601 const double Pr_Sn120 = 0.3258;
602 const double Pr_Sn122 = 0.0463;
603 const double Pr_Sn124 = 0.0579;
604 const double Pr_Sb121 = 0.5721;
605 const double Pr_Sb123 = 0.4279;
606 const double Pr_Te120 = 0.0009;
607 const double Pr_Te122 = 0.0255;
608 const double Pr_Te123 = 0.0089;
609 const double Pr_Te124 = 0.0474;
610 const double Pr_Te125 = 0.0707;
611 const double Pr_Te126 = 0.1884;
612 const double Pr_Te128 = 0.3174;
613 const double Pr_Te130 = 0.3408;
614 const double Pr_I127 = 1;
615 const double Pr_Xe124 = 0.0009;
616 const double Pr_Xe126 = 0.0009;
617 const double Pr_Xe128 = 0.0192;
618 const double Pr_Xe129 = 0.2644;
619 const double Pr_Xe130 = 0.0408;
620 const double Pr_Xe131 = 0.2118;
621 const double Pr_Xe132 = 0.2689;
622 const double Pr_Xe134 = 0.1044;
623 const double Pr_Xe136 = 0.0887;
624 const double Pr_Cs133 = 1;
625 const double Pr_Ba130 = 0.00106;
626 const double Pr_Ba132 = 0.00101;
627 const double Pr_Ba134 = 0.02417;
628 const double Pr_Ba135 = 0.06592;
629 const double Pr_Ba136 = 0.07854;
630 const double Pr_Ba137 = 0.11232;
631 const double Pr_Ba138 = 0.71698;
632 const double Pr_La138 = 0.0009;
633 const double Pr_La139 = 0.9991;
634 const double Pr_Ce136 = 0.00185;
635 const double Pr_Ce138 = 0.00251;
636 const double Pr_Ce140 = 0.8845;
637 const double Pr_Ce142 = 0.11114;
638 const double Pr_Pr141 = 1;
639 const double Pr_Nd142 = 0.272;
640 const double Pr_Nd143 = 0.122;
641 const double Pr_Nd144 = 0.238;
642 const double Pr_Nd145 = 0.083;
643 const double Pr_Nd146 = 0.172;
644 const double Pr_Nd148 = 0.057;
645 const double Pr_Nd150 = 0.056;
646 const double Pr_Pm145 = 1;
647 const double Pr_Sm144 = 0.0307;
648 const double Pr_Sm147 = 0.1499;
```

```
649 const double Pr_Sm148 = 0.1124;
650 const double Pr_Sm149 = 0.1382;
651 const double Pr_Sm150 = 0.0738;
652 const double Pr_Sm152 = 0.2675;
653 const double Pr_Sm154 = 0.2275;
654 const double Pr_Eu151 = 0.4781;
655 const double Pr_Eu153 = 0.5219;
656 const double Pr_Gd152 = 0.002;
657 const double Pr_Gd154 = 0.0218;
658 const double Pr_Gd155 = 0.148;
659 const double Pr_Gd156 = 0.2047;
660 const double Pr_Gd157 = 0.1565;
661 const double Pr_Gd158 = 0.2484;
662 const double Pr_Gd160 = 0.2186;
663 const double Pr_Tb159 = 1;
664 const double Pr_Dy156 = 0.0006;
665 const double Pr_Dy158 = 0.001;
666 const double Pr_Dy160 = 0.0234;
667 const double Pr_Dy161 = 0.1891;
668 const double Pr_Dy162 = 0.2551;
669 const double Pr_Dy163 = 0.249;
670 const double Pr_Dy164 = 0.2818;
671 const double Pr_Ho165 = 1;
672 const double Pr_Er162 = 0.0014;
673 const double Pr_Er164 = 0.0161;
674 const double Pr_Er166 = 0.3361;
675 const double Pr_Er167 = 0.2293;
676 const double Pr_Er168 = 0.2678;
677 const double Pr_Er170 = 0.1493;
678 const double Pr_Tm169 = 1;
679 const double Pr_Yb168 = 0.0013;
680 const double Pr_Yb170 = 0.0304;
681 const double Pr_Yb171 = 0.1428;
682 const double Pr_Yb172 = 0.2183;
683 const double Pr_Yb173 = 0.1613;
684 const double Pr_Yb174 = 0.3183;
685 const double Pr_Yb176 = 0.1276;
686 const double Pr_Lu175 = 0.9741;
687 const double Pr_Lu176 = 0.0259;
688 const double Pr_Hf174 = 0.0016;
689 const double Pr_Hf176 = 0.0526;
690 const double Pr_Hf177 = 0.186;
691 const double Pr_Hf178 = 0.2728;
692 const double Pr_Hf179 = 0.1362;
693 const double Pr_Hf180 = 0.3508;
694 const double Pr_Ta180 = 0.00012;
695 const double Pr_Ta181 = 0.99988;
696 const double Pr_W180 = 0.0012;
697 const double Pr_W182 = 0.265;
698 const double Pr_W183 = 0.1431;
699 const double Pr_W184 = 0.3064;
700 const double Pr_W186 = 0.2843;
701 const double Pr_Re185 = 0.374;
702 const double Pr_Re187 = 0.626;
703 const double Pr_Os184 = 0.0002;
704 const double Pr_Os186 = 0.0159;
705 const double Pr_Os187 = 0.0196;
706 const double Pr_Os188 = 0.1324;
707 const double Pr_Os189 = 0.1615;
708 const double Pr_Os190 = 0.2626;
709 const double Pr_Os192 = 0.4078;
710 const double Pr_Ir191 = 0.373;
711 const double Pr_Ir193 = 0.627;
712 const double Pr_Pt190 = 0.00014;
713 const double Pr_Pt192 = 0.00782;
714 const double Pr_Pt194 = 0.32967;
715 const double Pr_Pt195 = 0.33832;
716 const double Pr_Pt196 = 0.25242;
717 const double Pr_Pt198 = 0.07163;
718 const double Pr_Au197 = 1;
719 const double Pr_Hg196 = 0.0015;
720 const double Pr_Hg198 = 0.0997;
721 const double Pr_Hg199 = 0.1687;
722 const double Pr_Hg200 = 0.231;
723 const double Pr_Hg201 = 0.1318;
724 const double Pr_Hg202 = 0.2986;
725 const double Pr_Hg204 = 0.0687;
726 const double Pr_Tl203 = 0.29524;
727 const double Pr_Tl205 = 0.70476;
728 const double Pr_Pb204 = 0.014;
729 const double Pr_Pb206 = 0.241;
730 const double Pr_Pb207 = 0.221;
731 const double Pr_Pb208 = 0.524;
732 const double Pr_Bi209 = 1;
733 const double Pr_Po209 = 1;
734 const double Pr_At210 = 1;
735 const double Pr_Rn222 = 1;
```

```
736 const double Pr_Fr223 = 1;
737 const double Pr_Ra226 = 1;
738 const double Pr_Ac227 = 1;
739 const double Pr_Th232 = 1;
740 const double Pr_Pa231 = 1;
741 const double Pr_U234 = 0.000055;
742 const double Pr_U235 = 0.0072;
743 const double Pr_U238 = 0.992745;
744 const double Pr_Np237 = 1;
745 const double Pr_Pu244 = 1;
746 const double Pr_Am243 = 1;
747 const double Pr_Cm247 = 1;
748 const double Pr_Bk247 = 1;
749 const double Pr_Cf251 = 1;
750 const double Pr_Es252 = 1;
751 const double Pr_Fm257 = 1;
752 const double Pr_Md258 = 1;
753 const double Pr_No259 = 1;
754 const double Pr_Lr262 = 1;
755
756
760 class cPeriodicTableMap {
761
762     map<string, double> table;
763     vector<string> adductElements;
764
765 public:
766
767
771     cPeriodicTableMap();
772
773
779     double &operator[](string& element);
780
781
787     int count(string& element);
788
789
794     map<string, double>& getElements();
795
796
801     vector<string>& getAdductElements();
802
803 };
804
805
809 extern cPeriodicTableMap periodictablemap;
810
811
815 struct isotope {
816
820     double mass;
821
822
826     double probability;
827
828
832     isotope() {
833         clear();
834     }
835
836
840     void clear() {
841         mass = 0;
842         probability = 0;
843     }
844
845 };
846
847
851 class cIsotopeMap {
852
853     map< string, map<string, isotope> > table;
854
855 public:
856
857
861     cIsotopeMap();
862
863
869     map<string, isotope> &operator[](string& element);
870
871
877     int count(string& element);
878
879 };
880
```

```
881
885 extern cIsotopeMap isotopemap;
886
887
888 const double H2O = 2 * H + O;
889 const double NH3 = 3 * H + N;
890 const double CO = C + O;
891
892 const double A_ION = H - e - C - O;
893 const double B_ION = H - e;
894 const double C_ION = H - e + NH3;
895 const double X_ION = H - e + 2 * O + C;
896 const double Y_ION = H - e + H2O;
897 const double Z_ION = H - e + O - N - H;
898
899 const double PRECURSOR_ION = H - e + H2O;
900 const double PRECURSOR_ION_CYCLIC = H - e;
901
902 const double L0H_ION = H - e - 2 * H;
903 const double L1H_ION = H - e;
904 const double L2H_ION = H - e + 2 * H;
905 const double R1H_ION = H - e;
906 const double R2H_ION = H - e + 2 * H;
907 const double L1OH_ION = H - e + O;
908 const double L2OH_ION = H - e + H2O;
909 const double R1OH_ION = H - e + O;
910 const double R2OH_ION = H - e + H2O;
911
912 const double PRECURSOR_ION_LINEAR_POLYKETIDE_H_H = H - e + 2 * H;
913 const double PRECURSOR_ION_LINEAR_POLYKETIDE_H_OH = H - e + H2O;
914 const double PRECURSOR_ION_LINEAR_POLYKETIDE_OH_OH = H - e + 2 * H + 2 * O;
915
916
920 Q_DECLARE_METATYPE(ePeptideType);
921
922
929 double charge(double neutralmass, int charge);
930
931
938 double uncharge(double mzratio, int charge);
939
940
944 enum eFragmentIonType {
945     a_ion = 1,
946     b_ion,
947     c_ion,
948     x_ion,
949     y_ion,
950     z_ion,
951     precursor_ion,
952     cyclic_precursor_ion,
953     l0h_ion,
954     l1h_ion,
955     l2h_ion,
956     r1h_ion,
957     r2h_ion,
958     l1oh_ion,
959     l2oh_ion,
960     r1oh_ion,
961     r2oh_ion,
962     linear_polyketide_precursor_ion_h_h,
963     linear_polyketide_precursor_ion_h_oh,
964     linear_polyketide_precursor_ion_oh_oh,
965     cyclic_polyketide_precursor_ion,
966     fragmentIonTypeEnd
967 };
968
969
973 Q_DECLARE_METATYPE(vector<eFragmentIonType>);
974
975
979 struct cFragmentIonType {
980
984     string name;
985
986
990     string summary;
991
992
996     map<string, int> summarymap;
997
998
1002     double massdifference;
1003
1004
1008     bool nterminal;
1009
```

```
1010
1014     bool cterminal;
1015
1016
1020     bool positive;
1021
1022
1026     cFragmentIonType() {
1027         clear();
1028     }
1029
1030
1034     void clear() {
1035         name = "";
1036         summary = "";
1037         summarymap.clear();
1038         massdifference = 0;
1039         nterminal = false;
1040         cterminal = false;
1041         positive = true;
1042     }
1043
1044
1049     void store(ofstream& os);
1050
1051
1056     void load(ifstream& is);
1057
1058 };
1059
1060
1064 Q_DECLARE_METATYPE(vector<cFragmentIonType>);
1065
1066
1070 class cFragmentIons {
1071
1072     map<eFragmentIonType, cFragmentIonType> fragmentions;
1073
1074 public:
1075
1076
1080     cFragmentIons();
1081
1082
1089     cFragmentIons(bool cyclicnterminus, bool cyclicctterminus, string& precursoradduct);
1090
1091
1097     cFragmentIonType &operator[](eFragmentIonType iontype);
1098
1099
1106     void recalculateFragments(bool cyclicnterminus, bool cyclicctterminus, string& precursoradduct);
1107
1108
1113     void store(ofstream& os);
1114
1115
1120     void load(ifstream& is);
1121
1122 };
1123
1124
1128 Q_DECLARE_METATYPE(cFragmentIons);
1129
1130
1135 void initializeFragmentIonsForDeNovoGraphOfCyclicPeptides(vector<eFragmentIonType>& fragmentions);
1136
1137
1142 void initializeFragmentIonsForDeNovoGraphOfLinearPeptides(vector<eFragmentIonType>& fragmentions);
1143
1144
1149 void initializeFragmentIonsForDeNovoGraphOfTPeptides(vector<eFragmentIonType>& fragmentions);
1150
1151
1156 void initializeFragmentIonsForDeNovoGraphOfBranchCyclicPeptides(vector<eFragmentIonType>&
      fragmentions);
1157
1158
1163 void initializeFragmentIonsForDeNovoGraphOfLinearPolyketide(vector<eFragmentIonType>& fragmentions);
1164
1165
1170 void initializeFragmentIonsForDeNovoGraphOfCyclicPolyketide(vector<eFragmentIonType>& fragmentions);
1171
1172
1176 struct neutralLoss {
1177
1181     string summary;
1182
```

```
1183
1187     map<string, int> summarymap;
1188
1189
1193     double massdifference;
1194
1195
1199     neutralLoss() {
1200         clear();
1201     }
1202
1203
1207     void clear() {
1208         summary = "";
1209         summarymap.clear();
1210         massdifference = 0;
1211     }
1212
1213
1218     void store(ofstream& os);
1219
1220
1225     void load(ifstream& is);
1226
1227 };
1228
1229
1233 class cDefaultNeutralLosses {
1234
1235     vector<string> neutrallosses;
1236
1237 public:
1238
1242     cDefaultNeutralLosses();
1243
1244
1249     vector<string>& getNeutralLosses() {
1250         return neutrallosses;
1251     }
1252
1253 };
1254
1255
1259 extern cDefaultNeutralLosses defaultneutrallosses;
1260
1261
1270 int loadModificationsFromPlainTextStream(ifstream &stream, vector<cFragmentIonType>& modifications,
       string& errormessage, bool ignoreerrors);
1271
1272
1278 void storeModificationsToPlainTextStream(ofstream &stream, vector<cFragmentIonType>& modifications);
1279
1280
1281 #endif
1282
```

17.15 core/cGlobalPreferences.h File Reference

The representation of a set of peptide sequence candidates.

```
#include "core/utilities.h"
#include "core/cFragmentIons.h"
#include "core/cSummaryFormula.h"
#include <QString>
#include <QSettings>
#include <QFile>
```

Classes

- struct cIonType

  The struct for definition of a ion type.

- class cGlobalPreferences

  The class storing global preferences.

17.15.1 Detailed Description

The representation of a set of peptide sequence candidates.

17.16 cGlobalPreferences.h

Go to the documentation of this file.

```
1
7 #ifndef _CGLOBALPREFERENCES_H
8 #define _CGLOBALPREFERENCES_H
9
10 #include "core/utilities.h"
11 #include "core/cFragmentIons.h"
12 #include "core/cSummaryFormula.h"
13 #include <QString>
14 #include <QSettings>
15 #include <QFile>
16
17
21 struct cIonType {
22
26     string name;
27
28
32     string formula;
33
34
38     double massdifference;
39
40
44     bool positive;
45
46
50     int multiplier;
51
52
56     cIonType() {
57         clear();
58     }
59
60
64     void clear() {
65         name = "[M+H]+";
66         formula = "H+";
67         massdifference = H - e;
68         positive = true;
69         multiplier = 1;
70     }
71
72
77     void store(ofstream& os) {
78         storeString(name, os);
79         storeString(formula, os);
80         os.write((char *)&massdifference, sizeof(double));
81         os.write((char *)&positive, sizeof(bool));
82         os.write((char *)&multiplier, sizeof(int));
83     }
84
85
90     void load(ifstream& is) {
91         loadString(name, is);
92         loadString(formula, is);
93         is.read((char *)&massdifference, sizeof(double));
94         is.read((char *)&positive, sizeof(bool));
95         is.read((char *)&multiplier, sizeof(int));
96     }
97
98 };
99
100
104 class cGlobalPreferences {
105
106     QString settingsfile;
107
108 public:
109
110     vector<cIonType> customions;
111
112     int linebafprocessingmethod;
113     int profilebafprocessingmethod;
114
115     int rawdataprocessingmethod;
116
117     QString settingsdefaultdir;
```

```
118     QString peaklistsdefaultdir;
119     QString blocksdefaultdir;
120     QString sequencesdefaultdir;
121     QString modificationsdefaultdir;
122     QString openopticalimagedir;
123     QString openhistologyimagedir;
124     QString openmicroscopyimagedir;
125     QString resultsdefaultdir;
126     QString exportcsvdefaultdir;
127     QString exporthtmldefaultdir;
128     QString exportimagedefaultdir;
129
130     QString bookmarkname1;
131     QString bookmarkurl1;
132
133     QString bookmarkname2;
134     QString bookmarkurl2;
135
136     QString bookmarkname3;
137     QString bookmarkurl3;
138
139     QString bookmarkname4;
140     QString bookmarkurl4;
141
142     QString bookmarkname5;
143     QString bookmarkurl5;
144
145     QString bookmarkname6;
146     QString bookmarkurl6;
147
148     QString bookmarkname7;
149     QString bookmarkurl7;
150
151     QString bookmarkname8;
152     QString bookmarkurl8;
153
154     QString bookmarkname9;
155     QString bookmarkurl9;
156
157     QString bookmarkname10;
158     QString bookmarkurl10;
159
160     bool viewbookmarks;
161
162
166     cGlobalPreferences();
167
168
172     void setDefaultIonTypes();
173
174
178     void setDefaultDataProcessingMethods();
179
180
184     void setDefaultDirectories();
185
186
190     void setDefaultURLs();
191
192
196     void loadSettings();
197
198
202     void saveSettings();
203
204
205 };
206
207
208 #endif
209
```

17.17 core/clmzML.h File Reference

imzML parser.

```
#include <xercesc/parsers/XercesDOMParser.hpp>
#include <xercesc/dom/DOM.hpp>
#include <string>
#include <vector>
#include <sstream>
#include <QFile>
```

```
#include <QCryptographicHash>
#include "core/cMzML.h"
```

Classes

- struct cImzMLItem

  A parsed imzML item.
- class cImzML

  imzML parser.

17.17.1 Detailed Description

imzML parser.

17.18 cImzML.h

Go to the documentation of this file.

```
1
7 #ifndef _CIMZML_H
8 #define _CIMZML_H
9
10
11 #include <xercesc/parsers/XercesDOMParser.hpp>
12 #include <xercesc/dom/DOM.hpp>
13 #include <string>
14 #include <vector>
15 #include <sstream>
16 #include <QFile>
17 #include <QCryptographicHash>
18
19 #include "core/cMzML.h"
20
21 using namespace std;
22 using namespace xercesc;
23
24
25 class cPeaksList;
26
27
31 struct cImzMLItem {
32
36     int x;
37
38
42     int y;
43
44
48     long long unsigned mzstart;
49
50
54     long long unsigned mzlength;
55
56
60     long long unsigned intensitystart;
61
62
66     long long unsigned intensitylength;
67
68
72     string title;
73
74
78     cImzMLItem() {
79         x = 0;
80         y = 0;
81         mzstart = 0;
82         mzlength = 0;
83         intensitystart = 0;
84         intensitylength = 0;
85         title = "";
86     }
87
88 };
89
90
91
```

```
95 class cImzML {
96
97     XercesDOMParser* parser;
98     DOMDocument* document;
99
100     bool profilespectra;
101     bool use_64bit_float_mz_precision;
102     bool use_64bit_float_intensity_precision;
103     vector<cImzMLItem> imzmlitems;
104
105 public:
106
107
111     cImzML();
112
113
117     ~cImzML();
118
119
130     int parse(string& filename, int& defaultmaxx, int& defaultmaxy, int& defaultpixelsizex, int&
      defaultpixelsizey, eVendorType& vendor);
131
132
138     void updateRawDataToPeakList(vector<cPeaksList>& peaklists, string& convertedibdfilename);
139
140
145     vector<cImzMLItem>& getItems();
146
147
152     bool hasProfileSpectra();
153
154
159     bool use64BitMzPrecision();
160
161
166     bool use64BitIntensityPrecision();
167
168
173     void write(string& filename);
174
175 };
176
177
178 #endif
179
```

17.19 core/clsotopePatternCache.h File Reference

Isotope pattern cache (LRU cache is implemented).

```
#include <QMutex>
#include <unordered_map>
#include <list>
#include "core/utilities.h"
#include "core/cPeaksList.h"
```

Classes

- class clsotopePatternCache

  Isotope pattern cache (LRU cache is implemented).

17.19.1 Detailed Description

Isotope pattern cache (LRU cache is implemented).

17.20 clsotopePatternCache.h

Go to the documentation of this file.

```
1
7 #ifndef _CISOTOPEPATTERNCACHE_H
8 #define _CISOTOPEPATTERNCACHE_H
9
10
```

```
11 #include <QMutex>
12 #include <unordered_map>
13 #include <list>
14 #include "core/utilities.h"
15 #include "core/cPeaksList.h"
16
17
21 class cIsotopePatternCache :  public QMutex {
22
23     unordered_map<string, list<pair<string, cPeaksList> >::iterator > cache_map;
24     list<pair<string, cPeaksList> > cache_list;
25
26     int maxnumberofitems;
27
28 public:
29
30
35     cIsotopePatternCache(int size = 10000);
36
37
41     ~cIsotopePatternCache();
42
43
50     bool get(string& formula, cPeaksList& isotopepattern);
51
52
58     void put(string& formula, cPeaksList& isotopepattern);
59
60
64     void clear();
65
66 };
67
68
69 #endif
70
```

17.21 core/cMzML.h File Reference

mzML parser.

```
#include <xercesc/dom/DOM.hpp>
#include <xercesc/parsers/XercesDOMParser.hpp>
#include <xercesc/util/XMemory.hpp>
#include <xercesc/util/Base64.hpp>
#include <string>
#include <vector>
#include <sstream>
#include "core/utilities.h"
```

Classes

- class cMzML

  mzML parser.

Functions

- bool compareElementTagName (DOMElement *element, const char *name)

  Compare element's tag name with a string.
- string getAttribute (DOMElement *element, const char *name)

  Get an attribute.
- void setAttribute (DOMElement *element, const char *name, const char *value)

  Set an attribute.

17.21.1 Detailed Description

mzML parser.

17.21.2 Function Documentation

17.21.2.1 compareElementTagName()

```
bool compareElementTagName (
            DOMElement * element,
            const char * name )
```

Compare element's tag name with a string.

Parameters

element	pointer to element
name	string to be compared

Return values

bool	true if the strings are equal, false otherwise.

17.21.2.2 getAttribute()

```
string getAttribute (
            DOMElement * element,
            const char * name )
```

Get an attribute.

Parameters

element	pointer to element
name	attribute name

Return values

string	attribute value

17.21.2.3 setAttribute()

```
void setAttribute (
            DOMElement * element,
            const char * name,
            const char * value )
```

Set an attribute.

Parameters

element	pointer to element
name	attribute name
value	attribute value

17.22 cMzML.h

Go to the documentation of this file.

```
1
7 #ifndef _CMZML_H
8 #define _CMZML_H
9
10
11 #include <xercesc/dom/DOM.hpp>
12 #include <xercesc/parsers/XercesDOMParser.hpp>
13 #include <xercesc/util/XMemory.hpp>
14 #include <xercesc/util/Base64.hpp>
15 #include <string>
16 #include <vector>
17 #include <sstream>
18
19 #include "core/utilities.h"
20
21
22 using namespace std;
23 using namespace xercesc;
24
25
26 class cPeaksList;
27 class cMainThread;
28
29
36 bool compareElementTagName(DOMElement* element, const char* name);
37
38
45 string getAttribute(DOMElement* element, const char* name);
46
47
54 void setAttribute(DOMElement* element, const char* name, const char* value);
55
56
60 class cMzML {
61
62     XercesDOMParser* parser;
63     DOMDocument* document;
64
65     bool profilespectra;
66
67
68 public:
69
70
74     cMzML();
75
76
80     ~cMzML();
81
82
93     int parse(string& filename, vector<cPeaksList>& peaklists, int profilespectrumid, eModeType mode,
      cMainThread* os, bool& terminatecomputation);
94
95
100     bool hasProfileSpectra();
101
102
103 };
104
105
106 #endif
107
```

17.23 core/cParameters.h File Reference

Definitions of program parameters.

```
#include <iostream>
#include <fstream>
#include <set>
#include "core/utilities.h"
#include "core/cGlobalPreferences.h"
#include "core/cPeakListSeries.h"
#include "core/cBricksDatabase.h"
#include "core/cSequenceDatabase.h"
```

Classes

- class cParameters

  The class storing parameters of the application.
- struct hintStructure

  An auxiliary structure for search hints.

Enumerations

- enum ePeakListFileFormat {
  txt = 0 , **mgf** , **mzML** , **mzXML** ,
  baf , **raw** , **dat** , **imzML** ,
  mis , **ser** }

  Peak list file formats supported by the application.
- enum eScoreType {
  number_of_matched_peaks = 0 , **sum_of_relative_intensities** = 1 , **number_of_b_ions** = 2 , **number_of_y_ions** = 3 ,
  number_of_b_and_y_ions = 4 , **weighted_ratio_of_matched_peaks** = 5 , **cosine_similarity** = 6 }

  The types of scores between an experimental and a theoretical spectrum.

Functions

- bool checkSeniorRules (vector< int > &combarray, vector< int > &valences, int maxcomponents)

  Check Senior's filtering rules.
- bool checkAdvancedFilteringRules (bool noratiocheck, double sumofmasses, vector< int > &countsofelements, vector< string > &namesofelements)

  Check advanced filtering rules.
- **Q_DECLARE_METATYPE** (cParameters)

  Register cParameters by Qt.
- bool **compareHints** (const hintStructure &a, const hintStructure &b)

  An auxiliary comparator for search hints.
- void searchHintForDeNovo (double mzratio, vector< hintStructure > &experimentalpeaks, double fragmentmasserrortolerance, set< int > &hintset)

  Get IDs of spectra in which the theoretical m/z ratio can be found.

17.23.1 Detailed Description

Definitions of program parameters.

17.23.2 Function Documentation

17.23.2.1 checkAdvancedFilteringRules()

```
bool checkAdvancedFilteringRules (
            bool noratiocheck,
            double sumofmasses,
            vector< int > & countsofelements,
            vector< string > & namesofelements )
```

Check advanced filtering rules.
noratiocheck if true, N/O ratio check is enabled
sumofmasses sum of masses
countsofelements counts of elements
namesofelelements names of elements

Return values

bool	true, if advanced filtering rules are NOT ok

17.23.2.2 checkSeniorRules()

```
bool checkSeniorRules (
            vector< int > & combarray,
            vector< int > & valences,
            int maxcomponents )
```

Check Senior's filtering rules.

Parameters

combarray	current combination of elements
valences	vector of valences
maxcomponents	maximum number of components in a graph

Return values

bool	true, if Senior's rules are ok

17.23.2.3 searchHintForDeNovo()

```
void searchHintForDeNovo (
            double mzratio,
            vector< hintStructure > & experimentalpeaks,
            double fragmentmasserrortolerance,
            set< int > & hintset )
```

Get IDs of spectra in which the theoretical m/z ratio can be found.

Parameters

mzratio	theoretical m/z ratio
experimentalpeaks	a sorted vector of hint structures
fragmentmasserrortolerance	m/z error tolerance
hintset	output set of spectra IDs

17.24 cParameters.h

Go to the documentation of this file.

```
1
7 #ifndef _CPARAMETERS_H
8 #define _CPARAMETERS_H
9
10 #include <iostream>
11 #include <fstream>
12 #include <set>
13 #include "core/utilities.h"
14 #include "core/cGlobalPreferences.h"
15 #include "core/cPeakListSeries.h"
16 #include "core/cBricksDatabase.h"
17 #include "core/cSequenceDatabase.h"
18
```

```
19
20 class cMainThread;
21
22
23 using namespace std;
24 using namespace boost;
25
26
30 enum ePeakListFileFormat {
31     txt = 0,
32     mgf,
33     mzML,
34     mzXML,
35     baf,
36     raw,
37     dat,
38     imzML,
39     mis,
40     ser
41 };
42
43
47 enum eScoreType {
48     number_of_matched_peaks = 0,
49     sum_of_relative_intensities = 1,
50     number_of_b_ions = 2,
51     number_of_y_ions = 3,
52     number_of_b_and_y_ions = 4,
53     weighted_ratio_of_matched_peaks = 5,
54     cosine_similarity = 6
55 };
56
57
65 bool checkSeniorRules(vector<int>& combarray, vector<int>& valences, int maxcomponents);
66
67
76 bool checkAdvancedFilteringRules(bool noratiocheck, double sumofmasses, vector<int>& countsofelements,
       vector<string>& namesofelements);
77
78
82 class cParameters {
83
84     cMainThread* os;
85
86     void fixIntensities(cPeaksList& centroidspectrum, cPeaksList& profilespectrum);
87
88     void pushLossItem(vector<neutralLoss>& lossitems, string lossstr);
89
90     //double getMassAndCounts(vector<int>& combarray, vector<int>& countsofelements, vector<double>&
       massesofelements);
91
92
93 public:
94
95
99     cFragmentIons iondefinitions;
100
101
105     ePeptideType peptidetype;
106
107
111     vector<string> peaklistfilenames;
112
113
117     string originalpeaklistfilenames;
118
119
123     bool useprofiledata;
124
125
129     bool convertprofiledata;
130
131
135     string profiledatafilename;
136
137
141     int linebafprocessing;
142
143
147     vector<ePeakListFileFormat> peaklistfileformats;
148
149
153     vector<cPeakListSeries> peaklistseriesvector;
154
155
159     int scannumber;
160
```

```
161
165     double precursormass;
166
167
171     string precursoradduct;
172
173
177     vector<string> metaladducts;
178
179
183     double precursormasserrortolerance;
184
185
189     int precursorcharge;
190
191
195     double fragmentmasserrortolerance;
196
197
201     double minimumrelativeintensitythreshold;
202
203
207     unsigned minimumabsoluteintensitythreshold;
208
209
213     double minimummz;
214
215
219     double maximummz;
220
221
225     double minimumrt;
226
227
231     double maximumrt;
232
233
237     double fwhm;
238
239
243     bool enableratio54Fe56Fe;
244
245
249     double minratio54Fe56Fe;
250
251
255     double maxratio54Fe56Fe;
256
257
261     bool enableratio60Ni58Ni;
262
263
267     double minratio60Ni58Ni;
268
269
273     double maxratio60Ni58Ni;
274
275
279     bool enableratio62Ni58Ni;
280
281
285     double minratio62Ni58Ni;
286
287
291     double maxratio62Ni58Ni;
292
293
297     bool enableratio65Cu63Cu;
298
299
303     double minratio65Cu63Cu;
304
305
309     double maxratio65Cu63Cu;
310
311
315     bool enableratio66Zn64Zn;
316
317
321     double minratio66Zn64Zn;
322
323
327     double maxratio66Zn64Zn;
328
329
333     bool enableratio67Zn64Zn;
334
```

```
335
339     double minratio67Zn64Zn;
340
341
345     double maxratio67Zn64Zn;
346
347
351     bool enableratio68Zn64Zn;
352
353
357     double minratio68Zn64Zn;
358
359
363     double maxratio68Zn64Zn;
364
365
369     string bricksdatabasefilename;
370
371
375     cBricksDatabase bricksdatabase;
376
377
381     int maximumbricksincombinationbegin;
382
383
387     int maximumbricksincombinationmiddle;
388
389
393     int maximumbricksincombinationend;
394
395
399     int maximumbricksincombination;
400
401
405     double maximumcumulativemass;
406
407
411     string modificationsfilename;
412
413
417     int maximumnumberofthreads;
418
419
423     eModeType mode;
424
425
429     eScoreType scoretype;
430
431
435     int maximumcombinedlosses;
436
437
441     int maximumcombinedelements;
442
443
447     //bool clearhitswithoutparent;
448
449
453     bool reportunmatchedtheoreticalpeaks;
454
455
459     bool generateisotopepattern;
460
461
465     int minimumpatternsize;
466
467
471     int minimumfeaturesize;
472
473
477     int minimumiontypes;
478
479
483     bool basicformulacheck;
484
485
489     bool advancedformulacheck;
490
491
495     bool noratiocheck;
496
497
501     bool calculatefdrs;
502
503
507     double minimumannotationintensityrelative;
508
```

```
509
513     unsigned minimumannotationintensityabsolute;
514
515
519     double mzdifftolerance;
520
521
525     double intensitytolerance;
526
527
531     bool cyclicnterminus;
532
533
537     bool cyclicctterminus;
538
539
543     bool internalfragments;
544
545
549     bool enablescrambling;
550
551
555     bool similaritysearch;
556
557
561     bool regularblocksorder;
562
563
567     int hitsreported;
568
569
573     string sequencetag;
574
575
579     string originalsequencetag;
580
581
585     string searchedsequence;
586
587
591     string originalsearchedsequence;
592
593
597     string searchedsequenceNtermmodif;
598
599
603     string searchedsequenceCtermmodif;
604
605
609     string searchedsequenceTmodif;
610
611
615     string searchedsequenceformula;
616
617
621     int maximumnumberofcandidates;
622
623
627     int blindedges;
628
629
633     string sequencedatabasefilename;
634
635
639     cSequenceDatabase sequencedatabase;
640
641
645     vector<cFragmentIonType> searchedmodifications;
646
647
651     vector<eFragmentIonType> fragmentionsfordenovograph;
652
653
657     vector<cIonType> ionsfortheoreticalspectraMS1;
658
659
663     vector<eFragmentIonType> ionsfortheoreticalspectraMS2;
664
665
669     vector<neutralLoss> neutrallossesdefinitions;
670
671
675     vector<neutralLoss> originalneutrallossesdefinitions;
676
677
681     vector<int> neutrallossesfortheoreticalspectra;
682
```

```
683
687     int numberofgeneratedneutrallosses;
688
689
693     vector<int> originalneutrallossesfortheoreticalspectra;
694
695
699     vector<neutralLoss> originalelementsdefinitions;
700
701
705     vector<int> originalelementsfortheoreticalspectra;
706
707
711     vector<string> peakidtodesc;
712
713
717     vector<string> isotopeformulaidtodesc;
718
719
723     map<string, int> pchemresults;
724
725
729     int defaultmaxx;
730
731
735     int defaultmaxy;
736
737
741     int defaultpixelsizex;
742
743
747     int defaultpixelsizey;
748
749
753     eVendorType vendor;
754
755
759     cParameters();
760
761
765     void clear();
766
767
773     int checkAndPrepare(bool& terminatecomputation);
774
775
781     int prepareLossesAndCompounds(bool& terminatecomputation);
782
783
793     bool checkModifications(cSequence& sequence, int& startmodifid, int& endmodifid, int& middlemodifid,
    string& errormessage);
794
795
800     string printToString();
801
802
807     void setOutputStream(cMainThread& os);
808
809
813     void updateFragmentDefinitions();
814
815
822     int calculateNeutralLosses(bool& terminatecomputation, string& errormessage);
823
824
831     int generateCompounds(bool& terminatecomputation, string& errormessage);
832
833
838     void store(ofstream& os);
839
840
848     void load(ifstream& is, int fileversionpart1, int fileversionpart2, int fileversionpart3);
849
850
851 };
852
853
857 Q_DECLARE_METATYPE(cParameters);
858
859
863 struct hintStructure {
864
868     double mz;
869
870
874     int id;
875
```

```
876 };
877
878
882 bool compareHints(const hintStructure& a, const hintStructure& b);
883
884
885
893 void searchHintForDeNovo(double mzratio, vector<hintStructure>& experimentalpeaks, double
      fragmentmasserrortolerance, set<int>& hintset);
894
895
896 #endif
```

17.25 core/cPeak.h File Reference

The representation of a peak in a mass spectrum.

```
#include <string>
#include <fstream>
#include "core/cFragmentIons.h"
#include "core/cSummaryFormula.h"
```

Classes

- struct cPeak

  A structure representing a peak in a mass spectrum.

17.25.1 Detailed Description

The representation of a peak in a mass spectrum.

17.26 cPeak.h

Go to the documentation of this file.

```
1
7 #ifndef _CPEAK_H
8 #define _CPEAK_H
9
10 #include <string>
11 #include <fstream>
12 #include "core/cFragmentIons.h"
13 #include "core/cSummaryFormula.h"
14
15
16 using namespace std;
17
18
22 struct cPeak {
23
27     cSummaryFormula formula;
28
32     string isotopeformula;
33
37     int isotopeformulaid;
38
42     double mzratio;
43
44
48     double relativeintensity;
49
50
54     double absoluteintensity;
55
56
60     string description;
61
62
66     int descriptionid;
67
68
72     double matchedmz;
73
74
```

```
78      double matchedrelativeintensity;
79
80
84      double matchedabsoluteintensity;
85
86
90      double matchedppm;
91
92
96      eFragmentIonType iontype;
97
98
102      int neutrallosstype;
103
104
108      int matched;
109
110
114      int matchedid;
115
116
120      int charge;
121
122
126      int rotationid;
127
128
132      int seriesid;
133
134
138      int groupid;
139
140
144      int compoundid;
145
146
150      bool isotope;
151
152
156      bool removeme;
157
158
162      bool scrambled;
163
164
168      bool decoy;
169
170
174      int orderid;
175
176
180      cPeak();
181
182
187      cPeak(const cPeak& peak);
188
189
194      cPeak& operator=(const cPeak& peak);
195
196
200      void clear();
201
202
207      bool empty();
208
209
214      void store(ofstream& os);
215
216
221      void load(ifstream& is);
222
223
229      bool equals(cPeak& secondpeak);
230
231 };
232
233
234 #endif
```

17.27 core/cPeakListSeries.h File Reference

The representation of a series of peaklists.

```
#include <sstream>
#include "core/cPeaksList.h"
```

Classes

- class cPeakListSeries

  The class representing a series of peaklists.

17.27.1 Detailed Description

The representation of a series of peaklists.

17.28 cPeakListSeries.h

Go to the documentation of this file.

```
1
7 #ifndef _CPEAKLISTSERIES_H
8 #define _CPEAKLISTSERIES_H
9
10 #include <sstream>
11
12 #include "core/cPeaksList.h"
13
14
15 class cMainThread;
16
17
21 class cPeakListSeries {
22
23     vector<cPeaksList> peaklists;
24
25 public:
26
27
31     cPeakListSeries();
32
33
38     cPeakListSeries(const cPeakListSeries& peaklistseries);
39
40
45     cPeakListSeries& operator=(const cPeakListSeries& peaklistseries);
46
47
53     cPeaksList& operator[](int position);
54
55
59     void clear();
60
61
66     void resize(int size);
67
68
73     void addPeakList(cPeaksList& peaklist);
74
75
80     void loadFromPlainTextStream(ifstream &stream);
81
82
87     void loadFromBAFStream(ifstream &stream);
88
89
100     int loadFromProfileApexStream(string& filename, ifstream &stream, ifstream &titlestream, double
      fwhm, cMainThread* os, bool& terminatecomputation);
101
102
113     int loadFromMZMLStream(string& mzmlfilename, ifstream &mzmlstream, double fwhm, eModeType mode,
      cMainThread* os, bool& terminatecomputation);
114
115
130     int loadFromIMZMLStream(string& imzmlfilename, ifstream &ibdstream, double fwhm, int& defaultmaxx,
      int& defaultmaxy, int& defaultpixelsizex, int& defaultpixelsizey, eVendorType& vendor, cMainThread*
      os, bool& terminatecomputation);
131
132
137     void loadSpotList(ifstream &stream);
138
139
```

```
145     void loadFromMGFStream(ifstream &stream, int timeunit);
146
147
152     int size();
153
154
159     void store(ofstream& os);
160
161
169     void load(ifstream& is, int fileversionpart1, int fileversionpart2, int fileversionpart3);
170
171 };
172
173
174 #endif
```

17.29 core/cPeaksList.h File Reference

The representation of a peak list.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <algorithm>
#include "core/cPeak.h"
#include "core/cMzML.h"
#include "core/cImzML.h"
```

Classes

- class cPeaksList

  The class representing a peak list.

Functions

- bool compareRelativePeakIntensitiesDesc (const cPeak &a, const cPeak &b)

  Compare two peaks by relative intensities.
- bool isInPpmMassErrorTolerance (double experimentalmass, double theoreticalmass, double tolerance)

  Check whether two masses are inside a ppm mass error tolerance.
- double ppmError (double experimentalmass, double theoreticalmass)

  Compute a ppm error between two masses.
- bool searchHint (double mzratio, cPeaksList &experimentalpeaks, double fragmentmasserrortolerance)

  Check if an m/z ratio exists in an experimental spectrum.
- **Q_DECLARE_METATYPE** (cPeaksList)

  Register cPeaksList by Qt.

17.29.1 Detailed Description

The representation of a peak list.

17.29.2 Function Documentation

17.29.2.1 compareRelativePeakIntensitiesDesc()

```
bool compareRelativePeakIntensitiesDesc (
            const cPeak & a,
            const cPeak & b )
```

Compare two peaks by relative intensities.

Parameters

a	first peak
b	second peak

Return values

bool	true when the relative intensity of the peak *a* is bigger than the relative intensity of the peak *b*

17.29.2.2 isInPpmMassErrorTolerance()

```
bool isInPpmMassErrorTolerance (
            double experimentalmass,
            double theoreticalmass,
            double tolerance )
```

Check whether two masses are inside a ppm mass error tolerance.

Parameters

experimentalmass	first mass
theoreticalmass	second mass
tolerance	fragment mass error tolerance

Return values

bool	true when masses are inside a mass tolerance

17.29.2.3 ppmError()

```
double ppmError (
            double experimentalmass,
            double theoreticalmass )
```

Compute a ppm error between two masses.

Parameters

experimentalmass	first mass
theoreticalmass	second mass

Return values

double	ppm error

17.29.2.4 searchHint()

```
bool searchHint (
            double mzratio,
```

```
            cPeaksList & experimentalpeaks,
            double fragmentmasserrortolerance )
```

Check if an m/z ratio exists in an experimental spectrum.

Parameters

mzratio	theoretical m/z ratio
experimentalpeaks	an input experimental spectrum (sorted in ascending order)
fragmentmasserrortolerance	m/z error tolerance

Return values

true	if the theoretical m/z ratio was mathed; false otherwise

17.30 cPeaksList.h

Go to the documentation of this file.

```
1
7 #ifndef _CPEAKSLIST_H
8 #define _CPEAKSLIST_H
9
10 #include <iostream>
11 #include <fstream>
12 #include <vector>
13 #include <string>
14 #include <algorithm>
15
16 #include "core/cPeak.h"
17 #include "core/cMzML.h"
18 #include "core/cImzML.h"
19
20
21 using namespace std;
22
23
24 class cMainThread;
25
26
33 bool compareRelativePeakIntensitiesDesc(const cPeak& a, const cPeak& b);
34
35
43 bool isInPpmMassErrorTolerance(double experimentalmass, double theoreticalmass, double tolerance);
44
45
52 double ppmError(double experimentalmass, double theoreticalmass);
53
54
62 bool searchHint(double mzratio, cPeaksList& experimentalpeaks, double fragmentmasserrortolerance);
63
64
68 class cPeaksList {
69
70     vector<cPeak> peaks;
71
72     double rt;
73
74     int rtunit;
75
76     int x, y;
77
78     string title;
79
80 public:
81
82
86     cPeaksList();
87
88
93     cPeaksList(const cPeaksList& peakslist);
94
95
100     cPeaksList& operator=(const cPeaksList& peakslist);
101
102
```

```
106     void clear();
107
108
113     void attach(cPeaksList& peaklist);
114
115
120     void add(cPeak& peak);
121
122
127     void loadFromPlainTextStream(ifstream &stream);
128
129
134     void loadFromBAFStream(ifstream &stream);
135
136
144     void loadFromIBDStream(cImzMLItem& imzmlitem, ifstream &ibdstream, bool
      use_64bit_float_mz_precision, bool use_64bit_float_intensity_precision);
145
146
153     void storeToIBDStream(ofstream &ibdstream, bool use_64bit_float_mz_precision, bool
      use_64bit_float_intensity_precision);
154
155
161     void loadFromMGFStream(ifstream &stream, int timeunit);
162
163
168     int size();
169
170
177     string print(bool htmlterminatelines = false, bool printgroupid = false);
178
179
184     void sortbyMass(int limit = -1);
185
186
190     void sortbyRelativeIntensityDesc();
191
192
196     void sortbyAbsoluteIntensityDesc();
197
198
202     void sortbyGroupId();
203
204
208     void sortbyOrderId();
209
210
215     int normalizeIntenzity();
216
217
222     void normalizeIntenzityByValue(double value);
223
224
229     void cropRelativeIntenzity(double minimumrelativeintensitythreshold);
230
231
236     void cropAbsoluteIntenzity(unsigned minimumabsoluteintensitythreshold);
237
238
244     void cropMinimumMZRatio(double minimummz, double errortolerance);
245
246
252     void cropMaximumMZRatio(double maximummz, double errortolerance);
253
254
260     cPeak& operator[](int position);
261
262
267     void resize(int size);
268
269
274     double getMinimumMZRatio();
275
276
281     double getMaximumMZRatio();
282
283
290     void removeIsotopes(int maximumcharge, double fragmentmasserrortolerance, cMainThread* os = 0);
291
292
299     int find(double mzratio, double fragmentmasserrortolerance);
300
301
306     void remove(int position);
307
308
314     void removeChargeVariants(int maximumcharge, double fragmentmasserrortolerance);
```

```
315
316
323     void removeNeutralLoss(double loss, int maximumcharge, double fragmentmasserrortolerance);
324
325
329     void removeObsolete();
330
331
337     void maxHighestPeaksInWindow(int maximumnumberofpeaksinwindow, double windowsize);
338
339
350     double getMaximumRelativeIntensityFromMZInterval(double minmz, double maxmz, bool hidematched, bool
      hideunmatched, ePeptideType peptidetype, bool hidescrambled);
351
352
363     double getMaximumAbsoluteIntensityFromMZInterval(double minmz, double maxmz, bool hidematched, bool
      hideunmatched, ePeptideType peptidetype, bool hidescrambled);
364
365
370     double getMaximumRelativeIntensity();
371
372
377     double getMaximumAbsoluteIntensity();
378
379
384     void setRetentionTime(double rt);
385
386
391     double getRetentionTime();
392
393
398     void setRetentionTimeUnit(int unit);
399
400
405     int getRetentionTimeUnit();
406
407
413     void setCoordinates(int x, int y);
414
415
420     int getCoordinateX();
421
422
427     int getCoordinateY();
428
429
434     void store(ofstream& os);
435
436
444     void load(ifstream& is, int fileversionpart1, int fileversionpart2, int fileversionpart3);
445
446
451     string& getTitle();
452
453
458     void setTitle(string& title);
459
460
465     void reducePeakDescriptions(unordered_map<string, int>& peakdesctoid);
466
467
472     void reduceIsotopeFormulaDescriptions(unordered_map<string, int>& isotopeformuladesctoid);
473
474
478     void fillOrderIDs();
479
480
485     void setIsotopeFlags(bool value);
486
487
491     void markIsotopes();
492
493
499     bool equals(cPeaksList& secondpeaklist);
500
501 };
502
503
507 Q_DECLARE_METATYPE(cPeaksList);
508
509 #endif
```

17.31 core/cSequence.h File Reference

The representation of a sequence.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include "core/utilities.h"
#include "core/cFragmentIons.h"
#include "core/cSummaryFormula.h"
```

Classes

- class cSequence

  The class representing a sequence.

17.31.1 Detailed Description

The representation of a sequence.

17.32 cSequence.h

Go to the documentation of this file.

```
1
7 #ifndef _CSEQUENCE_H
8 #define _CSEQUENCE_H
9
10 #include <iostream>
11 #include <fstream>
12 #include <vector>
13 #include <string>
14 #include "core/utilities.h"
15 #include "core/cFragmentIons.h"
16 #include "core/cSummaryFormula.h"
17
18
19 using namespace std;
20 using namespace boost;
21
22
26 class cSequence {
27
28     ePeptideType peptidetype;
29     string sequence;
30     string nterminalmodification;
31     string cterminalmodification;
32     string branchmodification;
33     string name;
34     cSummaryFormula summary;
35     string reference;
36     double rtmin;
37     double rtmax;
38     bool decoy;
39
40 public:
41
42
46     cSequence();
47
48
52     void clear();
53
54
59     void store(ofstream& os);
60
61
69     void load(ifstream& is, int fileversionpart1, int fileversionpart2, int fileversionpart3);
70
71
76     ePeptideType getPeptideType();
77
78
83     string& getSequence();
```

```
84
85
90     string& getName();
91
92
97     double getMinimumRetentionTime();
98
99
104     double getMaximumRetentionTime();
105
106
111     string& getReference();
112
113
118     void setPeptideType(ePeptideType peptidetype);
119
120
125     void setSequence(const string& sequence);
126
127
132     void setName(const string& name);
133
134
140     void setRetentionTime(double rtmin, double rtmax);
141
142
147     void setReference(const string& reference);
148
149
154     void setNTterminalModification(const string& modification);
155
156
161     void setCTterminalModification(const string& modification);
162
163
168     void setBranchModification(const string& modification);
169
170
175     string& getNTterminalModification();
176
177
182     string& getCTterminalModification();
183
184
189     string& getBranchModification();
190
191
196     string getNameWithReferenceAsHTMLString();
197
198
203     void setSummaryFormula(const string& formula);
204
205
210     string getSummaryFormula();
211
212
217     bool isDecoy();
218
219
224     void setDecoy(bool decoy);
225
226 };
227
228
229 #endif
```

17.33 core/cSequenceDatabase.h File Reference

The database of sequences.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include "core/utilities.h"
#include "core/cSequence.h"
```

Classes

- class cSequenceDatabase

  The class representing a database of sequences.

17.33.1 Detailed Description

The database of sequences.

17.34 cSequenceDatabase.h

Go to the documentation of this file.

```
1
7 #ifndef _CSEQUENCEDATABASE_H
8 #define _CSEQUENCEDATABASE_H
9
10 #include <iostream>
11 #include <fstream>
12 #include <vector>
13 #include <string>
14 #include "core/utilities.h"
15 #include "core/cSequence.h"
16
17
18 using namespace std;
19
20
21 class cMainThread;
22
23
27 class cSequenceDatabase {
28
29     vector<cSequence> sequences;
30
31     void parseRT(bool keeprt);
32
33
34 public:
35
36
40     cSequenceDatabase();
41
42
48     void loadFromPlainTextStream(ifstream &stream, bool keeprt);
49
50
56     void storeToPlainTextStream(ofstream &stream, bool keeprt);
57
58
63     int size();
64
65
71     cSequence& operator[](int position);
72
73
77     void clear();
78
79
84     void push_back(cSequence& sequence);
85
86
91     void erase(int index);
92
93
98     void store(ofstream& os);
99
100
108      void load(ifstream& is, int fileversionpart1, int fileversionpart2, int fileversionpart3);
109
110
115      void attachDecoys(bool& terminatecomputation);
116
117 };
118
119
120 #endif
```

17.35 core/cSummaryFormula.h File Reference

The representation of a summary formula.

```
#include <string>
#include <fstream>
#include "core/cFragmentIons.h"
```

Classes

- class cSummaryFormula

  The representation of a summary formula.

Functions

- void reduceMapToStringFormula (map< string, int > &atoms, string &formula)

  Reduce a map of atoms to a string.
- void addStringFormulaToMap (string &formula, map< string, int > &atoms)

  Explode and add a summary formula string to a map.
- void mergeMaps (map< string, int > &source, map< string, int > &target)

  Merge two maps of atoms into a target map.
- void rechargeMap (int charge, map< string, int > &atoms)

  Change the charge from 1+ to another charge in a map.

17.35.1 Detailed Description

The representation of a summary formula.

17.35.2 Function Documentation

17.35.2.1 addStringFormulaToMap()

```
void addStringFormulaToMap (
            string & formula,
            map< string, int > & atoms )
```

Explode and add a summary formula string to a map.

Parameters

formula	an input string formula
atoms	an input/output map of atoms

17.35.2.2 mergeMaps()

```
void mergeMaps (
            map< string, int > & source,
            map< string, int > & target )
```

Merge two maps of atoms into a target map.

Parameters

source	a source map
target	a target map

17.35.2.3 rechargeMap()

```
void rechargeMap (
            int charge,
            map< string, int > & atoms )
```

Change the charge from 1+ to another charge in a map.

Parameters

charge	new charge
atoms	an input/output map of atoms

17.35.2.4 reduceMapToStringFormula()

```
void reduceMapToStringFormula (
            map< string, int > & atoms,
            string & formula )
```

Reduce a map of atoms to a string.

Parameters

atoms	an input map of atoms
formula	an output string formula

17.36 cSummaryFormula.h

Go to the documentation of this file.

```
1
7  #ifndef _CSUMMARYFORMULA_H
8  #define _CSUMMARYFORMULA_H
9
10 #include <string>
11 #include <fstream>
12 #include "core/cFragmentIons.h"
13
14
15 using namespace std;
16 using namespace boost;
17
18
19 class cPeaksList;
20 class cPeakListSeries;
21 class cBricksDatabase;
22
23
29 void reduceMapToStringFormula(map<string, int>& atoms, string& formula);
30
31
37 void addStringFormulaToMap(string& formula, map<string, int>& atoms);
38
39
45 void mergeMaps(map<string, int>& source, map<string, int>& target);
46
47
53 void rechargeMap(int charge, map<string, int>& atoms);
54
55
59 class cSummaryFormula {
60
61     bool partial;
62
63     string formula;
64
65     void combineAtoms(cPeakListSeries& peaklistseries, int peaklistseriessize, vector<int>&
      peaklistseriessizes, cPeaksList& isotopeprofile, vector<int>& depthvector, int depth, double mass,
```

```
      double intensity, double charge, double minimumabsoluteintensity, double& localmaximumintensity,
      double minimumrelativeintensity);
66
67     double getIntensity(cBricksDatabase& bricksprobabilities, vector<int>& combarray, int mostintenseid,
      int mostintenseatomscount, int isotopecount);
68
69     void getIsotopeSummary(string& description, cBricksDatabase& bricks, vector<int>& combarray, int
      mostintenseid, int mostintenseatomscount, int isotopecount, bool positive);
70
71     void reduceIsotopeProfile(cPeaksList& isotopeprofile, cPeaksList& reducedprofile, double fwhm);
72
73 public:
74
75
79     cSummaryFormula();
80
81
87     cSummaryFormula(string& formula, bool partial = false);
88
89
93     void clear();
94
95
101     void setFormula(const string& formula, bool partial = false);
102
103
108     string getSummary();
109
110
116     string getFancySummary(int charge);
117
118
123     bool isPartial();
124
125
130     void setPartial(bool partial);
131
132
138     void addFormula(string& formula, bool remove = false);
139
140
146     void setFromMap(map<string, int>& atoms, bool partial);
147
148
154     bool isValid(string& errormessage);
155
156
161     bool isEmpty();
162
163
168     bool hasAllElementsPositive();
169
170
175     double getMass();
176
177
187     cPeaksList getIsotopePattern(double fwhm, int charge, bool positive, int maxchangedatoms, bool
      writedescription);
188
189
194     void store(ofstream& os);
195
196
201     void load(ifstream& is);
202
203
209     bool equals(cSummaryFormula& secondformula);
210
211 };
212
213
214 #endif
```

17.37 core/cTheoreticalSpectrum.h File Reference

The representation of a theoretical mass spectrum.

```
#include <vector>
#include <set>
#include <unordered_set>
#include <map>
#include <string>
```

```
#include <QMetaType>
#include "core/cParameters.h"
#include "core/cPeakListSeries.h"
#include "core/cCandidateSet.h"
```

Classes

- struct splitSite

  The structure representing a splitting site of a cyclic peptide.
- struct matchedSeries

  An auxiliary structure for visualisation of matched series of fragment ions.
- class cTheoreticalSpectrum

  The class representing a theoretical mass spectrum.

Functions

- int getNumberOfScoreHits (vector< double > &scores, double value)

  Get the number of hits with equal or better score.

17.37.1 Detailed Description

The representation of a theoretical mass spectrum.

17.37.2 Function Documentation

17.37.2.1 getNumberOfScoreHits()

```
int getNumberOfScoreHits (
            vector< double > & scores,
            double value )
```

Get the number of hits with equal or better score.

Parameters

scores	vector of scores
value	current score

Return values

int	number of hits

17.38 cTheoreticalSpectrum.h

Go to the documentation of this file.

```
1
7 #ifndef _CTHEORETICALSPECTRUM_H
8 #define _CTHEORETICALSPECTRUM_H
9
10 #include <vector>
11 #include <set>
12 #include <unordered_set>
13 #include <map>
14 #include <string>
15 #include <QMetaType>
16
```

```
17 #include "core/cParameters.h"
18 #include "core/cPeakListSeries.h"
19 #include "core/cCandidateSet.h"
20
21 class cMainThread;
22 class cDeNovoGraph;
23
24
25 using namespace std;
26
27
34 int getNumberOfScoreHits(vector<double>& scores, double value);
35
36
40 struct splitSite {
41
45     int first;
46
47
51     int second;
52
53
54     splitSite() {
55         first = 0;
56         second = 0;
57     }
58 };
59
60
64 struct matchedSeries {
65
69     vector<int> series;
70
71
75     string name;
76
77
81     matchedSeries() {
82         series.clear();
83         name = "";
84     }
85
86
91     void store(ofstream& os);
92
93
98     void load(ifstream& is);
99
100 };
101
102
106 class cTheoreticalSpectrum {
107
108     cParameters* parameters;
109     cPeaksList theoreticalpeaks;
110     cPeaksList experimentalpeaks;
111     cCandidate candidate;
112     int experimentalpeaksmatched;
113     vector<set<int> > experimentalmatches;
114     int scrambledpeaksmatched;
115     map<eFragmentIonType, map<int, int> > matchedions;
116     int peakstested;
117     double experimentalpeaksmatchedratio;
118     double weightedpeaksratio;
119     double cosinesimilarity;
120     int unmatchedexperimentalpeakscount;
121     string coveragebyseries;
122     bool valid;
123     double sumofrelativeintensities;
124     vector<string> rotationslabels;
125     int validposition;
126     int reversevalidposition;
127     int seriescompleted;
128     int pathid;
129     map<int, double> targetscores;
130     map<int, double> decoyscores;
131     map<int, double> fdrscores;
132
133     // remove hits of peaks without hits of parent peaks
134     void clearFalseHits(map<eFragmentIonType, map<int, vector<int> > >& series,
      vector<eFragmentIonType>& ions, vector<int>& losses);
135
136     // search for matches of experimental and theoretical peaks
137     void searchForPeakPairs(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize, cPeaksList&
      experimentalpeaks, double fragmentmasserrortolerance);
138
139     // compute additional scores
```

```
140     void computeStatistics(bool writedescription);
141
142     // generate charged forms of fragments
143     void generateChargedFragments(cPeak& peak, map<string, int>& atoms, int& peaklistrealsize, int
    maxcharge, bool writedescription, bool disablesummary);
144
145     // generate precursor ion and its variants
146     void generatePrecursorIon(vector<int>& intcomposition, cBricksDatabase&
    bricksdatabasewithcombinations, int& theoreticalpeaksrealsize, bool writedescription);
147
148     // generate precursor ion and its variants for other type
149     void generatePrecursorIonForOther(int& theoreticalpeaksrealsize, bool writedescription);
150
151     // generate scrambled sequences
152     void generateScrambledIons(cBricksDatabase& bricksdatabase, bool writedescription, int&
    theoreticalpeaksrealsize);
153
154     // update the list of losses
155     void updateListOfNeutralLosses(cBricksDatabase& bricksdatabase, cBrick& block, vector<int>&
    currentlosses, vector<double>& fragmentlossmass, vector<string>& fragmentlosssummary, vector<
    map<string, int> >& fragmentlossmap, bool writedescription, bool disablesummary);
156
157     // generate internal fragments
158     void generateInternalFragments(cBricksDatabase& bricksdatabase, cPeak& peak, int maxcharge, int&
    peaklistrealsize, vector<int>& intcomposition, int pos, map<string, int>& atoms, vector<int>&
    currentlosses, eFragmentIonType fragmentiontype, ePeptideType peptidetype, TRotationInfo* trotation,
    bool writedescription, bool disablesummary);
159
160     // select and normalize scrambled sequences
161     void selectAndNormalizeScrambledSequences(unordered_set<string>& scrambledsequences);
162
163     // select a proper fragment ion type for an experimental peak when masses of more theoretical
    fragment ions collide
164     void selectHigherPriorityIonTypeCID(cPeak& experimentalpeak, cPeak& theoreticalpeak);
165
166     // add a peak to the list
167     void addPeakToList(cPeak& peak, int& theoreticalpeaksrealsize);
168
169     // add an adduct to a peak description
170     void addAdductToDescription(string& description, vector<string>& metals);
171
172     // add theoretical peaks containing metals
173     void addMetalPeaks(cPeak& peak, vector<string>& metals, int& peaklistrealsize, int peakcharge, bool
    writedescription);
174
175     // remove unmatched biometal isotopes
176     void removeUnmatchedMetalIsotopes(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize,
    cPeaksList& experimentalpeaks);
177
178     // remove unmatched isotope patterns
179     void removeUnmatchedIsotopePatterns(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize,
    cPeaksList& experimentalpeaks, cPeaksList& outputtheoreticalpeaks, bool storeunmatchedpeaks);
180
181     // remove unmatched features
182     int removeUnmatchedFeatures(bool lcms, int peaklistseriesvectorid, cPeaksList& theoreticalpeaks, int
    theoreticalpeaksrealsize, cPeaksList& experimentalpeaks, vector< vector<int> >& hintsindex/*, int
    id*/);
183
184     // remove unmatched compounds
185     int removeUnmatchedCompounds(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize, cPeaksList&
    experimentalpeaks, int minimumiontypes);
186
187     // remove compounds outside RT range
188     void removeCompoundsOutsideRTRange(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize,
    cPeaksList& experimentalpeaks);
189
190     // remove peaks under annotation threshold
191     void removePeaksUnderAnnotationThreshold(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize,
    cPeaksList& experimentalpeaks);
192
193     // remove isotopic patterns under annotation threshold
194     void removePatternsUnderAnnotationThreshold(cPeaksList& theoreticalpeaks, int
    theoreticalpeaksrealsize, cPeaksList& experimentalpeaks);
195
196     // remove unmatched patterns in fine isotopic patterns
197     void removeUnmatchedPatternsFineSpectra(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize,
    cPeaksList& experimentalpeaks);
198
199     // remove unmatched pattern by intensity ratio
200     void removeUnmatchedPatternsByIntensityRatio(cPeaksList& theoreticalpeaks, int
    theoreticalpeaksrealsize, cPeaksList& experimentalpeaks);
201
202     // remove unmatched pattern by m/z difference
203     void removeUnmatchedPatternsByMZDifference(cPeaksList& theoreticalpeaks, int
    theoreticalpeaksrealsize, cPeaksList& experimentalpeaks);
204
205     // calculate envelope scores
```

```
206     void calculateEnvelopeScores(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize, cPeaksList&
    experimentalpeaks);
207
208     // remove decoy peak matches
209     void removeDecoyPeakMatches(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize, cPeaksList&
    experimentalpeaks);
210
211     // normalize theoretical intensities
212     void normalizeTheoreticalIntensities(cPeaksList& theoreticalpeaks, int theoreticalpeaksrealsize);
213
214     // calibrate peaklist to the most intense theoretical peak
215     double recalibrateToMostIntenseTheoreticalPeak(cPeaksList& peaklist);
216
217     // calculate standard PPM deviation
218     double getStandardPPMDeviation(cPeaksList& theoreticalpeaks, int start, int stop);
219
220     // calculate standard intensity deviation
221     double getStandardIntensityDeviation(cPeaksList& theoreticalpeaks, int start, int stop, double
    maximumexperimentalintensity);
222
223     // calculate standard m/z difference deviation
224     double getStandardMZDifferenceDeviation(cPeaksList& theoreticalpeaks, int start, int stop);
225
226     // calculate the angle distance
227     double getAngleDistance(cPeaksList& theoreticalpeaks, int start, int stop);
228
229     // generate isotope patterns of fragment ions
230     void generateFragmentIsotopePatterns(int& theoreticalpeaksrealsize, bool writedescription,
    unordered_map<string, int>* isotopeformuladesctoid);
231
232     // fill annotations of experimental peaks and remove unmatched theoretical peaks
233     void fillExperimentalAnnotationsAndRemoveUnmatchedTheoreticalPeaks(int& theoreticalpeaksrealsize,
    ePeptideType peptidetype, cPeaksList& unmatchedpeaksinmatchedpatterns, bool
    reportunmatchedtheoreticalpeaks, bool writedescription);
234
235     // check the existence of an isotope
236     bool checkIsotopeHighRes(string& elementstring, string& isotopestring, double isotopemass1, double
    isotopemass2, double minintensityratio, double maxintensityratio, cPeaksList& theoreticalpeaks, int
    theoreticalpeaksrealsize, cPeaksList& experimentalpeaks, int start, int stop, int maximumintensityid,
    double maximumexperimentalintensity);
237
238     // check the existence of an isotope
239     bool checkIsotopeLowRes(string& elementstring, string& isotopestring, double isotopemass1, double
    isotopemass2, double minintensityratio, double maxintensityratio, cPeaksList& theoreticalpeaks, int
    theoreticalpeaksrealsize, cPeaksList& experimentalpeaks, int start, int stop, int maximumintensityid,
    double maximumexperimentalintensity);
240
241     // get intensity multiplier
242     double getIntensityMultiplier(cPeaksList& theoreticalpeaks, int peakid, string& elementstring,
    size_t startpos);
243
244
245 public:
246
247
251     cTheoreticalSpectrum();
252
253
259     cTheoreticalSpectrum(cParameters* parameters, cCandidate& candidate);
260
261
266     void clear(bool clearpeaklist = true);
267
268
273     cCandidate& getCandidate();
274
275
280     void setCandidate(cCandidate& candidate);
281
282
287     int getNumberOfPeaks();
288
289
301     int compareBranched(cPeaksList& sortedpeaklist, cBricksDatabase& bricksdatabasewithcombinations,
    bool writedescription, regex& sequencetag, regex& searchedsequence, cPeaksList&
    unmatchedpeaksinmatchedpatterns, unordered_map<string, int>* isotopeformuladesctoid);
302
303
315     int compareLinear(cPeaksList& sortedpeaklist, cBricksDatabase& bricksdatabasewithcombinations, bool
    writedescription, regex& sequencetag, regex& searchedsequence, cPeaksList&
    unmatchedpeaksinmatchedpatterns, unordered_map<string, int>* isotopeformuladesctoid);
316
317
329     int compareCyclic(cPeaksList& sortedpeaklist, cBricksDatabase& bricksdatabasewithcombinations, bool
    writedescription, regex& sequencetag, regex& searchedsequence, cPeaksList&
    unmatchedpeaksinmatchedpatterns, unordered_map<string, int>* isotopeformuladesctoid);
330
```

```
331
343     int compareBranchCyclic(cPeaksList& sortedpeaklist, cBricksDatabase& bricksdatabasewithcombinations,
        bool writedescription, regex& sequencetag, regex& searchedsequence, cPeaksList&
        unmatchedpeaksinmatchedpatterns, unordered_map<string, int>* isotopeformuladesctoid);
344
345
357     int compareLinearPolyketide(cPeaksList& sortedpeaklist, cBricksDatabase&
        bricksdatabasewithcombinations, bool writedescription, regex& sequencetag, regex& searchedsequence,
        cPeaksList& unmatchedpeaksinmatchedpatterns, unordered_map<string, int>* isotopeformuladesctoid);
358
359
371     int compareCyclicPolyketide(cPeaksList& sortedpeaklist, cBricksDatabase&
        bricksdatabasewithcombinations, bool writedescription, regex& sequencetag, regex& searchedsequence,
        cPeaksList& unmatchedpeaksinmatchedpatterns, unordered_map<string, int>* isotopeformuladesctoid);
372
373
382     int compareOther(cPeaksList& sortedpeaklist, bool writedescription, cPeaksList&
        unmatchedpeaksinmatchedpatterns, unordered_map<string, int>* isotopeformuladesctoid);
383
384
392     void generateMSSpectrum(int sequencestart, int sequencestop, bool& terminatecomputation, bool
        writedescription);
393
394
401     void generateFineMSSpectrum(int sequencestart, int sequencestop, bool& terminatecomputation);
402
403
414     void getHintsIndex(int id, int peaklistseriesvectorid, cTheoreticalSpectrum& tsfull, cPeaksList&
        unmatchedpeaksinmatchedpatterns, vector< vector<int> >& hintsindex, bool lcmsrt, bool skipcomparison);
415
416
427     void compareMSSpectrum(int id, int peaklistseriesvectorid, cTheoreticalSpectrum& tsfull, cPeaksList&
        unmatchedpeaksinmatchedpatterns, vector< vector<int> >& hintsindex, bool lcmsrt, bool skipcomparison);
428
429
436     void compareAverageMSSpectrum(cPeaksList& averagespectrum, cTheoreticalSpectrum& tsfull, cPeaksList&
        unmatchedpeaksinmatchedpatterns);
437
438
439
445     void finalizeMSSpectrum(cPeaksList& unmatchedpeaksinmatchedpatterns, bool writedescription);
446
447
452     int getNumberOfMatchedPeaks() const;
453
454
459     int getNumberOfScrambledPeaks();
460
461
468     int getNumberOfMatchedPeaks(eFragmentIonType iontype, int neutrallosstype) const;
469
470
475     int getNumberOfMatchedPeaksB() const;
476
477
482     int getNumberOfMatchedPeaksY() const;
483
484
489     int getNumberOfMatchedPeaksYB() const;
490
491
496     double getRatioOfMatchedPeaks();
497
498
503     double getWeightedRatioOfMatchedPeaks() const;
504
505
510     double getCosineSimilarity() const;
511
512
531     void generateNTerminalFragmentIons(int maxcharge, int& peaklistrealsize, vector<int>&
        intcomposition, eFragmentIonType fragmentiontype, int neutrallosstype, cBricksDatabase&
        bricksdatabase, bool writedescription, int rotationid, vector<splitSite>& splittingsites,
        vector<cFragmentIonType>& searchedmodifications, ePeptideType peptidetype, bool regularblocksorder,
        TRotationInfo* trotation = 0, eResidueLossType leftresiduelosstype = h2o_loss, bool
        hasfirstblockartificial = false);
532
533
552     void generateCTerminalFragmentIons(int maxcharge, int& peaklistrealsize, vector<int>&
        intcomposition, eFragmentIonType fragmentiontype, int neutrallosstype, cBricksDatabase&
        bricksdatabase, bool writedescription, int rotationid, vector<splitSite>& splittingsites,
        vector<cFragmentIonType>& searchedmodifications, ePeptideType peptidetype, bool regularblocksorder,
        TRotationInfo* trotation = 0, eResidueLossType rightresiduelosstype = h2o_loss, bool
        haslastblockartificial = false);
553
554
559     void sortByMass(int limit = -1);
```

```
560
561
566     void resizePeakList(int size);
567
568
573     double getSumOfRelativeIntensities() const;
574
575
581     cPeak& operator[](int position);
582
583
588     cPeaksList& getExperimentalSpectrum();
589
590
595     void setExperimentalSpectrum(cPeaksList& experimentalspectrum);
596
597
602     set<int>& getExperimentalMatches(int peakid);
603
604
609     string getCoverageBySeries();
610
611
616     int getUnmatchedExperimentalPeaksCount();
617
618
623     cPeaksList* getTheoreticalPeaks();
624
625
630     bool isValid();
631
632
637     int getValidPosition();
638
639
644     int getReverseValidPosition();
645
646
651     void setValidSequence(regex& searchedsequence);
652
653
658     vector<string>& getLabelsOfRotations();
659
660
665     int getNumberOfCompletedSeries();
666
667
672     void setNumberOfCompletedSeries(int numberofcompletedseries);
673
674
679     void setPathId(int pathid);
680
681
686     int getPathId() const;
687
688
693     void setParameters(cParameters* parameters);
694
695
701     double getTargetPatternScore(int groupid);
702
703
709     double getTargetPatternFDR(int groupid);
710
711
716     map<int, double>& getTargetScores();
717
718
723     map<int, double>& getDecoyScores();
724
725
732     void setFDRs(vector<double>& targetscoresvector, vector<double>& fdrs, cPeaksList&
    unmatchedpeaksinmatchedpatterns);
733
734
739     void store(ofstream& os);
740
741
749     void load(ifstream& is, int fileversionpart1, int fileversionpart2, int fileversionpart3);
750
751
757     bool equals(cTheoreticalSpectrum& secondtheoreticalspectrum);
758
759
764     void attach(cTheoreticalSpectrum& secondtheoreticalspectrum);
765
766 };
```

```
767
768
769 #endif
770
```

17.39 core/cTheoreticalSpectrumList.h File Reference

The representation of lists of theoretical spectra.

```
#include <algorithm>
#include <list>
#include <QMutex>
#include "parallel/cSpectrumAnalyzerThreadMS1.h"
#include "parallel/cSpectrumComparatorThreadMS1.h"
#include "parallel/cSpectrumComparatorThreadMS2.h"
#include "parallel/cSpectrumGeneratorThreadMS1.h"
```

Classes

- class cTheoreticalSpectrumList

  The class representing lists of theoretical spectra.

17.39.1 Detailed Description

The representation of lists of theoretical spectra.

17.40 cTheoreticalSpectrumList.h

Go to the documentation of this file.

```
1
7 #ifndef _CTHEORETICALSPECTRUMLIST_H
8 #define _CTHEORETICALSPECTRUMLIST_H
9
10 #include <algorithm>
11 #include <list>
12 #include <QMutex>
13
14 #include "parallel/cSpectrumAnalyzerThreadMS1.h"
15 #include "parallel/cSpectrumComparatorThreadMS1.h"
16 #include "parallel/cSpectrumComparatorThreadMS2.h"
17 #include "parallel/cSpectrumGeneratorThreadMS1.h"
18
19 class cMainThread;
20
21
25 class cTheoreticalSpectrumList {
26
27     vector<vector<cTheoreticalSpectrum> > theoreticalspectra;
28     list<cTheoreticalSpectrum> spectrumbuffer;
29     cMainThread* os;
30     cParameters* parameters;
31     cDeNovoGraph* graph;
32
33     double currentworstscore;
34
35     QMutex mutex;
36
37     double getCurrentWorstScore();
38
39     void computeNumbersOfCompletedSeries(int fileid);
40
41     void fixRegularExpression(string& s);
42
43 public:
44
45
49     cTheoreticalSpectrumList();
50
51
55     void clear();
56
57
```

```
62      void clear(int fileid);
63
64
71      void initialize(cMainThread& os, cParameters& parameters, cDeNovoGraph* graph);
72
73
79      void add(int fileid, cTheoreticalSpectrum& theoreticalspectrum);
80
81
88      cTheoreticalSpectrum& get(int fileid, int order);
89
90
95      int size();
96
97
103      int size(int fileid);
104
105
110      void resize(int size);
111
112
118      void resize(int fileid, int size);
119
120
128      int parallelCompareAndStore(int fileid, cCandidateSet& candidates, bool& terminatecomputation);
129
130
139      double updatekNNList(cTheoreticalSpectrum& theoreticalspectrum, int theoreticalpeaksrealsize, double
     score, regex* rxsearchedsequence);
140
141
146      void sortAndFitSize(int fileid);
147
148 };
149
150
151 #endif
152
```

17.41 core/utilities.h File Reference

Auxiliary funtions and structures.

```
#include <fstream>
#include <iostream>
#include <vector>
#include <deque>
#include <map>
#include <unordered_map>
#include <string>
#include <sstream>
#include <iomanip>
#include <cfloat>
#include <cmath>
#include <QString>
#include <QByteArray>
#include <QMetaType>
#include <QAbstractItemModel>
#include <QModelIndex>
#include <boost/regex.hpp>
```

Classes

- struct cCoordinateInfo

  Store aggregated information about a coordinate.

- struct compareStringBySize

  Compare items in a string vector by size.

Macros

- #define **WIN** 0
- #define **UNX** 1
- #define **OSX** 2
- #define **OS_TYPE** WIN

Enumerations

- enum eVendorType { **unknownvendor** = 0 , **bruker** = 1 , **waters** = 2 }

  Vendor types.
- enum eModeType {
  denovoengine = 0 , **singlecomparison** = 1 , **databasesearch** = 2 , **dereplication** = 3 ,
  compoundsearch = 4 }

  Running modes of the application.
- enum ePeptideType {
  linear = 0 , **cyclic** = 1 , **branched** = 2 , **branchcyclic** = 3 ,
  linearpolyketide = 4 , **cyclicpolyketide** = 5 , **other** = 6 }

  The types of peptides supported by the application.

Functions

- void storeIntVector (vector< int > &v, ofstream &os)

  Store a vector of integers into an output stream.
- void loadIntVector (vector< int > &v, ifstream &is)

  Load a vector of integers from an input stream.
- void storeDoubleVector (vector< double > &v, ofstream &os)

  Store a vector of doubles into an output stream.
- void loadDoubleVector (vector< double > &v, ifstream &is)

  Load a vector of doubles from an input stream.
- void storeString (string &s, ofstream &os)

  Store a string into an output stream.
- void loadString (string &s, ifstream &is)

  Load a string from an input stream.
- void storeStringVector (vector< string > &v, ofstream &os)

  Store a vector of strings into an output stream.
- void loadStringVector (vector< string > &v, ifstream &is)

  Load a vector of strings from an input stream.
- void storeStringIntMap (map< string, int > &map, ofstream &os)

  Store a map<string, int> into an output stream.
- void loadStringIntMap (map< string, int > &map, ifstream &is)

  Load a map<string, int> from an input stream.
- void storeStringIntMapVector (vector< map< string, int > > &vector, ofstream &os)

  Store a vector of map<string, int> into an output stream.
- void loadStringIntMapVector (vector< map< string, int > > &vector, ifstream &is)

  Load a vector of map<string, int> from an input stream.
- void storeIntStringMap (map< int, string > &map, ofstream &os)

  Store a map<int, string> into an output stream.
- void loadIntStringMap (map< int, string > &map, ifstream &is)

  Load a map<int, string> from an input stream.
- void convertStringIntUnorderedMapToStringVector (unordered_map< string, int > &map, vector< string > &vector)

  Convert an unordered_map<string, int> to a vector<string>.

- bool isWhiteSpace (char c)

  Check if c is a white space.
- bool isWhiteSpaceExceptSpace (char c)

  Check if c is a white space (except normal space).
- string & removeWhiteSpacesExceptSpaces (string &s)

  Remove white spaces except normal spaces from a string.
- bool checkRegex (ePeptideType peptidetype, string &sequence, string &errormessage)

  Check the syntax of a peptide sequence.
- bool checkFile (string filename)

  Check if a file exists.
- bool compareDoubles (double &first, double &second, double epsilon=0.0000001f)

  Check if two doubles are equal.
- bool compareStringVectors (vector< string > &first, vector< string > &second)

  Check if two vectors of strings are equal.
- bool compareMapsIntDouble (map< int, double > &first, map< int, double > &second, double epsilon=0.↩ 0000001f)

  Check if two maps of doubles are equal.
- void parseBranch (ePeptideType peptidetype, string &composition, vector< string > &vectorcomposition, int &branchstart, int &branchend)

  Parse a position of a branch from a peptide sequence.
- ePeptideType getPeptideTypeFromString (string &s)

  Convert a string to peptide type.
- string getStringFromPeptideType (ePeptideType peptidetype)

  Convert the peptide type to a string.
- double cropPrecisionToSixDecimals (double value)

  Crop a precision of a double to six decimal places.
- QByteArray cropDecimalsByteArray (double value)

  Crop decimals of a double.
- QByteArray cropPrecisionToSixDecimalsByteArray (double value)

  Crop a precision of a double to six decimal places.
- string getShortFileName (string filename)

  Get a short filename from a string.
- **Q_DECLARE_METATYPE** (vector< cCoordinateInfo >)

  Register vector<cCoordinateInfo> by Qt.
- double fact (int value)

  Get factorial of a value.
- double median (vector< double > &vector)

  Get median from a vector of doubles.
- string stripHTML (string &htmlstring)

  Strip HTML tags from a HTML string.
- void stripIsomers (string &peptidesequence)

  Strip isomers of building blocks from an input peptide sequence.
- void stripIsomersFromStringVector (vector< string > &acronyms)

  Strip isomers of building blocks from an input string vector.
- bool proxyModelCheckInt (QAbstractItemModel *model, int index, int row, int column, QString str, const QModelIndex &parent)

  Proxy model - int comparator.
- bool proxyModelCheckDouble (QAbstractItemModel *model, int index, int row, int column, QString str, const QModelIndex &parent)

  Proxy model - double comparator.

- bool proxyModelCheckString (QAbstractItemModel ∗model, int index, int row, int column, QString &itemstr, QString str, const QModelIndex &parent, bool wholeword, Qt::CaseSensitivity casesensitive)

  Proxy model - string comparator.
- bool isCompatibleVersion (int fileversionpart1, int fileversionpart2, int fileversionpart3, int lastcompatiblepart1, int lastcompatiblepart2, int lastcompatiblepart3)

  Check the compatibility of CycloBranch version.
- string fixLabelCharge (string &str, int sourcecharge, int targetcharge)

  Fix the name of a multiply charged ion in MS mode.

Variables

- const double **pi** = 3.141592653589793

  The pi value.
- const double **econst** = 2.718281828459045

  The e constant.
- const int **defaultwinsizex** = 1350

  Default window size x.
- const int **defaultwinsizey** = 875

  Default window size y.
- QString **appname**

  The name of the application.
- QString **appversion**

  The version of the application.
- QString **lastcompatibleappversion**

  Last compatible version of the application.
- QString **installdir**

  A directory where the application is installed under Linux or OSX.

17.41.1 Detailed Description

Auxiliary funtions and structures.

17.41.2 Function Documentation

17.41.2.1 checkFile()

```
bool checkFile (
            string filename )
```

Check if a file exists.

Parameters

filename	filename

Return values

bool	true if the file exists, false otherwise

17.41.2.2 checkRegex()

```
bool checkRegex (
            ePeptideType peptidetype,
            string & sequence,
            string & errormessage )
```

Check the syntax of a peptide sequence.

Parameters

peptidetype	peptide type
sequence	peptide sequence
errormessage	an error message if false is returned

Return values

bool	true when the syntax is correct, false otherwise

17.41.2.3 compareDoubles()

```
bool compareDoubles (
            double & first,
            double & second,
            double epsilon = 0.0000001f )
```

Check if two doubles are equal.

Parameters

first	first value
second	second value
epsilon	epsilon value for comparison

Return values

bool	true if the values are equal

17.41.2.4 compareMapsIntDouble()

```
bool compareMapsIntDouble (
            map< int, double > & first,
            map< int, double > & second,
            double epsilon = 0.0000001f )
```

Check if two maps of doubles are equal.

Parameters

first	first map
second	second map
epsilon	epsilon value for double comparison

Return values

bool	true if the maps are equal

17.41.2.5 compareStringVectors()

```
bool compareStringVectors (
            vector< string > & first,
            vector< string > & second )
```

Check if two vectors of strings are equal.

Parameters

first	first vector
second	second vector

Return values

bool	true if the vectors are equal

17.41.2.6 convertStringIntUnorderedMapToStringVector()

```
void convertStringIntUnorderedMapToStringVector (
            unordered_map< string, int > & map,
            vector< string > & vector )
```

Convert an unordered_map<string, int> to a vector<string>.

Parameters

map	an input map
vector	an output vector

17.41.2.7 cropDecimalsByteArray()

```
QByteArray cropDecimalsByteArray (
            double value )
```

Crop decimals of a double.

Parameters

value	an input value

Return values

QByteArray	a value with the limited precision

17.41.2.8 cropPrecisionToSixDecimals()

```
double cropPrecisionToSixDecimals (
            double value )
```

Crop a precision of a double to six decimal places.

Parameters

value	an input value

Return values

double	a value with the limited precision

17.41.2.9 cropPrecisionToSixDecimalsByteArray()

```
QByteArray cropPrecisionToSixDecimalsByteArray (
            double value )
```

Crop a precision of a double to six decimal places.

Parameters

value	an input value

Return values

QByteArray	a value with the limited precision

17.41.2.10 fact()

```
double fact (
            int value )
```

Get factorial of a value.

Parameters

value	value

Return values

double	factorial of the value

17.41.2.11 fixLabelCharge()

```
string fixLabelCharge (
            string & str,
            int sourcecharge,
            int targetcharge )
```

Fix the name of a multiply charged ion in MS mode.

Parameters

str	original ion name
sourcecharge	source charge
targetcharge	target charge

Return values

string	fixed name

17.41.2.12 getPeptideTypeFromString()

```
ePeptideType getPeptideTypeFromString (
            string & s )
```

Convert a string to peptide type.

Parameters

s	string

Return values

ePeptideType	type of peptide

17.41.2.13 getShortFileName()

```
string getShortFileName (
            string filename )
```

Get a short filename from a string.

Parameters

filename	long filename

Return values

string	short filename

17.41.2.14 getStringFromPeptideType()

```
string getStringFromPeptideType (
            ePeptideType peptidetype )
```

Convert the peptide type to a string.

Parameters

peptidetype	type of peptide

Return values

string	string

17.41.2.15 isCompatibleVersion()

```
bool isCompatibleVersion (
            int fileversionpart1,
            int fileversionpart2,
            int fileversionpart3,
            int lastcompatiblepart1,
            int lastcompatiblepart2,
            int lastcompatiblepart3 )
```

Check the compatibility of CycloBranch version.

Parameters

fileversionpart1	tested version (part 1)
fileversionpart2	tested version (part 2)
fileversionpart3	tested version (part 3)
lastcompatiblepart1	last compatible version (part 1)
lastcompatiblepart2	last compatible version (part 2)
lastcompatiblepart3	last compatible version (part 3)

Return values

bool	true if the versions are compatible

17.41.2.16 isWhiteSpace()

```
bool isWhiteSpace (
            char c )
```

Check if *c* is a white space.

Parameters

c	char

Return values

bool	true if *c* is a white space; false otherwise.

17.41.2.17 isWhiteSpaceExceptSpace()

```
bool isWhiteSpaceExceptSpace (
            char c )
```

Check if *c* is a white space (except normal space).

Parameters

c	char

Return values

bool	true if *c* is a white space (except normal space); false otherwise.

17.41.2.18 loadDoubleVector()

```
void loadDoubleVector (
            vector< double > & v,
            ifstream & is )
```

Load a vector of doubles from an input stream.

Parameters

v	reference to a vector of doubles
is	reference to an input stream

17.41.2.19 loadIntStringMap()

```
void loadIntStringMap (
            map< int, string > & map,
            ifstream & is )
```

Load a map<int, string> from an input stream.

Parameters

map	reference to a map
is	reference to an input stream

17.41.2.20 loadIntVector()

```
void loadIntVector (
            vector< int > & v,
            ifstream & is )
```

Load a vector of integers from an input stream.

Parameters

v	reference to a vector of integers
is	reference to an input stream

17.41.2.21 loadString()

```
void loadString (
            string & s,
```

```
            ifstream & is )
```

Load a string from an input stream.

Parameters

s	reference to a string
is	reference to an input stream

17.41.2.22 loadStringIntMap()

```
void loadStringIntMap (
            map< string, int > & map,
            ifstream & is )
```

Load a map<string, int> from an input stream.

Parameters

map	reference to a map
is	reference to an input stream

17.41.2.23 loadStringIntMapVector()

```
void loadStringIntMapVector (
            vector< map< string, int > > & vector,
            ifstream & is )
```

Load a vector of map<string, int> from an input stream.

Parameters

vector	reference to a vector
is	reference to an input stream

17.41.2.24 loadStringVector()

```
void loadStringVector (
            vector< string > & v,
            ifstream & is )
```

Load a vector of strings from an input stream.

Parameters

v	reference to a vector of strings
is	reference to an input stream

17.41.2.25 median()

```
double median (
            vector< double > & vector )
```

Get median from a vector of doubles.

Parameters

vector	vector of doubles

Return values

double	median value

17.41.2.26 parseBranch()

```
void parseBranch (
            ePeptideType peptidetype,
            string & composition,
            vector< string > & vectorcomposition,
            int & branchstart,
            int & branchend )
```

Parse a position of a branch from a peptide sequence.

Parameters

peptidetype	peptide type
composition	a peptide sequence where all blocks "[blockX]-[blockY]" were replaced by ids "1-2"; but "\(" and "\)" are still included
vectorcomposition	an output vector with ids of blocks represented as strings
branchstart	start position of a branch (an output value)
branchend	end position of a branch (an output value)

17.41.2.27 proxyModelCheckDouble()

```
bool proxyModelCheckDouble (
            QAbstractItemModel * model,
            int index,
            int row,
            int column,
            QString str,
            const QModelIndex & parent )
```

Proxy model - double comparator.

Parameters

model	model
index	index
row	row
column	column
str	query string
parent	parent

Return values

bool	true if the item is valid

17.41.2.28 proxyModelCheckInt()

```
bool proxyModelCheckInt (
            QAbstractItemModel * model,
            int index,
            int row,
            int column,
            QString str,
            const QModelIndex & parent )
```

Proxy model - int comparator.

Parameters

model	model
index	index
row	row
column	column
str	query string
parent	parent

Return values

bool	true if the item is valid

17.41.2.29 proxyModelCheckString()

```
bool proxyModelCheckString (
            QAbstractItemModel * model,
            int index,
            int row,
            int column,
            QString & itemstr,
            QString str,
            const QModelIndex & parent,
            bool wholeword,
            Qt::CaseSensitivity casesensitive )
```

Proxy model - string comparator.

Parameters

model	model
index	index
row	row
column	column
itemstr	string of current item
str	query string
parent	parent
wholeword	wholeword
casesensitive	casesensitive

Return values

bool	true if the item is valid

17.41.2.30 removeWhiteSpacesExceptSpaces()

```
string & removeWhiteSpacesExceptSpaces (
            string & s )
```

Remove white spaces except normal spaces from a string.

Parameters

s	reference to a string

Return values

string	reference to the input string

17.41.2.31 storeDoubleVector()

```
void storeDoubleVector (
            vector< double > & v,
            ofstream & os )
```

Store a vector of doubles into an output stream.

Parameters

v	reference to a vector of doubles
os	reference to an output stream

17.41.2.32 storeIntStringMap()

```
void storeIntStringMap (
            map< int, string > & map,
            ofstream & os )
```

Store a map<int, string> into an output stream.

Parameters

map	reference to a map
os	reference to an output stream

17.41.2.33 storeIntVector()

```
void storeIntVector (
            vector< int > & v,
            ofstream & os )
```

Store a vector of integers into an output stream.

Parameters

v	reference to a vector of integers
os	reference to an output stream

17.41.2.34 storeString()

```
void storeString (
            string & s,
            ofstream & os )
```

Store a string into an output stream.

Parameters

s	reference to a string
os	reference to an output stream

17.41.2.35 storeStringIntMap()

```
void storeStringIntMap (
            map< string, int > & map,
            ofstream & os )
```

Store a map<string, int> into an output stream.

Parameters

map	reference to a map
os	reference to an output stream

17.41.2.36 storeStringIntMapVector()

```
void storeStringIntMapVector (
            vector< map< string, int > > & vector,
            ofstream & os )
```

Store a vector of map<string, int> into an output stream.

Parameters

vector	reference to a vector
os	reference to an output stream

17.41.2.37 storeStringVector()

```
void storeStringVector (
            vector< string > & v,
            ofstream & os )
```

Store a vector of strings into an output stream.

Parameters

v	reference to a vector of strings
os	reference to an output stream

17.41.2.38 stripHTML()

```
string stripHTML (
            string & htmlstring )
```

Strip HTML tags from a HTML string.

Parameters

htmlstring	reference to a string with HTML tags

Return values

string	stripped string

17.41.2.39 stripIsomers()

```
void stripIsomers (
            string & peptidesequence )
```

Strip isomers of building blocks from an input peptide sequence.

Parameters

peptidesequence	a peptide sequence

17.41.2.40 stripIsomersFromStringVector()

```
void stripIsomersFromStringVector (
            vector< string > & acronyms )
```

Strip isomers of building blocks from an input string vector.

Parameters

acronyms	vector of building block acronyms with isomers

17.42 utilities.h

Go to the documentation of this file.

```
1
7 #ifndef _UTILITIES_H
8 #define _UTILITIES_H
9
10 #include <fstream>
11 #include <iostream>
12 #include <vector>
13 #include <deque>
14 #include <map>
```

```
15 #include <unordered_map>
16 #include <string>
17 #include <sstream>
18 #include <iomanip>
19 #include <cfloat>
20 #include <cmath>
21 #include <QString>
22 #include <QByteArray>
23 #include <QMetaType>
24 #include <QAbstractItemModel>
25 #include <QModelIndex>
26 #include <boost/regex.hpp>
27
28 class cBrick;
29
30
31 #define WIN 0   // Windows
32 #define UNX 1   // Linux/UNIX
33 #define OSX 2   // Mac OS X
34
35
36 #ifdef LINUX
37 #define OS_TYPE UNX
38 #define sscanf_s sscanf
39 #define sprintf_s sprintf
40 #else
41 #ifdef MACOSX
42 #define OS_TYPE OSX
43 #define sscanf_s sscanf
44 #define sprintf_s sprintf
45 #else
46 #define OS_TYPE WIN
47 #endif
48 #endif
49
50
51 using namespace std;
52 using namespace boost;
53
54
58 const double pi = 3.141592653589793;
59
60
64 const double econst = 2.718281828459045;
65
66
70 const int defaultwinsizex = 1350;
71
72
76 const int defaultwinsizey = 875;
77
78
82 enum eVendorType {
83     unknownvendor = 0,
84     bruker = 1,
85     waters = 2
86 };
87
88
92 enum eModeType {
93     denovoengine = 0,
94     singlecomparison = 1,
95     databasesearch = 2,
96     dereplication = 3,
97     compoundsearch = 4
98 };
99
100
104 enum ePeptideType {
105     linear = 0,
106     cyclic = 1,
107     branched = 2,
108     branchcyclic = 3,
109     linearpolyketide = 4,
110     cyclicpolyketide = 5,
111     other = 6
112 };
113
114
118 extern QString appname;
119
120
124 extern QString appversion;
125
126
130 extern QString lastcompatibleappversion;
131
```

```
132
136 extern QString installdir;
137
138
144 void storeIntVector(vector<int>& v, ofstream& os);
145
146
152 void loadIntVector(vector<int>& v, ifstream& is);
153
154
160 void storeDoubleVector(vector<double>& v, ofstream& os);
161
162
168 void loadDoubleVector(vector<double>& v, ifstream& is);
169
170
176 void storeString(string& s, ofstream& os);
177
178
184 void loadString(string& s, ifstream& is);
185
186
192 void storeStringVector(vector<string>& v, ofstream& os);
193
194
200 void loadStringVector(vector<string>& v, ifstream& is);
201
202
208 void storeStringIntMap(map<string, int>& map, ofstream& os);
209
210
216 void loadStringIntMap(map<string, int>& map, ifstream& is);
217
218
224 void storeStringIntMapVector(vector< map<string, int> >& vector, ofstream& os);
225
226
232 void loadStringIntMapVector(vector< map<string, int> >& vector, ifstream& is);
233
234
240 void storeIntStringMap(map<int, string>& map, ofstream& os);
241
242
248 void loadIntStringMap(map<int, string>& map, ifstream& is);
249
250
256 void convertStringIntUnorderedMapToStringVector(unordered_map<string, int>& map, vector<string>&
    vector);
257
258
264 bool isWhiteSpace(char c);
265
266
272 bool isWhiteSpaceExceptSpace(char c);
273
274
280 string& removeWhiteSpacesExceptSpaces(string& s);
281
282
290 bool checkRegex(ePeptideType peptidetype, string& sequence, string& errormessage);
291
292
298 bool checkFile(string filename);
299
300
308 bool compareDoubles(double& first, double& second, double epsilon = 0.0000001f);
309
310
317 bool compareStringVectors(vector<string>& first, vector<string>& second);
318
319
327 bool compareMapsIntDouble(map<int, double>& first, map<int, double>& second, double epsilon =
    0.0000001f);
328
329
338 void parseBranch(ePeptideType peptidetype, string& composition, vector<string>& vectorcomposition, int&
    branchstart, int& branchend);
339
340
346 ePeptideType getPeptideTypeFromString(string& s);
347
348
354 string getStringFromPeptideType(ePeptideType peptidetype);
355
356
362 double cropPrecisionToSixDecimals(double value);
363
```

```
364
370 QByteArray cropDecimalsByteArray(double value);
371
372
378 QByteArray cropPrecisionToSixDecimalsByteArray(double value);
379
380
386 string getShortFileName(string filename);
387
388
392 struct cCoordinateInfo {
393
397     int id;
398
399
403     int x;
404
405
409     int y;
410
411
415     double relativeintensity;
416
417
421     double absoluteintensity;
422
423
427     string description;
428
429
433     cCoordinateInfo() {
434         id = 0;
435         x = 0;
436         y = 0;
437         relativeintensity = 0;
438         absoluteintensity = 0;
439         description = "";
440     }
441
442 };
443
444
448 Q_DECLARE_METATYPE(vector<cCoordinateInfo>);
449
450
456 double fact(int value);
457
458
464 double median(vector<double>& vector);
465
466
472 string stripHTML(string& htmlstring);
473
474
479 void stripIsomers(string& peptidesequence);
480
481
486 void stripIsomersFromStringVector(vector<string>& acronyms);
487
488
492 struct compareStringBySize {
493
500     bool operator()(const std::string& first, const std::string& second) {
501         size_t size1 = first.size();
502         size_t size2 = second.size();
503         if (first.size() < second.size()) {
504             return true;
505         }
506         if (first.size() > second.size()) {
507             return false;
508         }
509         return first.compare(second) < 0;
510     }
511
512 };
513
514
525 bool proxyModelCheckInt(QAbstractItemModel* model, int index, int row, int column, QString str, const
      QModelIndex& parent);
526
527
538 bool proxyModelCheckDouble(QAbstractItemModel* model, int index, int row, int column, QString str, const
      QModelIndex& parent);
539
540
554 bool proxyModelCheckString(QAbstractItemModel* model, int index, int row, int column, QString& itemstr,
      QString str, const QModelIndex& parent, bool wholeword, Qt::CaseSensitivity casesensitive);
```

```
555
556
567 bool isCompatibleVersion(int fileversionpart1, int fileversionpart2, int fileversionpart3, int
      lastcompatiblepart1, int lastcompatiblepart2, int lastcompatiblepart3);
568
569
577 string fixLabelCharge(string& str, int sourcecharge, int targetcharge);
578
579
580 #endif
581
```

17.43 gui/cAboutWidget.h File Reference

The implementation of the dialog 'Help->About'.

```
#include <QWidget>
#include <QFileInfo>
```

Classes

- class cAboutWidget

  The widget representing the dialog 'About'.

17.43.1 Detailed Description

The implementation of the dialog 'Help->About'.

17.44 cAboutWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CABOUTWIDGET_H
8 #define _CABOUTWIDGET_H
9
10 #include <QWidget>
11 #include <QFileInfo>
12
13
14 // forward declaration
15 class QVBoxLayout;
16 class QTextBrowser;
17 class QDialogButtonBox;
18
19
23 class cAboutWidget :  public QWidget
24 {
25     Q_OBJECT
26
27 public:
28
29
34     cAboutWidget(QWidget* parent = (QWidget *)0);
35
36
40     ~cAboutWidget();
41
42
47     void closeEvent(QCloseEvent *event);
48
49
50 protected:
51
52
57     void keyPressEvent(QKeyEvent *event);
58
59
60 private:
61     QWidget* parent;
62     QVBoxLayout* layout;
63     QTextBrowser* message;
64     QDialogButtonBox* buttonbox;
65
66
```

```
67 private slots:
68
69     void accept();
70
71 };
72
73 #endif
```

17.45 gui/cBranchCyclicWidget.h File Reference

Visualization of a branch-cyclic peptide.

```
#include <math.h>
#include <QWidget>
#include <QPalette>
#include "core/cTheoreticalSpectrum.h"
#include "gui/cLinearWidget.h"
#include "gui/cCyclicWidget.h"
```

Classes

- class cBranchCyclicWidget

  Visualization of a branch-cyclic peptide.

Functions

- void generateBranchLabelsDown (bool nterminal, int rotationid, unordered_set< cIonLabel, hash_cIonLabel > &labels, cParameters *parameters, cTheoreticalSpectrum *theoreticalspectrum, int centerx, int topmargin, int horizontalstep, int verticalstep, string visibleionseries, string visibleneutralloss, int visiblerotationid, int branchstart)

  Generate labels of fragment ions on a branch of a branch-cyclic peptide (direction down).
- void generateBranchLabelsUp (bool nterminal, int rotationid, unordered_set< cIonLabel, hash_cIonLabel > &labels, cParameters *parameters, cTheoreticalSpectrum *theoreticalspectrum, int centerx, int topmargin, int horizontalstep, int verticalstep, string visibleionseries, string visibleneutralloss, int visiblerotationid, int branchend, int fragmentend)

  Generate labels of fragment ions on a branch of a branch-cyclic peptide (direction up).

17.45.1 Detailed Description

Visualization of a branch-cyclic peptide.

17.45.2 Function Documentation

17.45.2.1 generateBranchLabelsDown()

```
void generateBranchLabelsDown (
            bool nterminal,
            int rotationid,
            unordered_set< cIonLabel, hash_cIonLabel > & labels,
            cParameters * parameters,
            cTheoreticalSpectrum * theoreticalspectrum,
            int centerx,
            int topmargin,
            int horizontalstep,
            int verticalstep,
            string visibleionseries,
            string visibleneutralloss,
```

```
            int visiblerotationid,
            int branchstart )
```

Generate labels of fragment ions on a branch of a branch-cyclic peptide (direction down).

Parameters

nterminal	if true only nterminal fragment ions are drawn, if false only cterminal fragment ions are drawn
rotationid	id of a sequence rotation
labels	labels of fragment ions
parameters	a pointer to parameters of the application
theoreticalspectrum	a theoretical spectrum
centerx	x coordinate of a center of the ring
topmargin	top margin in pixels
horizontalstep	horizontal step (determine width of blocks)
verticalstep	vertical step (determine height of blocks)
visibleionseries	name of a ion series to be visualized
visibleneutralloss	name of a neutral loss to be visualized
visiblerotationid	id of a rotation whose fragment ions are visualized
branchstart	starting position of a branch

17.45.2.2 generateBranchLabelsUp()

```
void generateBranchLabelsUp (
            bool nterminal,
            int rotationid,
            unordered_set< cIonLabel, hash_cIonLabel > & labels,
            cParameters * parameters,
            cTheoreticalSpectrum * theoreticalspectrum,
            int centerx,
            int topmargin,
            int horizontalstep,
            int verticalstep,
            string visibleionseries,
            string visibleneutralloss,
            int visiblerotationid,
            int branchend,
            int fragmentend )
```

Generate labels of fragment ions on a branch of a branch-cyclic peptide (direction up).

Parameters

nterminal	if true only nterminal fragment ions are drawn, if false only cterminal fragment ions are drawn
rotationid	id of a sequence rotation
labels	labels of fragment ions
parameters	a pointer to parameters of the application
theoreticalspectrum	a theoretical spectrum
centerx	x coordinate of a center of the ring
topmargin	top margin in pixels
horizontalstep	horizontal step (determine width of blocks)
verticalstep	vertical step (determine height of blocks)
visibleionseries	name of a ion series to be visualized

Parameters

visibleneutralloss	name of a neutral loss to be visualized
visiblerotationid	id of a rotation whose fragment ions are visualized
branchend	end position of a branch
fragmentend	last fragment to draw

17.46 cBranchCyclicWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CBRANCHCYCLICWIDGET_H
8 #define _CBRANCHCYCLICWIDGET_H
9
10 #include <math.h>
11 #include <QWidget>
12 #include <QPalette>
13
14 #include "core/cTheoreticalSpectrum.h"
15 #include "gui/cLinearWidget.h"
16 #include "gui/cCyclicWidget.h"
17
18
19 // forward declaration
20 class QPaintEvent;
21
22
39 void generateBranchLabelsDown(bool nterminal, int rotationid, unordered_set<cIonLabel, hash_cIonLabel>&
      labels, cParameters* parameters, cTheoreticalSpectrum* theoreticalspectrum, int centerx, int
      topmargin, int horizontalstep, int verticalstep, string visibleionseries, string visibleneutralloss,
      int visiblerotationid, int branchstart);
40
41
59 void generateBranchLabelsUp(bool nterminal, int rotationid, unordered_set<cIonLabel, hash_cIonLabel>&
      labels, cParameters* parameters, cTheoreticalSpectrum* theoreticalspectrum, int centerx, int
      topmargin, int horizontalstep, int verticalstep, string visibleionseries, string visibleneutralloss,
      int visiblerotationid, int branchend, int fragmentend);
60
61
65 class cBranchCyclicWidget :  public QWidget
66 {
67     Q_OBJECT
68
69 public:
70
71
75     cBranchCyclicWidget();
76
77
83     void initialize(cParameters* parameters, cTheoreticalSpectrum* theoreticalspectrum);
84
85
91     void exportToPDF(QString filename, bool postscript);
92
93
98     void exportToPNG(QString filename);
99
100
105     void exportToSVG(QString filename);
106
107
112     void setReportIsomers(bool reportisomers);
113
114
119     void hidePeakLabels(bool state);
120
121
122 protected:
123
124
129     void paintEvent(QPaintEvent *event);
130
131
132 private:
133
134     void paint(QPainter& painter);
135
136     cParameters* parameters;
137     cTheoreticalSpectrum* theoreticalspectrum;
```

```
138
139     string visibleionseries;
140     string visibleneutralloss;
141     int visiblerotationid;
142     int visibletrotationid;
143
144     bool reportisomers;
145     bool hidelabels;
146
147
148 private slots:
149
150     void ionSeriesChanged(QString text);
151
152     void neutralLossChanged(QString text);
153
154     void rotationChanged(int index);
155
156     void trotationChanged(int index);
157
158 };
159
160 #endif
```

17.47 gui/cBranchedWidget.h File Reference

Visualization of a branched peptide.

```
#include <QWidget>
#include "core/cTheoreticalSpectrum.h"
#include "gui/cLinearWidget.h"
```

Classes

- class cBranchedWidget

  Visualization of a branched peptide.

17.47.1 Detailed Description

Visualization of a branched peptide.

17.48 cBranchedWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CBRANCHEDWIDGET_H
8 #define _CBRANCHEDWIDGET_H
9
10 #include <QWidget>
11
12 #include "core/cTheoreticalSpectrum.h"
13 #include "gui/cLinearWidget.h"
14
15 // forward declaration
16 class QPaintEvent;
17
18
22 class cBranchedWidget :  public QWidget
23 {
24     Q_OBJECT
25
26 public:
27
28
32     cBranchedWidget();
33
34
40     void initialize(cParameters* parameters, cTheoreticalSpectrum* theoreticalspectrum);
41
42
48     void exportToPDF(QString filename, bool postscript);
49
50
55     void exportToPNG(QString filename);
```

```
56
57
62     void exportToSVG(QString filename);
63
64
69     void setReportIsomers(bool reportisomers);
70
71
76     void hidePeakLabels(bool state);
77
78
79 protected:
80
81
86     void paintEvent(QPaintEvent *event);
87
88
89 private:
90
91     void paint(QPainter& painter);
92
93     cParameters* parameters;
94     cTheoreticalSpectrum* theoreticalspectrum;
95
96     string visibleionseries;
97     string visibleneutralloss;
98     int visibletrotationid;
99
100     bool reportisomers;
101     bool hidelabels;
102
103
104 private slots:
105
106     void ionSeriesChanged(QString text);
107
108     void neutralLossChanged(QString text);
109
110     void trotationChanged(int index);
111
112 };
113
114 #endif
```

17.49 gui/cBricksDatabaseProxyModel.h File Reference

A proxy model for editor of database of building blocks.

```
#include <QString>
#include <QVariant>
#include <QComboBox>
#include <QSortFilterProxyModel>
#include <QLineEdit>
#include "core/utilities.h"
```

Classes

- class cBricksDatabaseProxyModel

  A proxy model for editor of database of building blocks.

17.49.1 Detailed Description

A proxy model for editor of database of building blocks.

17.50 cBricksDatabaseProxyModel.h

Go to the documentation of this file.

```
1
7 #ifndef _CBRICKSDATABASEPROXYMODEL_H
8 #define _CBRICKSDATABASEPROXYMODEL_H
9
10 #include <QString>
```

```
11 #include <QVariant>
12 #include <QComboBox>
13 #include <QSortFilterProxyModel>
14 #include <QLineEdit>
15 #include "core/utilities.h"
16
17
21 class cBricksDatabaseProxyModel :  public QSortFilterProxyModel {
22
23     Q_OBJECT
24
25 private:
26
27     QComboBox* filteroperator;
28     QComboBox* filtercombobox1;
29     QComboBox* filtercomparatorcombobox1;
30     QLineEdit* filterline1;
31     QComboBox* filtercombobox2;
32     QComboBox* filtercomparatorcombobox2;
33     QLineEdit* filterline2;
34     bool wholeword;
35
36 public:
37
38
43     cBricksDatabaseProxyModel(QObject *parent = 0);
44
45
56     void initialize(QComboBox* rowsfilteroperator, QComboBox* rowsfiltercombobox1, QComboBox*
       rowsfiltercomparatorcombobox1, QLineEdit* rowsfilterline1, QComboBox* rowsfiltercombobox2, QComboBox*
       rowsfiltercomparatorcombobox2, QLineEdit* rowsfilterline2);
57
58
63     void setWholeWord(bool wholeword);
64
65
66 protected:
67
68
75     bool filterAcceptsRow(int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE;
76
77
84     bool lessThan(const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE;
85
86
87 };
88
89
90 #endif
91
```

17.51 gui/cBricksDatabaseWidget.h File Reference

An editor of building blocks database.

```
#include <QWidget>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include <QMainWindow>
#include <QToolBar>
#include <QAction>
#include <QTableView>
#include <QStandardItemModel>
#include <QStandardItem>
#include <QComboBox>
#include <fstream>
#include "core/utilities.h"
#include "core/cGlobalPreferences.h"
#include "core/cBricksDatabase.h"
#include "gui/cBricksDatabaseProxyModel.h"
#include "gui/cMultipleButtonDelegate.h"
#include "gui/cCheckBoxDelegate.h"
```

Classes

- class cBricksDatabaseWidget

  An editor of building blocks database.

Functions

- int numberOfOccurrences (const string &s, char c)

  Number of occurrences of a char in a string.

17.51.1 Detailed Description

An editor of building blocks database.

17.51.2 Function Documentation

17.51.2.1 numberOfOccurrences()

```
int numberOfOccurrences (
            const string & s,
            char c )
```

Number of occurrences of a char in a string.

Parameters

s	string
c	char

Return values

int	number of occurrences of *c* in *s*

17.52 cBricksDatabaseWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CBRICKSDATABASEWIDGET_H
8 #define _CBRICKSDATABASEWIDGET_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QUrl>
13 #include <QFileInfo>
14 #include <QMainWindow>
15 #include <QToolBar>
16 #include <QAction>
17 #include <QTableView>
18 #include <QStandardItemModel>
19 #include <QStandardItem>
20 #include <QComboBox>
21 #include <fstream>
22 #include "core/utilities.h"
23 #include "core/cGlobalPreferences.h"
24 #include "core/cBricksDatabase.h"
25 #include "gui/cBricksDatabaseProxyModel.h"
26 #include "gui/cMultipleButtonDelegate.h"
27 #include "gui/cCheckBoxDelegate.h"
28
29
30 // forward declaration
31 class QHBoxLayout;
32 class QVBoxLayout;
```

```
33 class QPushButton;
34 class QLineEdit;
35 class QMenuBar;
36 class QMenu;
37
38
45 int numberOfOccurrences(const string& s, char c);
46
47
51 class cBricksDatabaseWidget :  public QMainWindow
52 {
53     Q_OBJECT
54
55 public:
56
57
63     cBricksDatabaseWidget(cGlobalPreferences* globalpreferences, QWidget* parent = (QWidget *)0);
64
65
69     ~cBricksDatabaseWidget();
70
71
76     void closeEvent(QCloseEvent *event);
77
78
83     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
84
85
86 private:
87
88     QString editorname;
89
90     QWidget* parent;
91
92     cGlobalPreferences* globalpreferences;
93
94     QMenuBar* menuBar;
95     QMenu* menuFile;
96     QMenu* menuEdit;
97     QMenu* menuHelp;
98
99     QToolBar* toolbarFile;
100     QAction* actionNewDatabase;
101     QAction* actionOpenDatabase;
102     QAction* actionSaveDatabase;
103     QAction* actionSaveDatabaseAs;
104     QAction* actionImportDatabase;
105     QAction* actionCloseWindow;
106
107     QToolBar* toolbarEdit;
108     QAction* actionAddRow;
109     QAction* actionRemoveSelectedRows;
110     QAction* actionSelectAll;
111     QAction* actionUnselectAll;
112
113     QToolBar* toolbarHelp;
114     QAction* actionHTMLDocumentation;
115
116     QToolBar* toolbarFilter;
117     QWidget* rowsfilterwidget;
118     QHBoxLayout* rowsfilterhbox;
119     QComboBox* rowsfilteroperator;
120     QComboBox* rowsfiltercombobox1;
121     QComboBox* rowsfiltercomparatorcombobox1;
122     QLineEdit* rowsfilterline1;
123     QComboBox* rowsfiltercombobox2;
124     QComboBox* rowsfiltercomparatorcombobox2;
125     QLineEdit* rowsfilterline2;
126     QCheckBox* rowsfiltercasesensitive;
127     QCheckBox* rowsfilterwholeword;
128     QPushButton* rowsfilterbutton;
129     QPushButton* rowsfilterclearbutton;
130
131     QTableView* database;
132     QStandardItemModel* databasemodel;
133     cBricksDatabaseProxyModel* proxymodel;
134     QVBoxLayout* mainlayout;
135     QWidget* mainwidget;
136
137     QString databasefile;
138     QString lastdir;
139     ifstream inputstream;
140     ofstream outputstream;
141     cBricksDatabase bricks;
142
143     bool datamodified;
144
```

```
145     void deleteTable();
146
147     void resetHeader();
148
149     bool checkTable();
150
151     bool checkFormula(int row, cSummaryFormula& formula);
152
153     void setDataModified(bool datamodified);
154
155 protected:
156
157
162     void keyPressEvent(QKeyEvent *event);
163
164
165 private slots:
166
167     void closeWindow();
168
169     void openDatabase();
170
171     bool saveDatabase();
172
173     bool saveDatabaseAs();
174
175     void saveDatabaseCheck();
176
177     void addRow();
178
179     void removeSelectedRows();
180
181     void itemChanged(QStandardItem* item);
182
183     void headerItemClicked(int index);
184
185     void filterRows();
186
187     void resetFilter();
188
189     void createNewDatabase();
190
191     void importDatabase();
192
193     void selectAll();
194
195     void unselectAll();
196
197     void showHTMLDocumentation();
198
199 };
200
201 #endif
202
```

17.53 gui/cCalibrationChart.h File Reference

Calibration chart.

```
#include <QWidget>
#include <QDesktopServices>
#include <QToolBar>
#include <QMenuBar>
#include <QMenu>
#include <QUrl>
#include <QFileInfo>
#include <QMainWindow>
#include <QStandardItemModel>
#include <QStandardItem>
#include <QHeaderView>
#include <QItemDelegate>
#include <QApplication>
#include "core/cGlobalPreferences.h"
#include "gui/cCalibrationChartScene.h"
#include "gui/cEventFilter.h"
#include "gui/cCalibrationChartProxyModel.h"
```

Classes

- class cCalibrationChart

  Calibration chart.

17.53.1 Detailed Description

Calibration chart.

17.54 cCalibrationChart.h

Go to the documentation of this file.

```
1
7 #ifndef _CCALIBRATIONCHART_H
8 #define _CCALIBRATIONCHART_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QToolBar>
13 #include <QMenuBar>
14 #include <QMenu>
15 #include <QUrl>
16 #include <QFileInfo>
17 #include <QMainWindow>
18 #include <QStandardItemModel>
19 #include <QStandardItem>
20 #include <QHeaderView>
21 #include <QItemDelegate>
22 #include <QApplication>
23 #include "core/cGlobalPreferences.h"
24 #include "gui/cCalibrationChartScene.h"
25 #include "gui/cEventFilter.h"
26 #include "gui/cCalibrationChartProxyModel.h"
27
28
29 // forward declaration
30 class QHBoxLayout;
31 //class QVBoxLayout;
32 class QPushButton;
33 class QSplitter;
34 class QDoubleSpinBox;
35 class QAction;
36 class QLabel;
37 class QTableView;
38
39
43 class cCalibrationChart :  public QMainWindow
44 {
45     Q_OBJECT
46
47 public:
48
49
55     cCalibrationChart(cGlobalPreferences* globalpreferences, QWidget *parent);
56
57
61     ~cCalibrationChart();
62
63
68     void closeEvent(QCloseEvent *event);
69
70
75     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
76
77
87     void setLineParameters(int equationtype, double a, double b, vector<double> xvalues, vector<double>
      yvalues, vector<double> sd);
88
89
98     void setData(vector<double> xvalues, vector<double> yvalues, vector<double> sd, vector<string>
      datagroups, vector<int>& datatimevalues);
99
100
114     void createTable(double a, double b, vector<double>& calibrationxvalues, vector<double>&
      calibrationyvalues, vector<double>& calibrationsd, vector<string>& datagroups, vector<double>&
      dataxvalues, vector<double>& datayvalues, vector<double>& datasd, vector<int>& datatimevalues);
115
```

```
116
120     void deleteTable();
121
122
123 protected:
124
125
130     void keyPressEvent(QKeyEvent *event);
131
132
133 private:
134
135     cGlobalPreferences* globalpreferences;
136     QWidget* parent;
137
138     QMenuBar* menuBar;
139     QMenu* menuFile;
140     QMenu* menuView;
141     QMenu* menuHelp;
142
143     QToolBar* toolbarFile;
144     QToolBar* toolbarView;
145     QToolBar* toolbarHelp;
146     QToolBar* toolbarGraph;
147
148     QAction* actionExportCSV;
149     QAction* actionExportSpectrum;
150     QAction* actionCloseWindow;
151     QAction* actionZoomIn;
152     QAction* actionZoomOut;
153     QAction* actionZoomReset;
154     QAction* actionHideLabels;
155     //QAction* actionMouseXValueSelection;
156     QAction* actionHTMLDocumentation;
157
158     QWidget* widgetgraphtype;
159     QHBoxLayout* hboxgraphtype;
160     QLabel* labelgraphtype;
161     QComboBox* comboboxgraphtype;
162
163     QWidget* widgetxvalue;
164     QHBoxLayout* hboxxvalue;
165     QLabel* labelxvalue;
166     QDoubleSpinBox* minxvalue;
167     QLabel* labelseparator;
168     QDoubleSpinBox* maxxvalue;
169     QPushButton* setxvalueinterval;
170     QPushButton* resetxvalueinterval;
171
172     cCalibrationChartScene* chartscene;
173
174     QTableView* database;
175     QStandardItemModel* databasemodel;
176     cCalibrationChartProxyModel* proxymodel;
177
178     QSplitter* vsplitter;
179
180     QString lastdirexporttocsv;
181
182 signals:
183
184
190     void emitXValueInterval(double minxvalue, double maxxvalue);
191
192
193 private slots:
194
195
196     void exportToCsv();
197
198
199     void updateXValueInterval(double minxvalue, double maxxvalue);
200
201
202     void setXValueInterval();
203
204
205     void graphTypeChanged(int type);
206
207
208     void exportScene();
209
210
211     void hideLabels(bool state);
212
213
214     void showHTMLDocumentation();
```

```
215
216
217     void closeWindow();
218
219 };
220
221 #endif
222
```

17.55 gui/cCalibrationChartProxyModel.h File Reference

A table proxy model.

```
#include <QString>
#include <QVariant>
#include <QSortFilterProxyModel>
#include <QComboBox>
#include <QLineEdit>
#include "core/cParameters.h"
```

Classes

- class cCalibrationChartProxyModel

  A table proxy model.

17.55.1 Detailed Description

A table proxy model.

17.56 cCalibrationChartProxyModel.h

Go to the documentation of this file.

```
1
7 #ifndef _CCALIBRATIONCHARTPROXYMODEL_H
8 #define _CCALIBRATIONCHARTPROXYMODEL_H
9
10 #include <QString>
11 #include <QVariant>
12 #include <QSortFilterProxyModel>
13 #include <QComboBox>
14 #include <QLineEdit>
15 #include "core/cParameters.h"
16
17
21 class cCalibrationChartProxyModel :  public QSortFilterProxyModel {
22
23     Q_OBJECT
24
25 public:
26
31     cCalibrationChartProxyModel(QObject *parent = 0);
32
33 protected:
34
41     bool lessThan(const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE;
42
43 };
44
45
46 #endif
47
```

17.57 gui/cCalibrationChartScene.h File Reference

Calibration chart scene.

```
#include <iostream>
#include <cmath>
#include <deque>
```

```
#include <QWidget>
#include <QGraphicsView>
#include <QRectF>
```

Classes

- class cCalibrationChartScene

  The implementation of an isotopic pattern simulator (graphics).

Functions

- double calculateR2Value (double a, double b, vector< double > xvalues, vector< double > yvalues)

  Calculate the coefficient of determination R^2.
- double calculateLOD (double a, vector< double > xvalues, vector< double > yvalues)

  Calculate limit of detection.
- double calculateLOQ (double a, vector< double > xvalues, vector< double > yvalues)

  Calculate limit of quantitation.

17.57.1 Detailed Description

Calibration chart scene.

17.57.2 Function Documentation

17.57.2.1 calculateLOD()

```
double calculateLOD (
            double a,
            vector< double > xvalues,
            vector< double > yvalues )
```

Calculate limit of detection.

Parameters

a	a
xvalues	x coordinates of points
yvalues	y coordinates of points

Return values

double	limit of detection

17.57.2.2 calculateLOQ()

```
double calculateLOQ (
            double a,
            vector< double > xvalues,
            vector< double > yvalues )
```

Calculate limit of quantitation.

Parameters

a	a
xvalues	x coordinates of points
yvalues	y coordinates of points

Return values

double	limit of quantitation

17.57.2.3 calculateR2Value()

```
double calculateR2Value (
            double a,
            double b,
            vector< double > xvalues,
            vector< double > yvalues )
```

Calculate the coefficient of determination R^2.

Parameters

a	a
b	b
xvalues	x coordinates of points
yvalues	y coordinates of points

Return values

double	R^2

17.58 cCalibrationChartScene.h

Go to the documentation of this file.

```
1
7 #ifndef _CCALIBRATIONCHARTSCENE_H
8 #define _CCALIBRATIONCHARTSCENE_H
9
10 #include <iostream>
11 #include <cmath>
12 #include <deque>
13 #include <QWidget>
14 #include <QGraphicsView>
15 #include <QRectF>
16
17
18 using namespace std;
19
20
21 // forward declaration
22 class QPaintEvent;
23 class QGraphicsScene;
24 class QMatrix;
25 class QGraphicsItemGroup;
26 class QGraphicsRectItem;
27 class QGraphicsSimpleTextItem;
28
29
38 double calculateR2Value(double a, double b, vector<double> xvalues, vector<double> yvalues);
39
40
48 double calculateLOD(double a, vector<double> xvalues, vector<double> yvalues);
```

```
49
50
58 double calculateLOQ(double a, vector<double> xvalues, vector<double> yvalues);
59
60
64 class cCalibrationChartScene :  public QGraphicsView
65 {
66     Q_OBJECT
67
68 public:
69
70
75     cCalibrationChartScene(QWidget* parent = 0);
76
77
81     ~cCalibrationChartScene();
82
83
87     void initialize();
88
89
95     void exportToPDF(QString filename, bool postscript);
96
97
102     void exportToSVG(QString filename);
103
104
109     void exportToPNG(QString filename);
110
111
116     void hideLabels(bool state);
117
118
128     void setLineParameters(int equationtype, double a, double b, vector<double> xvalues, vector<double>
    yvalues, vector<double> sd);
129
130
139     void setData(vector<double> datax, vector<double> datay, vector<double> datasd, vector<string>
    datagroups, vector<int> datatimevalues);
140
141
146     void setGraphType(int type);
147
148
153     double getMaximumConcentration();
154
155
160     int getMaximumCollectionTime();
161
162
163 protected:
164
165
170     void wheelEvent(QWheelEvent *event);
171
172
177     void mouseMoveEvent(QMouseEvent *event);
178
179
184     void mouseReleaseEvent(QMouseEvent *event);
185
186
191     void mousePressEvent(QMouseEvent *event);
192
193
198     void showEvent(QShowEvent *event);
199
200
205     void resizeEvent(QResizeEvent *event);
206
207
208 private:
209
210     QWidget* parent;
211     QGraphicsScene* scene;
212     QMatrix originalmatrix;
213
214     QGraphicsItemGroup* zoomgroup;
215     QGraphicsRectItem* zoomrect;
216     QGraphicsSimpleTextItem* zoomsimpletextitem;
217
218     QGraphicsSimpleTextItem* cursorsimpletextitem;
219
220     //cPeaksList isotopicpattern;
221
222     int graphtype;
223
```

```
224     qreal currentscale;
225     qreal factor;
226
227     int origwidth;
228     int origheight;
229
230     bool calledbyresizeevent;
231     deque<int> oldwidth;
232     deque<int> oldheight;
233
234     int pressedx;
235     int pressedy;
236
237     int currentx;
238     int currenty;
239
240     double minxvalue;
241     double maxxvalue;
242
243     int topmargin;
244     int bottommargin;
245     int leftmargin;
246     int rightmargin;
247
248     bool firstshow;
249     bool enablemousexvalueselection;
250
251     int equationtype;
252
253     double a;
254     double b;
255
256     vector<double> xvalues;
257     vector<double> yvalues;
258
259     vector<double> sd;
260
261     vector<double> datax;
262     vector<double> datay;
263
264     vector<double> datasd;
265     vector<string> datagroups;
266
267     vector<int> datatimevalues;
268
269     bool hidelabels;
270
271
272     double getValueFromXPosition(int x, int w);
273
274
275     int getXPositionFromValue(double value, int w);
276
277
278     double getValueFromYPosition(int y, int h);
279
280
281     int getYPositionFromValue(double intensity, int h);
282
283
284     void redrawScene();
285
286
287     void updateZoomGroup();
288
289
290     void calculateMinMaxXValue();
291
292
293     double getMaximumYValue();
294
295
296 signals:
297
298
304     void updateXValueInterval(double minxvalue, double maxxvalue);
305
306
307 private slots:
308
309
310     void zoomIn();
311
312
313     void zoomOut();
314
315
```

```
316     void normalSize();
317
318
319     void setXValueInterval(double minxvalue, double maxxvalue);
320
321
322     void resetXValueInterval();
323
324
325     //void enableMouseXValueSelectionTool(bool enable);
326
327 };
328
329 #endif
```

17.59 gui/cCheckBoxDelegate.h File Reference

A checkbox item delegate.

```
#include <QApplication>
#include <QString>
#include <QVariant>
#include <QStyledItemDelegate>
#include <QPainter>
#include <QModelIndex>
#include <QEvent>
#include <QAbstractItemModel>
#include <QCheckBox>
#include <QMouseEvent>
```

Classes

- class cCheckBoxDelegate

  A checkbox item delegate.

17.59.1 Detailed Description

A checkbox item delegate.

17.60 cCheckBoxDelegate.h

Go to the documentation of this file.

```
1
7 #ifndef _CCHECKBOXDELEGATE_H
8 #define _CCHECKBOXDELEGATE_H
9
10 #include <QApplication>
11 #include <QString>
12 #include <QVariant>
13 #include <QStyledItemDelegate>
14 #include <QPainter>
15 #include <QModelIndex>
16 #include <QEvent>
17 #include <QAbstractItemModel>
18 #include <QCheckBox>
19 #include <QMouseEvent>
20
21
25 class cCheckBoxDelegate :  public QStyledItemDelegate {
26
27     Q_OBJECT
28
29 public:
30
31
36     cCheckBoxDelegate(QObject* parent = (QObject *)0);
37
38
45     void paint(QPainter* painter, const QStyleOptionViewItem& option, const QModelIndex& index) const;
46
```

```
47
48 protected:
49
50
58     bool editorEvent(QEvent* event, QAbstractItemModel* model, const QStyleOptionViewItem& option, const
       QModelIndex& index);
59
60
61 };
62
63
64 #endif
65
```

17.61 gui/cChromatogramWindow.h File Reference

Chromatogram window.

```
#include <QMainWindow>
#include <QDesktopServices>
#include <QToolBar>
#include <QMenuBar>
#include <QMenu>
#include <QAction>
#include <QSpinBox>
#include <QHBoxLayout>
#include <QFileDialog>
#include <QLabel>
#include <QPushButton>
#include <QComboBox>
#include "core/utilities.h"
#include "core/cGlobalPreferences.h"
#include "gui/cChromatogramWindowWidget.h"
```

Classes

- class cChromatogramWindow

  Chromatogram window.

17.61.1 Detailed Description

Chromatogram window.

17.62 cChromatogramWindow.h

Go to the documentation of this file.

```
1
7 #ifndef _CCHROMATOGRAMWINDOW_H
8 #define _CCHROMATOGRAMWINDOW_H
9
10 #include <QMainWindow>
11 #include <QDesktopServices>
12 #include <QToolBar>
13 #include <QMenuBar>
14 #include <QMenu>
15 #include <QAction>
16 #include <QSpinBox>
17 #include <QHBoxLayout>
18 #include <QFileDialog>
19 #include <QLabel>
20 #include <QPushButton>
21 #include <QComboBox>
22 #include "core/utilities.h"
23 #include "core/cGlobalPreferences.h"
24 #include "gui/cChromatogramWindowWidget.h"
25
26
30 class cChromatogramWindow :  public QMainWindow
```

```
31 {
32     Q_OBJECT
33
34 public:
35
36
43     cChromatogramWindow(cGlobalPreferences* globalpreferences, cTheoreticalSpectrumList&
       listoftheoreticalspectra, QWidget* parent = (QWidget *)0);
44
45
49     ~cChromatogramWindow();
50
51
56     void closeEvent(QCloseEvent *event);
57
58
63     void recalculateTICChromatogram(int activefileid);
64
65
70     void setAbsoluteIntensityEnabled(bool enabled);
71
72
77     void setPeakShape(int peakshape);
78
79
84     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
85
86
87 private:
88
89     cGlobalPreferences* globalpreferences;
90     cTheoreticalSpectrumList* listoftheoreticalspectra;
91     QWidget* parent;
92
93     QMenuBar* menuBar;
94     QMenu* menuFile;
95     QMenu* menuView;
96     QMenu* menuHelp;
97
98     QToolBar* toolbarFile;
99     QAction* actionExportImage;
100     QAction* actionCloseWindow;
101
102     QToolBar* toolbarView;
103     QAction* actionZoomIn;
104     QAction* actionZoomOut;
105     QAction* actionZoomReset;
106     QAction* actionMouseSelection;
107     QAction* actionRetentionTime;
108     QAction* actionAbsoluteIntensity;
109     QAction* actionRawData;
110     QAction* actionHideTIC;
111     QAction* actionHideEIC;
112     QAction* actionHideLabels;
113
114     QWidget* peakshapewidget;
115     QHBoxLayout* peakshapehboxlayout;
116     QLabel* peakshapelabel;
117     QComboBox* comboboxpeakshape;
118
119     QToolBar* toolbarTime;
120
121     QWidget* widgetretentiontime;
122     QHBoxLayout* hboxretentiontime;
123     QLabel* labelretentiontime;
124     QDoubleSpinBox* minretentiontime;
125     QLabel* labelseparatorretentiontime;
126     QDoubleSpinBox* maxretentiontime;
127     QPushButton* setretentiontimeinterval;
128     QPushButton* resetretentiontimeinterval;
129
130     QWidget* widgetscanid;
131     QHBoxLayout* hboxscanid;
132     QLabel* labelscanid;
133     QSpinBox* minscanid;
134     QLabel* labelseparatorscanid;
135     QSpinBox* maxscanid;
136     QPushButton* setscanidinterval;
137     QPushButton* resetscanidinterval;
138     QPushButton* sumofspectra;
139
140     QToolBar* toolbarHelp;
141     QAction* actionHTMLDocumentation;
142
143     cChromatogramWindowWidget* chromatogramwindowwidget;
144
145     QString lastdirexportimage;
```

```
146
147
148 protected:
149
150
155     void keyPressEvent(QKeyEvent *event);
156
157
158 signals:
159
160
166     void emitRetentionTimeInterval(double mintime, double maxtime);
167
168
174     void emitScanIDInterval(int minid, int maxid);
175
176
181     void doubleClickedScanIDSignal(int scanid);
182
183
189     void calculateAvgSpectrum(int minid, int maxid);
190
191
192 private slots:
193
194
195     void exportImage();
196
197
198     void closeWindow();
199
200
201     void updateRetentionTimeInterval(double mintime, double maxtime);
202
203
204     void setRetentionTimeInterval();
205
206
207     void updateScanIDInterval(int minid, int maxid);
208
209
210     void setScanIDInterval();
211
212
213     void setFilterOptionsSlot(cPeaksList eic);
214
215
216     void showHTMLDocumentation();
217
218
219     void chromatogramDoubleClickedSlot(int scanid);
220
221
222     void showAvgSpectrum();
223
224 };
225
226 #endif
```

17.63 gui/cChromatogramWindowWidget.h File Reference

Chromatogram window widget.

```
#include <QWidget>
#include <QGraphicsView>
#include <QGraphicsSimpleTextItem>
#include <QScrollBar>
#include <QMessageBox>
#include <QPrinter>
#include <QSvgGenerator>
#include <QImage>
#include <QMouseEvent>
#include <QProgressDialog>
#include "core/utilities.h"
#include "core/cTheoreticalSpectrumList.h"
#include "gui/cEventFilter.h"
#include "gui/cPatternSimulatorSceneWidget.h"
```

Classes

- class cChromatogramWindowWidget

  Chromatogram window widget.

Functions

- double getGaussianArea (double sigma, double height)

  Get area under Gaussian function.
- double getGaussianAreas (vector< double > &sigmas, vector< double > &heights)

  Get area of EIC peaks (Gaussian function).
- double getExponentiallyModifiedGaussianArea (double x, double h, double ny, double sigma, double tau, bool fronting)

  Get area under Exponentially Modified Gaussian function.
- double getExponentiallyModifiedGaussianAreas (vector< double > &xvalues, vector< double > &heights, vector< double > &nys, vector< double > &sigmas, vector< double > &taus, bool fronting)

  Get area of EIC peaks (Exponentially Modified Gaussian function).
- double erfcx (double z)

  Calculate erfcx(z).
- double erfcxinv (double z)

  Calculate erfcxinv(z).
- double computeExponentiallyModifiedGaussFunction (double x, double h, double ny, double sigma, double tau, bool fronting)

  Compute value of exponentially modified Gaussian function.
- double computeExponentialFunction (double x, double base, double tau, bool fronting)

  Compute value of exponential function.
- void calculateCaruana (vector< double > &x, vector< double > &y, double &ny, double &sigma, double &height)

  Calculate ny, sigma, and height using Caruana's algorithm.
- void calculateGuo (vector< double > &x, vector< double > &y, int iterations, double &ny, double &sigma, double &height)

  Calculate ny, sigma, and height using Guo's algorithm.
- void calculateBaseTau (vector< double > &x, vector< double > &y, int iterations, bool fronting, double ynorm, double &base, double &tau)

  Calculate base and tau for exponential distribution.
- void calculateTauUsingWeightedAverage (vector< double > &x, vector< double > &y, bool fronting, double &tau)

  Calculate tau for exponential distribution using weighted average.
- void calculateGaussianParameters (cPeaksList &eicchromatogram, vector< double > &rtimes, int timeunit, bool absoluteintensity, bool usert, vector< double > &nys, vector< double > &sigmas, vector< double > &as)

  Get parameters of Gaussians functions (nys, sigmas, and heights).
- void calculateExponentialParameters (cPeaksList &eicchromatogram, vector< double > &rtimes, int timeunit, bool absoluteintensity, bool usert, bool fronting, double ynorm, vector< double > &bases, vector< double > &taus)

  Get parameters of Exponential functions (bases and taus).

17.63.1 Detailed Description

Chromatogram window widget.

17.63.2 Function Documentation

17.63.2.1 calculateBaseTau()

```
void calculateBaseTau (
            vector< double > & x,
            vector< double > & y,
            int iterations,
            bool fronting,
            double ynorm,
            double & base,
            double & tau )
```

Calculate base and tau for exponential distribution.

Parameters

x	input vector of x-coordinates
y	input vector of y-coordinates
iterations	number of iterations
fronting	true = fronting enabled, false = tailing enabled
ynorm	y axis normalization value
base	output value of base
tau	output value of tau

17.63.2.2 calculateCaruana()

```
void calculateCaruana (
            vector< double > & x,
            vector< double > & y,
            double & ny,
            double & sigma,
            double & height )
```

Calculate ny, sigma, and height using Caruana's algorithm.

Parameters

x	input vector of x-coordinates
y	input vector of y-coordinates
ny	output value of ny
sigma	output value of sigma
height	output value of height

17.63.2.3 calculateExponentialParameters()

```
void calculateExponentialParameters (
            cPeaksList & eicchromatogram,
            vector< double > & rtimes,
            int timeunit,
            bool absoluteintensity,
            bool usert,
            bool fronting,
            double ynorm,
            vector< double > & bases,
            vector< double > & taus )
```

Get parameters of Exponential functions (bases and taus).

Parameters

eicchromatogram	input EIC chromatogram
rtimes	input retention time vector
timeunit	RT time unit
absoluteintensity	use absolute intensities
usert	use RT instead of scan ids
fronting	true = fronting enabled, false = tailing enabled
ynorm	y axis normalization value
bases	output vector of bases
taus	output vector of taus

17.63.2.4 calculateGaussianParameters()

```
void calculateGaussianParameters (
            cPeaksList & eicchromatogram,
            vector< double > & rtimes,
            int timeunit,
            bool absoluteintensity,
            bool usert,
            vector< double > & nys,
            vector< double > & sigmas,
            vector< double > & as )
```

Get parameters of Gaussians functions (nys, sigmas, and heights).

Parameters

eicchromatogram	input EIC chromatogram
rtimes	input retention time vector
timeunit	RT time unit
absoluteintensity	use absolute intensities
usert	use RT instead of scan ids
nys	output vector of nys
sigmas	output vector of sigmas
as	output vector of heights

17.63.2.5 calculateGuo()

```
void calculateGuo (
            vector< double > & x,
            vector< double > & y,
            int iterations,
            double & ny,
            double & sigma,
            double & height )
```

Calculate ny, sigma, and height using Guo's algorithm.

Parameters

x	input vector of x-coordinates

Parameters

y	input vector of y-coordinates
iterations	number of iterations
ny	output value of ny
sigma	output value of sigma
height	output value of height

17.63.2.6 calculateTauUsingWeightedAverage()

```
void calculateTauUsingWeightedAverage (
            vector< double > & x,
            vector< double > & y,
            bool fronting,
            double & tau )
```

Calculate tau for exponential distribution using weighted average.

Parameters

x	input vector of x-coordinates
y	input vector of y-coordinates
fronting	true = fronting enabled, false = tailing enabled
tau	output value of tau

17.63.2.7 computeExponentialFunction()

```
double computeExponentialFunction (
            double x,
            double base,
            double tau,
            bool fronting )
```

Compute value of exponential function.

Parameters

x	x
base	base
tau	tau
fronting	true = fronting enabled, false = tailing enabled

Return values

double	f(x)

17.63.2.8 computeExponentiallyModifiedGaussFunction()

```
double computeExponentiallyModifiedGaussFunction (
            double x,
            double h,
```

```
            double ny,
            double sigma,
            double tau,
            bool fronting )
```

Compute value of exponentially modified Gaussian function.

Parameters

x	x
h	h
ny	ny
sigma	sigma
tau	tau
fronting	true = fronting enabled, false = tailing enabled

Return values

double	f(x)

17.63.2.9 erfcx()

```
double erfcx (
            double z )
```

Calculate erfcx(z).

Parameters

z	input value

Return values

erfcx(z)	

17.63.2.10 erfcxinv()

```
double erfcxinv (
            double z )
```

Calculate erfcxinv(z).

Parameters

z	input value

Return values

erfcxinv(z)	

17.63.2.11 getExponentiallyModifiedGaussianArea()

```
double getExponentiallyModifiedGaussianArea (
            double x,
            double h,
            double ny,
            double sigma,
            double tau,
            bool fronting )
```

Get area under Exponentially Modified Gaussian function.

Parameters

x	some value of x
h	height of Gaussian
ny	ny of Gaussian
sigma	sigma of Gaussian
tau	tau of Exponential
fronting	true = fronting enabled, false = tailing enabled

Return values

double	area

17.63.2.12 getExponentiallyModifiedGaussianAreas()

```
double getExponentiallyModifiedGaussianAreas (
            vector< double > & xvalues,
            vector< double > & heights,
            vector< double > & nys,
            vector< double > & sigmas,
            vector< double > & taus,
            bool fronting )
```

Get area of EIC peaks (Exponentially Modified Gaussian function).

Parameters

xvalues	vector of x values
heights	vector of heights of Gaussians
nys	vector of nys of Gaussians
sigmas	vector of sigmas of Gaussians
taus	vector of taus of Exponentials
fronting	true = fronting enabled, false = tailing enabled

Return values

double	area

17.63.2.13 getGaussianArea()

```
double getGaussianArea (
            double sigma,
            double height )
```

Get area under Gaussian function.

Parameters

sigma	sigma
height	height

Return values

double	area

17.63.2.14 getGaussianAreas()

```
double getGaussianAreas (
            vector< double > & sigmas,
            vector< double > & heights )
```

Get area of EIC peaks (Gaussian function).

Parameters

sigmas	input vector of sigmas
heights	input vector of heights

Return values

double	area

17.64 cChromatogramWindowWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CCHROMATOGRAMWINDOWWIDGET_H
8 #define _CCHROMATOGRAMWINDOWWIDGET_H
9
10 #include <QWidget>
11 #include <QGraphicsView>
12 #include <QGraphicsSimpleTextItem>
13 #include <QScrollBar>
14 #include <QMessageBox>
15 #include <QPrinter>
16 #include <QSvgGenerator>
17 #include <QImage>
18 #include <QMouseEvent>
19 #include <QProgressDialog>
20 #include "core/utilities.h"
21 #include "core/cTheoreticalSpectrumList.h"
22 #include "gui/cEventFilter.h"
23 #include "gui/cPatternSimulatorSceneWidget.h"
24
25
26 using namespace std;
27
28
35 double getGaussianArea(double sigma, double height);
36
37
```

```
44 double getGaussianAreas(vector<double>& sigmas, vector<double>& heights);
45
46
57 double getExponentiallyModifiedGaussianArea(double x, double h, double ny, double sigma, double tau, bool
       fronting);
58
59
70 double getExponentiallyModifiedGaussianAreas(vector<double>& xvalues, vector<double>& heights,
       vector<double>& nys, vector<double>& sigmas, vector<double>& taus, bool fronting);
71
72
78 double erfcx(double z);
79
80
86 double erfcxinv(double z);
87
88
99 double computeExponentiallyModifiedGaussFunction(double x, double h, double ny, double sigma, double tau,
       bool fronting);
100
101
110 double computeExponentialFunction(double x, double base, double tau, bool fronting);
111
112
121 void calculateCaruana(vector<double>& x, vector<double>& y, double& ny, double& sigma, double& height);
122
123
133 void calculateGuo(vector<double>& x, vector<double>& y, int iterations, double& ny, double& sigma,
       double& height);
134
135
146 void calculateBaseTau(vector<double>& x, vector<double>& y, int iterations, bool fronting, double ynorm,
       double& base, double& tau);
147
148
156 void calculateTauUsingWeightedAverage(vector<double>& x, vector<double>& y, bool fronting, double& tau);
157
158
170 void calculateGaussianParameters(cPeaksList& eicchromatogram, vector<double>& rtimes, int timeunit, bool
       absoluteintensity, bool usert, vector<double>& nys, vector<double>& sigmas, vector<double>& as);
171
172
185 void calculateExponentialParameters(cPeaksList& eicchromatogram, vector<double>& rtimes, int timeunit,
       bool absoluteintensity, bool usert, bool fronting, double ynorm, vector<double>& bases,
       vector<double>& taus);
186
187
191 class cChromatogramWindowWidget :  public QGraphicsView
192 {
193     Q_OBJECT
194
195 public:
196
197
203     cChromatogramWindowWidget(cTheoreticalSpectrumList& listoftheoreticalspectra, QWidget* parent = 0);
204
205
209     ~cChromatogramWindowWidget();
210
211
217     void exportToPDF(QString filename, bool postscript);
218
219
224     void exportToSVG(QString filename);
225
226
231     void exportToPNG(QString filename);
232
233
238     void recalculateTICChromatogram(int activefileid);
239
240
244     void setEICChromatogram(cPeaksList eic);
245
246
247 protected:
248
249
254     void wheelEvent(QWheelEvent *event);
255
256
261     void mouseMoveEvent(QMouseEvent *event);
262
263
268     void mouseReleaseEvent(QMouseEvent *event);
269
270
```

```
275     void mousePressEvent(QMouseEvent *event);
276
277
282     void mouseDoubleClickEvent(QMouseEvent *event);
283
284
289     void showEvent(QShowEvent *event);
290
291
296     void resizeEvent(QResizeEvent *event);
297
298
299 private:
300
301     cTheoreticalSpectrumList* listoftheoreticalspectra;
302     cPeaksList ticchromatogram;
303     cPeaksList eicchromatogram;
304     vector<double> rtimes;
305
306     QWidget* parent;
307     QGraphicsScene* scene;
308     QMatrix originalmatrix;
309
310     QGraphicsItemGroup* zoomgroup;
311     QGraphicsRectItem* zoomrect;
312     QGraphicsSimpleTextItem* zoomsimpletextitem;
313
314     int origwidth;
315     int origheight;
316
317     qreal currentscale;
318     qreal factor;
319
320     int pressedx;
321     int pressedy;
322
323     int currentx;
324     int currenty;
325
326     double mintime;
327     double maxtime;
328
329     int minscan;
330     int maxscan;
331
332     int topmargin;
333     int bottommargin;
334     int leftmargin;
335     int rightmargin;
336
337     bool firstshow;
338     bool enablemouseselection;
339
340     bool retentiontime;
341     bool absoluteintensity;
342     bool rawdatastate;
343     int peakshape;
344     bool hidetic;
345     bool hideeic;
346     bool hidelabels;
347
348
349     int getScanIDFromXPosition(int x, int w);
350
351
352     int getXPositionFromScanID(int scanid, int w);
353
354
355     void redrawScene();
356
357
358     void updateZoomGroup();
359
360
361     void calculateMinMaxScan();
362
363
364 signals:
365
371     void updateRetentionTimeInterval(double mintime, double maxtime);
372
373
379     void updateScanIDInterval(int minid, int maxid);
380
381
386     void chromatogramWidgetDoubleClicked(int scanid);
387
```

```
388
389 private slots:
390
391
392     void zoomIn();
393
394
395     void zoomOut();
396
397
398     void normalSize();
399
400
401     void retentionTimeStateChanged(bool state);
402
403
404     void absoluteIntensityStateChanged(bool state);
405
406
407     void rawDataStateChanged(bool state);
408
409
410     void setRetentionTimeInterval(double mintime, double maxtime);
411
412
413     void resetRetentionTimeInterval();
414
415
416     void setScanIDInterval(int minid, int maxid);
417
418
419     void resetScanIDInterval();
420
421
422     void enableMouseSelectionTool(bool enable);
423
424
425     void hideTIC(bool state);
426
427
428     void hideEIC(bool state);
429
430
431     void hideLabels(bool state);
432
433
434     void peakShapeChanged(int shape);
435
436 };
437
438 #endif
```

17.65 gui/cComboBoxDelegate.h File Reference

A combobox item delegate.

```
#include <QApplication>
#include <QString>
#include <QLocale>
#include <QVariant>
#include <QStyledItemDelegate>
#include <QPainter>
#include <QModelIndex>
#include <QEvent>
#include <QAbstractItemModel>
#include <QHBoxLayout>
#include <QMouseEvent>
#include <QStandardItemModel>
#include <QComboBox>
#include "core/cFragmentIons.h"
```

Classes

- class cComboBoxDelegate

A combobox item delegate.

17.65.1 Detailed Description

A combobox item delegate.

17.66 cComboBoxDelegate.h

Go to the documentation of this file.

```
1
7 #ifndef _CCOMBOBOXDELEGATE_H
8 #define _CCOMBOBOXDELEGATE_H
9
10 #include <QApplication>
11 #include <QString>
12 #include <QLocale>
13 #include <QVariant>
14 #include <QStyledItemDelegate>
15 #include <QPainter>
16 #include <QModelIndex>
17 #include <QEvent>
18 #include <QAbstractItemModel>
19 #include <QHBoxLayout>
20 #include <QMouseEvent>
21 #include <QStandardItemModel>
22 #include <QComboBox>
23 #include "core/cFragmentIons.h"
24
28 class cComboBoxDelegate :  public QStyledItemDelegate {
29
30     Q_OBJECT
31
32 public:
33
34
39     cComboBoxDelegate(QObject* parent = (QObject *)0);
40
41
48     void paint(QPainter* painter, const QStyleOptionViewItem& option, const QModelIndex& index) const;
49
50
56     void setEditorData(QWidget* editor, const QModelIndex& index) const;
57
58
65     void setModelData(QWidget* editor, QAbstractItemModel* model, const QModelIndex& index) const;
66
67
74     QWidget* createEditor(QWidget* parent, const QStyleOptionViewItem& option, const QModelIndex& index)
    const;
75
76 };
77
78
79 #endif
80
```

17.67 gui/cCyclicWidget.h File Reference

Visualization of a cyclic peptide.

```
#include <math.h>
#include <QWidget>
#include <QPalette>
#include "core/cTheoreticalSpectrum.h"
#include "gui/cLinearWidget.h"
```

Classes

- class cCyclicWidget

  Visualization of a cyclic peptide.

Functions

- void paintCircle (QPainter &painter, vector< string > &acronymsofblocks, int centerx, int centery, int radius, double angle, int horizontalstep, int linesize, int cornerlinesize, bool drawdashlines, int visiblerotationid, unordered_set< cIonLabel, hash_cIonLabel > &labels)

  Paint the circle of a cyclic or a branch-cyclic peptide.
- void generateCyclicLabelsToRight (bool nterminal, int rotationid, int rotationstart, int fragmentstart, int fragmentend, int numberofringblocks, unordered_set< cIonLabel, hash_cIonLabel > &labels, cParameters *parameters, cTheoreticalSpectrum *theoreticalspectrum, int centerx, int centery, int radius, double angle, int linesize, int cornerlinesize, string visibleionseries, string visibleneutralloss, int visiblerotationid, int branchstart, int branchend)

  Generate labels of fragment ions of a cyclic peptide (direction right).
- void generateCyclicLabelsToLeft (bool nterminal, int rotationid, int rotationstart, int fragmentstart, int fragmentend, int numberofringblocks, unordered_set< cIonLabel, hash_cIonLabel > &labels, cParameters *parameters, cTheoreticalSpectrum *theoreticalspectrum, int centerx, int centery, int radius, double angle, int linesize, int cornerlinesize, string visibleionseries, string visibleneutralloss, int visiblerotationid, int branchstart, int branchend)

  Generate labels of fragment ions of a cyclic peptide (direction left).

17.67.1 Detailed Description

Visualization of a cyclic peptide.

17.67.2 Function Documentation

17.67.2.1 generateCyclicLabelsToLeft()

```
void generateCyclicLabelsToLeft (
            bool nterminal,
            int rotationid,
            int rotationstart,
            int fragmentstart,
            int fragmentend,
            int numberofringblocks,
            unordered_set< cIonLabel, hash_cIonLabel > & labels,
            cParameters * parameters,
            cTheoreticalSpectrum * theoreticalspectrum,
            int centerx,
            int centery,
            int radius,
            double angle,
            int linesize,
            int cornerlinesize,
            string visibleionseries,
            string visibleneutralloss,
            int visiblerotationid,
            int branchstart,
            int branchend )
```

Generate labels of fragment ions of a cyclic peptide (direction left).

Parameters

nterminal	if true only nterminal fragment ions are drawn, if false only cterminal fragment ions are drawn
rotationid	id of a sequence rotation
rotationstart	starting point on a ring to draw a series

Parameters

fragmentstart	first fragment to draw
fragmentend	last fragment to draw
numberofringblocks	number of building blocks on the ring
labels	labels of fragment ions
parameters	a pointer to parameters of the application
theoreticalspectrum	a theoretical spectrum
centerx	x coordinate of a center of the ring
centery	y coordinate of a center of the ring
radius	radius of the ring
angle	an angle between two building blocks
linesize	length of separators between building blocks
cornerlinesize	length of endings of separators between building blocks
visibleionseries	name of a ion series to be visualized
visibleneutralloss	name of a neutral loss to be visualized
visiblerotationid	id of a rotation whose fragment ions are visualized
branchstart	starting position of a branch
branchend	end position of a branch

17.67.2.2 generateCyclicLabelsToRight()

```
void generateCyclicLabelsToRight (
            bool nterminal,
            int rotationid,
            int rotationstart,
            int fragmentstart,
            int fragmentend,
            int numberofringblocks,
            unordered_set< cIonLabel, hash_cIonLabel > & labels,
            cParameters * parameters,
            cTheoreticalSpectrum * theoreticalspectrum,
            int centerx,
            int centery,
            int radius,
            double angle,
            int linesize,
            int cornerlinesize,
            string visibleionseries,
            string visibleneutralloss,
            int visiblerotationid,
            int branchstart,
            int branchend )
```

Generate labels of fragment ions of a cyclic peptide (direction right).

Parameters

nterminal	if true only nterminal fragment ions are drawn, if false only cterminal fragment ions are drawn
rotationid	id of a sequence rotation
rotationstart	starting point on a ring to draw a series
fragmentstart	first fragment to draw
fragmentend	last fragment to draw

Parameters

numberofringblocks	number of building blocks on the ring
labels	labels of fragment ions
parameters	a pointer to parameters of the application
theoreticalspectrum	a theoretical spectrum
centerx	x coordinate of a center of the ring
centery	y coordinate of a center of the ring
radius	radius of the ring
angle	an angle between two building blocks
linesize	length of separators between building blocks
cornerlinesize	length of endings of separators between building blocks
visibleionseries	name of a ion series to be visualized
visibleneutralloss	name of a neutral loss to be visualized
visiblerotationid	id of a rotation whose fragment ions are visualized
branchstart	starting position of a branch
branchend	end position of a branch

17.67.2.3 paintCircle()

```
void paintCircle (
            QPainter & painter,
            vector< string > & acronymsofblocks,
            int centerx,
            int centery,
            int radius,
            double angle,
            int horizontalstep,
            int linesize,
            int cornerlinesize,
            bool drawdashlines,
            int visiblerotationid,
            unordered_set< cIonLabel, hash_cIonLabel > & labels )
```

Paint the circle of a cyclic or a branch-cyclic peptide.

Parameters

painter	a reference to current QPainter
acronymsofblocks	acronyms of building blocks
centerx	x coordinate of a center of the ring
centery	y coordinate of a center of the ring
radius	radius of the ring
angle	an angle between two building blocks
horizontalstep	determine width of rectangles with the acronyms of building blocks
linesize	length of separators between building blocks
cornerlinesize	length of endings of separators between building blocks
drawdashlines	true when dashed lines between blocks and numbers of blocks are painted, false otherwise
visiblerotationid	an identifier of a rotation of a cyclic peptide
labels	labels of fragment ions

17.68 cCyclicWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CCYCLICWIDGET_H
8 #define _CCYCLICWIDGET_H
9
10 #include <math.h>
11 #include <QWidget>
12 #include <QPalette>
13
14 #include "core/cTheoreticalSpectrum.h"
15 #include "gui/cLinearWidget.h"
16
17
18 // forward declaration
19 class QPaintEvent;
20
21
37 void paintCircle(QPainter& painter, vector<string>& acronymsofblocks, int centerx, int centery, int
      radius, double angle, int horizontalstep, int linesize, int cornerlinesize, bool drawdashlines, int
      visiblerotationid, unordered_set<cIonLabel, hash_cIonLabel>& labels);
38
39
63 void generateCyclicLabelsToRight(bool nterminal, int rotationid, int rotationstart, int fragmentstart,
      int fragmentend, int numberofringblocks, unordered_set<cIonLabel, hash_cIonLabel>& labels,
      cParameters* parameters, cTheoreticalSpectrum* theoreticalspectrum, int centerx, int centery, int
      radius, double angle, int linesize, int cornerlinesize, string visibleionseries, string
      visibleneutralloss, int visiblerotationid, int branchstart, int branchend);
64
65
89 void generateCyclicLabelsToLeft(bool nterminal, int rotationid, int rotationstart, int fragmentstart, int
      fragmentend, int numberofringblocks, unordered_set<cIonLabel, hash_cIonLabel>& labels, cParameters*
      parameters, cTheoreticalSpectrum* theoreticalspectrum, int centerx, int centery, int radius, double
      angle, int linesize, int cornerlinesize, string visibleionseries, string visibleneutralloss, int
      visiblerotationid, int branchstart, int branchend);
90
91
95 class cCyclicWidget :  public QWidget
96 {
97     Q_OBJECT
98
99 public:
100
101
105     cCyclicWidget();
106
107
113     void initialize(cParameters* parameters, cTheoreticalSpectrum* theoreticalspectrum);
114
115
121     void exportToPDF(QString filename, bool postscript);
122
123
128     void exportToPNG(QString filename);
129
130
135     void exportToSVG(QString filename);
136
137
142     void setReportIsomers(bool reportisomers);
143
144
149     void hidePeakLabels(bool state);
150
151
152 protected:
153
154
159     void paintEvent(QPaintEvent *event);
160
161
162 private:
163
164     void paint(QPainter& painter);
165
166     cParameters* parameters;
167     cTheoreticalSpectrum* theoreticalspectrum;
168
169     string visibleionseries;
170     string visibleneutralloss;
171     int visiblerotationid;
172
173     bool reportisomers;
174     bool hidelabels;
175
```

```
176
177 private slots:
178
179     void ionSeriesChanged(QString text);
180
181     void neutralLossChanged(QString text);
182
183     void rotationChanged(int index);
184
185 };
186
187 #endif
```

17.69 gui/cDefineCalibrationWidget.h File Reference

The widget for definition of calibration curve.

```
#include <QWidget>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include "core/utilities.h"
```

Classes

- class cDefineCalibrationWidget

  The widget for definition of calibration curve.

17.69.1 Detailed Description

The widget for definition of calibration curve.

17.70 cDefineCalibrationWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CDEFINECALIBRATIONWIDGET_H
8 #define _CDEFINECALIBRATIONWIDGET_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QUrl>
13 #include <QFileInfo>
14
15 #include "core/utilities.h"
16
17
18 // forward declaration
19 class QScrollArea;
20 class QVBoxLayout;
21 class QHBoxLayout;
22 class QGridLayout;
23 class QLabel;
24 class QListWidget;
25 class QCheckBox;
26 class QLineEdit;
27 class QSpinBox;
28 class QDoubleSpinBox;
29 class QPushButton;
30 class QComboBox;
31 class QFormLayout;
32
33
37 class cDefineCalibrationWidget :  public QWidget
38 {
39     Q_OBJECT
40
41 public:
42
43
47     cDefineCalibrationWidget();
48
49
```

```
53      ~cDefineCalibrationWidget();
54
55
60      void closeEvent(QCloseEvent *event);
61
62
67      void store(ofstream& os);
68
69
77      void load(ifstream& is, int fileversionpart1, int fileversionpart2, int fileversionpart3);
78
79
85      void prepareWidget(vector<string>& filenames, vector<string>& compounds);
86
87
103      void getData(vector<int>& usedvector, vector<int>& datatypesvector, vector<string>& groupsvector,
       vector<double>& concentrationsvector, vector<int>& timesvector, vector<int>& selectedionsvector, int&
       equationtype, double& manuala, double& manualb, int& eictype, int& peakshape, int& standard);
104
105
106 private:
107
108     QScrollArea* scrollarea;
109     QWidget* scrollwidget;
110     QVBoxLayout* scrolllayout;
111
112     QVBoxLayout* vlayout;
113     QWidget* filelistwidget;
114
115     QGridLayout* gridlayout;
116
117     QLabel* uselabel;
118     QLabel* filelabel;
119     QLabel* grouplabel;
120     QLabel* datatypelabel;
121     QLabel* concentrationlabel;
122     QLabel* timelabel;
123
124     vector<QCheckBox*> checkboxes;
125     vector<QLineEdit*> lineedits;
126     vector<QComboBox*> comboboxes;
127     vector<QLineEdit*> groupnames;
128     vector<QDoubleSpinBox*> doublespinboxes;
129     vector<QSpinBox*> timespinboxes;
130
131     QLabel* compoundslabel;
132     QListWidget* compoundslist;
133     QVBoxLayout* compoundslayout;
134     QWidget* compoundswidget;
135
136     QFormLayout* formlayout;
137     QWidget* formwidget;
138
139     QComboBox* equationcombobox;
140     QDoubleSpinBox* doublespinboxa;
141     QDoubleSpinBox* doublespinboxb;
142     QComboBox* eiccombobox;
143     QComboBox* peakshapecombobox;
144     QComboBox* standardcombobox;
145
146     QPushButton* okbutton;
147     QPushButton* cancelbutton;
148     QHBoxLayout* buttonslayout;
149     QWidget* buttonswidget;
150
151     vector<int> internalusedvector;
152     vector<string> internalfilenames;
153     vector<int> internaldatatypesvector;
154     vector<string> internalgroupsvector;
155     vector<double> internalconcentrationsvector;
156     vector<int> internaltimesvector;
157
158     vector<string> internalcompounds;
159     vector<int> internalselectedionsvector;
160
161     int internalequationtype;
162     double internala;
163     double internalb;
164     int internaleictype;
165     int internalpeakshape;
166     int internalstandard;
167
168 protected:
169
170
175     void keyPressEvent(QKeyEvent *event);
176
```

```
177
178 private slots:
179
180     void okButtonReleased();
181
182
183     void cancelButtonReleased();
184
185
186     void dataTypeChanged(int index);
187
188
189     void equationChanged(int index);
190
191
192 signals:
193
194
198     void calibrationRedefined();
199
200 };
201
202 #endif
203
```

17.71 gui/cDrawPeptideWidget.h File Reference

Draw a peptide from a sequence of building blocks.

```
#include <QWidget>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include "gui/cLinearWidget.h"
#include "gui/cCyclicWidget.h"
#include "gui/cBranchedWidget.h"
#include "gui/cBranchCyclicWidget.h"
```

Classes

- class cDrawPeptideWidget

  Draw a peptide from a sequence of building blocks.

17.71.1 Detailed Description

Draw a peptide from a sequence of building blocks.

17.72 cDrawPeptideWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CDRAWPEPTIDEWIDGET_H
8 #define _CDRAWPEPTIDEWIDGET_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QUrl>
13 #include <QFileInfo>
14 #include "gui/cLinearWidget.h"
15 #include "gui/cCyclicWidget.h"
16 #include "gui/cBranchedWidget.h"
17 #include "gui/cBranchCyclicWidget.h"
18
19
20 // forward declaration
21 class QTextEdit;
22 class QTextBrowser;
23 class QHBoxLayout;
24 class QVBoxLayout;
25 class QFormLayout;
26 class QScrollArea;
```

```
27 class QCheckBox;
28 class QPushButton;
29 class QComboBox;
30 class QDoubleSpinBox;
31 class QLineEdit;
32 class QSpinBox;
33 class QLabel;
34 class QAction;
35 class QMenu;
36 class QToolButton;
37 class QToolBar;
38
39
43 class cDrawPeptideWidget :  public QWidget
44 {
45     Q_OBJECT
46
47 public:
48
49
54     cDrawPeptideWidget(QWidget* parent = (QWidget *)0);
55
56
60     ~cDrawPeptideWidget();
61
62
67     void closeEvent(QCloseEvent *event);
68
69
75     void setSequence(int peptidetypeindex, QString sequence);
76
77
78 protected:
79
80
85     void keyPressEvent(QKeyEvent *event);
86
87
88 private:
89
90     QWidget* parent;
91
92     QWidget* drawpeptideformwidget;
93     QFormLayout* drawpeptideformlayout;
94     QComboBox* peptidetypecombobox;
95
96     QLineEdit* sequenceline;
97     QLabel* sequencestatuslabel;
98     QAction *sequenceactioncopy;
99     QAction *sequenceactionsetsequence;
100     QAction *sequenceactionsettag;
101     QAction *sequenceactioninsert;
102     QMenu *sequencemenu;
103     QToolButton* sequencetoolbutton;
104     QToolBar *sequencetoolbar;
105     QHBoxLayout* sequencelayout;
106
107     QCheckBox* separateblocks;
108     QSpinBox* numberofblocksbackbone;
109     QSpinBox* numberofblocksbranch;
110     QSpinBox* branchposition;
111     QHBoxLayout* backboneblocks;
112     QWidget* backboneblockswidget;
113     QHBoxLayout* branchblocks;
114     QWidget* branchblockswidget;
115
116     cTheoreticalSpectrum theoreticalspectrum;
117     cLinearWidget* linearwidget;
118     cCyclicWidget* cyclicwidget;
119     cBranchedWidget* branchedwidget;
120     cBranchCyclicWidget* branchcyclicwidget;
121
122     QVBoxLayout* vbox;
123     QHBoxLayout* mainbox;
124
125
126     void drawPeptide(vector<string>& composition, cBricksDatabase& bricksdb, int branchstart, int
    branchend);
127
128
129 private slots:
130
131     void separateBlocksChanged(int state);
132
133     void numberOfBackboneBlocksChanged(int numberofblocks);
134
135     void numberOfBranchBlocksChanged(int numberofblocks);
```

```
136
137     void peptideTypeChanged(int index);
138
139     void sequenceCopy();
140
141     void sequenceSetSequence();
142
143     void sequenceSetTag();
144
145     void sequenceInsert();
146
147     void blocksChanged();
148
149     void sequenceChanged();
150
151
152 signals:
153
159     void sendSequence(int peptidetypeindex, QString sequence);
160
161
167     void sendTag(int peptidetypeindex, QString tag);
168
169
175     void sendInsert(int peptidetypeindex, QString sequence);
176
177 };
178
179 #endif
```

17.73 gui/cEventFilter.h File Reference

An event filter.

```
#include <QObject>
#include <QEvent>
#include <QKeyEvent>
```

Classes

- class cEventFilter

  An event filter.

17.73.1 Detailed Description

An event filter.

17.74 cEventFilter.h

Go to the documentation of this file.

```
1
7 #ifndef _CEVENTFILTER_H
8 #define _CEVENTFILTER_H
9
10 #include <QObject>
11 #include <QEvent>
12 #include <QKeyEvent>
13
14
18 class cEventFilter :  public QObject {
19
20     Q_OBJECT
21
22  protected:
23
29     bool eventFilter(QObject *obj, QEvent *event);
30
31 };
32
33
34 #endif
35
```

17.75 gui/cExportDialog.h File Reference

The implementation of a dialog to export spectrum/peptide image.

```
#include <QWidget>
#include <QDialog>
#include <QHBoxLayout>
#include <QFormLayout>
#include <QRadioButton>
#include <QPushButton>
```

Classes

- class cExportDialog

  The implementation of a dialog to export spectrum/peptide image.

17.75.1 Detailed Description

The implementation of a dialog to export spectrum/peptide image.

17.76 cExportDialog.h

Go to the documentation of this file.

```
1
7 #ifndef _CEXPORTDIALOG_H
8 #define _CEXPORTDIALOG_H
9
10 #include <QWidget>
11 #include <QDialog>
12 #include <QHBoxLayout>
13 #include <QFormLayout>
14 #include <QRadioButton>
15 #include <QPushButton>
16
17
21 class cExportDialog :  public QDialog
22 {
23     Q_OBJECT
24
25 private:
26
27     QWidget* parent;
28
29     QHBoxLayout* exportbox;
30
31     QFormLayout* formlayout;
32     QPushButton* exportbutton;
33     QRadioButton* exportspectrum;
34     QRadioButton* exportpeptide;
35
36 public:
37
38
43     cExportDialog(QWidget* parent = (QWidget *)0);
44
45
49     ~cExportDialog();
50
51
52 private slots:
53
54
55     void processExport();
56
57 };
58
59 #endif
```

17.77 gui/cFindDialog.h File Reference

The implementation of a find dialog.

```
#include <QWidget>
#include <QDialog>
#include <QHBoxLayout>
#include <QFormLayout>
#include <QLineEdit>
#include <QCheckBox>
#include <QPushButton>
#include <QTextDocument>
```

Classes

- class cFindDialog

  The implementation of a find dialog.

17.77.1 Detailed Description

The implementation of a find dialog.

17.78 cFindDialog.h

Go to the documentation of this file.

```
1
7 #ifndef _CFINDDIALOG_H
8 #define _CFINDDIALOG_H
9
10 #include <QWidget>
11 #include <QDialog>
12 #include <QHBoxLayout>
13 #include <QFormLayout>
14 #include <QLineEdit>
15 #include <QCheckBox>
16 #include <QPushButton>
17 #include <QTextDocument>
18
19
23 class cFindDialog :  public QDialog
24 {
25     Q_OBJECT
26
27 private:
28
29     QWidget* parent;
30
31     QHBoxLayout* searchbox;
32
33     QFormLayout* formlayout;
34     QLineEdit* searchline;
35     QPushButton* searchbutton;
36     QCheckBox* matchcase;
37     QCheckBox* matchwholeword;
38
39 public:
40
41
46     cFindDialog(QWidget* parent = (QWidget *)0);
47
48
52     ~cFindDialog();
53
54
55 private slots:
56
57
58     void findAll();
59
60 };
61
62 #endif
```

17.79 gui/cFormulaPredictor.h File Reference

Predicts the molecular formula from the exact mass.

```
#include <QWidget>
#include <QDesktopServices>
#include <QToolBar>
#include <QMenuBar>
#include <QMenu>
#include <QUrl>
#include <QFileInfo>
#include <QMainWindow>
#include <QTextDocument>
#include <QTableView>
#include <QStandardItemModel>
#include <QStandardItem>
#include <QHeaderView>
#include <QProgressDialog>
#include <QTextStream>
#include "core/cGlobalPreferences.h"
#include "core/cPeaksList.h"
#include "gui/cViewButtonDelegate.h"
#include "gui/cFragmentIonsListWidget.h"
#include "gui/cNeutralLossesListWidget.h"
#include "gui/cSpectrumDetailProxyModel.h"
#include "gui/cEventFilter.h"
#include "gui/cPubChemSearchWidget.h"
#include "parallel/cFormulaPredictorThread.h"
```

Classes

- class cFormulaPredictor

  Predicts the molecular formula from the exact mass.

17.79.1 Detailed Description

Predicts the molecular formula from the exact mass.

17.80 cFormulaPredictor.h

Go to the documentation of this file.

```
1
7 #ifndef _CFORMULAPREDICTOR_H
8 #define _CFORMULAPREDICTOR_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QToolBar>
13 #include <QMenuBar>
14 #include <QMenu>
15 #include <QUrl>
16 #include <QFileInfo>
17 #include <QMainWindow>
18 #include <QTextDocument>
19 #include <QTableView>
20 #include <QStandardItemModel>
21 #include <QStandardItem>
22 #include <QHeaderView>
23 #include <QProgressDialog>
24 #include <QTextStream>
25 #include "core/cGlobalPreferences.h"
26 #include "core/cPeaksList.h"
27 #include "gui/cViewButtonDelegate.h"
28 #include "gui/cFragmentIonsListWidget.h"
29 #include "gui/cNeutralLossesListWidget.h"
30 #include "gui/cSpectrumDetailProxyModel.h"
```

```
31 #include "gui/cEventFilter.h"
32 #include "gui/cPubChemSearchWidget.h"
33 #include "parallel/cFormulaPredictorThread.h"
34
35
36 // forward declaration
37 class QTextBrowser;
38 class QHBoxLayout;
39 class QVBoxLayout;
40 class QFormLayout;
41 class QCheckBox;
42 class QPushButton;
43 class QSplitter;
44 class QComboBox;
45 class QDoubleSpinBox;
46 class QSpinBox;
47 class QAction;
48 class QLabel;
49 class QLineEdit;
50
51
55 class cFormulaPredictor :  public QMainWindow
56 {
57     Q_OBJECT
58
59 public:
60
61
67     cFormulaPredictor(cGlobalPreferences* globalpreferences, QWidget *parent);
68
69
73     ~cFormulaPredictor();
74
75
80     void closeEvent(QCloseEvent *event);
81
82
86     void applyGlobalPreferences();
87
88
89 protected:
90
91
96     void keyPressEvent(QKeyEvent *event);
97
98
99 private:
100
101     cGlobalPreferences* globalpreferences;
102     QWidget* parent;
103
104     cFormulaPredictorThread* formulapredictorthread;
105     int chargevalue;
106     int maximumcombinedelements;
107     bool basicformulacheck;
108     bool advancedformulacheck;
109     bool noratiocheck;
110     double mzvalue;
111     double mzerror;
112     vector<cIonType> selectedions;
113     vector<string> selectedelements;
114     vector<cFormulaPredictorRowItem> ionitems;
115     bool success;
116     bool terminatecomputation;
117     string errmsg;
118
119     map<string, int> pchemdata;
120
121     QMenuBar* menuBar;
122     QMenu* menuFile;
123     QMenu* menuSearch;
124     QMenu* menuHelp;
125
126     QToolBar* toolbarFile;
127     QToolBar* toolbarSearch;
128     QToolBar* toolbarHelp;
129
130     QAction* actionExportTable;
131     QAction* actionCloseWindow;
132     QAction* actionSearchPubChem;
133     QAction* actionHTMLDocumentation;
134
135     QDoubleSpinBox* doublespinboxmz;
136     QSpinBox* spinboxcharge;
137     QDoubleSpinBox* doublespinboxmzerror;
138     cFragmentIonsListWidget* widgetiontypes;
139     cNeutralLossesListWidget* widgetelements;
```

```
140     QSpinBox* spinboxmaximumcombinedelements;
141     QCheckBox* checkboxbasicformula;
142     QCheckBox* checkboxadvancedformula;
143     QCheckBox* checkboxnoratio;
144     QPushButton* predictbutton;
145     QPushButton* cancelbutton;
146 
147     QFormLayout* formlayout;
148     QWidget* formwidget;
149 
150     QTableView* peakstable;
151     QStandardItemModel* peakstablemodel;
152     cSpectrumDetailProxyModel* proxymodel;
153 
154     QSplitter* vsplitter1;
155     QSplitter* vsplitter2;
156     QSplitter* hsplitter;
157 
158     void preparePeaksTable();
159 
160     void deleteTable();
161 
162 
163 private slots:
164 
165 
166     void predictButtonReleased();
167 
168 
169     void cancelButtonReleased();
170 
171 
172     void exportTableToCSV();
173 
174 
175     void showHTMLDocumentation();
176 
177 
178     void closeWindow();
179 
180 
181     void resetIonTypes();
182 
183 
184     void formulaPredictorThreadFinished();
185 
186 
187     void searchPubChem();
188 
189 };
190 
191 #endif
192 
```

17.81 gui/cFragmentIonsListWidget.h File Reference

The implementation of a widget where types of fragment ions generated in theoretical spectra are selected.

```
#include <string>
#include <QListWidget>
#include <QHBoxLayout>
#include <QVBoxLayout>
```

Classes

- class cFragmentIonsListWidget

  The widget for selection of fragment ion types.

17.81.1 Detailed Description

The implementation of a widget where types of fragment ions generated in theoretical spectra are selected.

17.82 cFragmentIonsListWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CFRAGMENTIONSLISTWIDGET_H
8 #define _CFRAGMENTIONSLISTWIDGET_H
9
10 #include <string>
11 #include <QListWidget>
12 #include <QHBoxLayout>
13 #include <QVBoxLayout>
14
15
16 // forward declaration
17 class QPushButton;
18 class QHBoxLayout;
19 class QVBoxLayout;
20
21
25 class cFragmentIonsListWidget :  public QWidget
26 {
27     Q_OBJECT
28
29 public:
30
31
36     cFragmentIonsListWidget(QObject* parent);
37
38
42     ~cFragmentIonsListWidget();
43
44
49     QListWidget* getList();
50
51
52 private:
53     QObject* parent;
54
55     QListWidget* list;
56     QPushButton* selectallbutton;
57     QPushButton* clearallbutton;
58     QPushButton* resetbutton;
59
60     QHBoxLayout* hbox;
61     QVBoxLayout* vbox1;
62     QVBoxLayout* vbox2;
63
64
65 private slots:
66
67     void selectAllItems();
68
69     void clearAllItems();
70
71     void sendResetReleased();
72
73 signals:
74
78     void resetReleased();
79
80 };
81
82 #endif
```

17.83 gui/cGraphWidget.h File Reference

The implementation of the dialog 'View->Graph'.

```
#include <string>
#include <fstream>
#include <QWidget>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include "core/utilities.h"
```

Classes

- class cGraphWidget

  The widget representing the dialog 'View->Graph'.

17.83.1 Detailed Description

The implementation of the dialog 'View->Graph'.

17.84 cGraphWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CGRAPHWIDGET_H
8 #define _CGRAPHWIDGET_H
9
10 #include <string>
11 #include <fstream>
12 #include <QWidget>
13 #include <QDesktopServices>
14 #include <QUrl>
15 #include <QFileInfo>
16 #include "core/utilities.h"
17
18
19 using namespace std;
20
21
22 // forward declaration
23 class QTextBrowser;
24 class QVBoxLayout;
25
26
30 class cGraphWidget :  public QWidget
31 {
32     Q_OBJECT
33
34 public:
35
36
40     cGraphWidget();
41
42
46     ~cGraphWidget();
47
48
54     void setHTML(string html, bool reportisomers);
55
56
61     string getHTML();
62
63
68     void updateView(bool reportisomers);
69
70
75     void closeEvent(QCloseEvent *event);
76
77
82     void store(ofstream& os);
83
84
89     void load(ifstream& is);
90
91
92 private:
93
94     QVBoxLayout* layout;
95     QTextBrowser* textbrowser;
96     string htmlstring;
97
98
99 protected:
100
101
106     void keyPressEvent(QKeyEvent *event);
107
108
109 };
110
111 #endif
```

17.85 gui/cHTMLDelegate.h File Reference

A delegate to display rich text.

```
#include <QApplication>
#include <QString>
#include <QVariant>
#include <QStyledItemDelegate>
#include <QPainter>
#include <QModelIndex>
#include <QEvent>
#include <QAbstractItemModel>
#include <QCheckBox>
#include <QMouseEvent>
#include <QTextDocument>
```

Classes

- class cHTMLDelegate

  A delegate to display rich text.

17.85.1 Detailed Description

A delegate to display rich text.

17.86 cHTMLDelegate.h

Go to the documentation of this file.

```
1
7 #ifndef _CHTMLDELEGATE_H
8 #define _CHTMLDELEGATE_H
9
10 #include <QApplication>
11 #include <QString>
12 #include <QVariant>
13 #include <QStyledItemDelegate>
14 #include <QPainter>
15 #include <QModelIndex>
16 #include <QEvent>
17 #include <QAbstractItemModel>
18 #include <QCheckBox>
19 #include <QMouseEvent>
20 #include <QTextDocument>
21
22
26 class cHTMLDelegate :  public QStyledItemDelegate {
27
28     Q_OBJECT
29
30 public:
31
32
37     cHTMLDelegate(QObject* parent = (QObject *)0);
38
39
46     void paint(QPainter* painter, const QStyleOptionViewItem& option, const QModelIndex& index) const;
47
48
49     //QSize sizeHint (const QStyleOptionViewItem& option, const QModelIndex& index) const;
50
51
52 };
53
54
55 #endif
56
```

17.87 gui/cHTMLExportDialog.h File Reference

The implementation of HTML export dialog.

```
#include <QWidget>
#include <QDialog>
#include <QHBoxLayout>
#include <QFormLayout>
#include <QDialogButtonBox>
#include <QGroupBox>
#include <QCheckBox>
```

Classes

- class cHTMLExportDialog

  The implementation of HTML export dialog.

17.87.1 Detailed Description

The implementation of HTML export dialog.

17.88 cHTMLExportDialog.h

Go to the documentation of this file.

```
1
7 #ifndef _CHTMLEXPORTDIALOG_H
8 #define _CHTMLEXPORTDIALOG_H
9
10 #include <QWidget>
11 #include <QDialog>
12 #include <QHBoxLayout>
13 #include <QFormLayout>
14 #include <QDialogButtonBox>
15 #include <QGroupBox>
16 #include <QCheckBox>
17
18
22 class cHTMLExportDialog :  public QDialog
23 {
24     Q_OBJECT
25
26 private:
27
28     QWidget* parent;
29
30     QFormLayout* form1;
31     QFormLayout* form2;
32
33     QGroupBox* group1;
34     QGroupBox* group2;
35
36     QHBoxLayout* hbox;
37     QDialogButtonBox* buttonbox;
38     QVBoxLayout* vbox;
39
40 public:
41
45     QCheckBox* checkboxparameters;
46
47
51     QCheckBox* checkboxdenovo;
52
53
57     QCheckBox* checkboxlogwindow;
58
59
63     QCheckBox* checkboxsummaryresultstable;
64
65
69     QCheckBox* checkboxsummarypeakstable;
70
71
75     QCheckBox* checkboxpeakstable;
76
77
81     QCheckBox* checkboxunmatchedexperimental;
82
83
87     QCheckBox* checkboxunmatchedtheoretical;
```

```
88
89
93     QCheckBox* checkboxdetails;
94
95
100     cHTMLExportDialog(QWidget* parent = (QWidget *)0);
101
102
106     ~cHTMLExportDialog();
107
108 };
109
110 #endif
111
```

17.89 gui/cImageWindow.h File Reference

Image window.

```
#include <QMainWindow>
#include <QDesktopServices>
#include <QToolBar>
#include <QAction>
#include <QGridLayout>
#include <QCheckBox>
#include <QSlider>
#include <QSplitter>
#include <QRadioButton>
#include <QSpinBox>
#include <QDoubleSpinBox>
#include <QInputDialog>
#include <QMessageBox>
#include <QScrollArea>
#include <QProgressDialog>
#include "core/utilities.h"
#include "core/cGlobalPreferences.h"
#include "gui/cImageWindowWidget.h"
#include "gui/cImageWindowImportDialog.h"
#include "gui/cImageWindowTabOptical.h"
```

Classes

- struct microscopyLayerInfo

  Paramaters of a microscopy layer.
- class cImageWindow

  Image window.

17.89.1 Detailed Description

Image window.

17.90 cImageWindow.h

Go to the documentation of this file.

```
1
7 #ifndef _CIMAGEWINDOW_H
8 #define _CIMAGEWINDOW_H
9
10 #include <QMainWindow>
11 #include <QDesktopServices>
12 #include <QToolBar>
13 #include <QAction>
14 #include <QGridLayout>
15 #include <QCheckBox>
```

```
16 #include <QSlider>
17 #include <QSplitter>
18 #include <QRadioButton>
19 #include <QSpinBox>
20 #include <QDoubleSpinBox>
21 #include <QInputDialog>
22 #include <QMessageBox>
23 #include <QScrollArea>
24 #include <QProgressDialog>
25 #include "core/utilities.h"
26 #include "core/cGlobalPreferences.h"
27 #include "gui/cImageWindowWidget.h"
28 #include "gui/cImageWindowImportDialog.h"
29 #include "gui/cImageWindowTabOptical.h"
30
31
32 // forward declaration
33 class QHBoxLayout;
34 class QVBoxLayout;
35 class QPushButton;
36 class QImage;
37 class QLabel;
38 class QAction;
39 class QToolBar;
40 class QMenuBar;
41 class QMenu;
42
43
47 struct microscopyLayerInfo {
51     int type;
52
53
57     int navigationlayer;
58
59
63     QString sourcename;
64
65
69     QString imagename;
70
71
75     bool flipx;
76
77
81     bool flipy;
82
83
87     double x;
88
89
93     double y;
94
95
99     double width;
100
101
105     double height;
106
107
111     double angle;
112
113
117     double defaultwidth;
118
119
123     double defaultheight;
124
125
129     double stagex;
130
131
135     double stagey;
136
137
141     QImage* image;
142
143
147     bool reduced;
148
149
153     int thresholdmin;
154
155
159     int thresholdmax;
160
161
165     microscopyLayerInfo() {
```

```
166         image = new QImage();
167         clear();
168     }
169 
170 
174     microscopyLayerInfo(const microscopyLayerInfo& layer) {
175         type = layer.type;
176         navigationlayer = layer.navigationlayer;
177         sourcename = layer.sourcename;
178         imagename = layer.imagename;
179         flipx = layer.flipx;
180         flipy = layer.flipy;
181         x = layer.x;
182         y = layer.y;
183         width = layer.width;
184         height = layer.height;
185         angle = layer.angle;
186         defaultwidth = layer.defaultwidth;
187         defaultheight = layer.defaultheight;
188         stagex = layer.stagex;
189         stagey = layer.stagey;
190         image = new QImage();
191         *image = *layer.image;
192         reduced = layer.reduced;
193         thresholdmin = layer.thresholdmin;
194         thresholdmax = layer.thresholdmax;
195     }
196 
197 
201     ~microscopyLayerInfo() {
202         delete image;
203     }
204 
205 
209     void clear() {
210         type = -1;
211         navigationlayer = 3;
212         sourcename.clear();
213         imagename.clear();
214         flipx = false;
215         flipy = false;
216         x = 0;
217         y = 0;
218         width = 0;
219         height = 0;
220         angle = 0;
221         defaultwidth = 0;
222         defaultheight = 0;
223         stagex = 0;
224         stagey = 0;
225 
226         delete image;
227         image = new QImage();
228 
229         reduced = false;
230         thresholdmin = 0;
231         thresholdmax = 255;
232     }
233 };
234 
235 
239 class cImageWindow :  public QMainWindow
240 {
241     Q_OBJECT
242 
243 public:
244 
245 
251     cImageWindow(cGlobalPreferences* globalpreferences, QWidget* parent = (QWidget *)0);
252 
253 
257     ~cImageWindow();
258 
259 
264     void closeEvent(QCloseEvent *event);
265 
266 
276     void setDefaultValues(string imzmlfilename, int defaultmaxx, int defaultmaxy, int defaultpixelsizex,
      int defaultpixelsizey, eVendorType vendor);
277 
278 
283     void addLayer(QString name);
284 
285 
290     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
291 
292 
```

```
293 private:
294
295     QString editorname;
296
297     cImageWindowImportDialog importdialog;
298
299     cGlobalPreferences* globalpreferences;
300     QWidget* parent;
301
302     QMenuBar* menuBar;
303     QMenu* menuFile;
304     QMenu* menuView;
305     QMenu* menuHelp;
306
307     QToolBar* toolbarFile;
308     QAction* actionLoadLayers;
309     QAction* actionSaveLayers;
310     QAction* actionSaveLayersAs;
311     QAction* actionOpenImage;
312     QAction* actionImportDialog;
313     QAction* actionExportImage;
314     QAction* actionClearAll;
315     QAction* actionCloseWindow;
316
317     QToolBar* toolbarView;
318     QAction* actionShowSelection;
319     QAction* actionColorScale;
320     QAction* actionAbsoluteIntensity;
321     QAction* actionZoomIn;
322     QAction* actionZoomOut;
323     QAction* actionZoomReset;
324     QAction* actionFlipHistologyHorizontal;
325     QAction* actionFlipHistologyVertical;
326     QAction* actionFlipMicroscopyHorizontal;
327     QAction* actionFlipMicroscopyVertical;
328
329     QLabel* rulerLabel;
330     QDoubleSpinBox* rulerValue;
331
332     QToolBar* toolbarRegion;
333     QWidget* regionwidget;
334     QHBoxLayout* regionhbox;
335     QLabel* xfromlabel;
336     QSpinBox* xfrom;
337     QLabel* xtolabel;
338     QSpinBox* xto;
339     QLabel* yfromlabel;
340     QSpinBox* yfrom;
341     QLabel* ytolabel;
342     QSpinBox* yto;
343     QPushButton* setregionbutton;
344     QPushButton* resetregionbutton;
345
346     QToolBar* toolbarMaxCoordinates;
347     QWidget* maxcoordinateswidget;
348     QHBoxLayout* maxcoordinateshbox;
349     QLabel* leftshiftlabel;
350     QSpinBox* leftshift;
351     QLabel* topshiftlabel;
352     QSpinBox* topshift;
353     QLabel* maxxlabel;
354     QSpinBox* maxx;
355     QLabel* maxylabel;
356     QSpinBox* maxy;
357     QLabel* pixelsizexlabel;
358     QSpinBox* pixelsizexspinbox;
359     QLabel* pixelsizeylabel;
360     QSpinBox* pixelsizeyspinbox;
361     QPushButton* setmaxbutton;
362     QPushButton* setdefaultbutton;
363
364     QToolBar* toolbarHistology;
365     QWidget* histologywidget;
366     QHBoxLayout* histologyhbox;
367     QLabel* histologyxlabel;
368     QSpinBox* histologyx;
369     QLabel* histologyylabel;
370     QSpinBox* histologyy;
371     QLabel* histologywidthlabel;
372     QSpinBox* histologywidth;
373     QLabel* histologyheightlabel;
374     QSpinBox* histologyheight;
375     QLabel* histologyanglelabel;
376     QDoubleSpinBox* histologyangle;
377
378     QToolBar* toolbarMicroscopy;
379     QWidget* microscopywidget;
```

```
380     QHBoxLayout* microscopyhbox;
381     QLabel* microscopyxlabel;
382     QDoubleSpinBox* microscopyx;
383     QLabel* microscopyylabel;
384     QDoubleSpinBox* microscopyy;
385     QLabel* microscopywidthlabel;
386     QDoubleSpinBox* microscopywidth;
387     QLabel* microscopyheightlabel;
388     QDoubleSpinBox* microscopyheight;
389     QLabel* microscopyanglelabel;
390     QDoubleSpinBox* microscopyangle;
391     QLabel* microscopynavigationlabel;
392     QComboBox* microscopynavigationcombobox;
393     QPushButton* microscopydefaultbutton;
394     QPushButton* microscopygobutton;
395     QPushButton* microscopyprevbutton;
396     QPushButton* microscopynextbutton;
397
398     QToolBar* toolbarHelp;
399     QAction* actionHTMLDocumentation;
400
401     QImage* image;
402     QImage* histologyimage;
403
404     // layer == 0 - compounds
405     // layer == 1 - optical image
406     // layer == 2 - histology image
407     // layer == 3 - microscopy navigation image
408     // layer >= 4 - microscopy images
409     vector<microscopyLayerInfo> microscopylayers;
410
411     cImageWindowWidget* imagewindowwidget;
412     QGridLayout* layerslayout;
413     QWidget* layerswidget;
414     QScrollArea* layersscrollarea;
415     QSplitter* mainwidget;
416
417     int layerscount;
418     int activelayer;
419
420     QString lastdiropticalimage;
421     QString lastdirhistologyimage;
422     QString lastdirmicroscopyimage;
423     QString lastdirexportimage;
424     QString lastdirlayersfile;
425
426     bool layersfileloaded;
427     bool saveascalled;
428
429     string imzmlfilename;
430
431     int defaultmaxx;
432     int defaultmaxy;
433
434     int defaultpixelsizex;
435     int defaultpixelsizey;
436
437     eVendorType vendor;
438
439     bool redrawenabled;
440
441     void colorSpinBoxes(int layerid);
442
443     void openOpticalImage();
444
445     void reopenOpticalImage(QString filename);
446
447     void openHistologyImage();
448
449     void reopenHistologyImage(QString filename);
450
451     bool parseMicroscopyImage(QString& filename, double& pixelwidth, double& pixelheight, int&
    resolutionx, int& resolutiony, double& realwidth, double& stagex, double& stagey);
452
453     void openMicroscopyImage(int layer, const QString &layername);
454
455     void reopenMicroscopyImage(int layer, QString filename, bool reopen);
456
457     void clearLayers();
458
459     void updateMicroscopyNavigationCombobox(int numberofitems);
460
461     bool checkLoadedImage(QImage* image);
462
463     void readLifImageOptical(string filename, cLifMetadata& metadata, QImage& image, bool
    enableprogress);
464
```

```
465     void readLifImageFluorescence(string filename, cLifMetadata& metadata, QImage& image, int
      minthreshold, int maxthreshold, bool enableprogress);
466
467     void openLifImage(string liffilename, int layer, cLifMetadata& metadata, bool enableprogress);
468
469     void setNavigationLayer(int layer, int navigation);
470
471
472 protected:
473
474
479     void keyPressEvent(QKeyEvent *event);
480
481
486     void keyReleaseEvent(QKeyEvent *event);
487
488
489 private slots:
490
491
492     void closeWindow();
493
494
495     void selectImageType();
496
497
498     void imageTypeSelected(const QString &s);
499
500
501     void openImportDialog();
502
503
504     void exportImage();
505
506
507     void setFilterOptionsSlot(vector<cCoordinateInfo> coordinates, bool operatortype, string
      columnname1, string comparatorname1, string filterstring1, string columnname2, string comparatorname2,
      string filterstring2, bool casesensitive, bool wholeword);
508
509
510     void setMaxButtonReleased();
511
512
513     void setDefaultButtonReleased();
514
515
516     void setRegionButtonReleased();
517
518
519     void histologyChanged();
520
521
522     void histologyPositionChanged(int value);
523
524
525     void histologyAngleChanged(double value);
526
527
528     void microscopyChanged();
529
530
531     void microscopyDefaultButtonReleased();
532
533
534     void microscopyGoButtonReleased();
535
536
537     void microscopyPrevButtonReleased();
538
539
540     void microscopyNextButtonReleased();
541
542
543     void microscopyPositionChanged(double value);
544
545
546     void microscopyAngleChanged(double value);
547
548
549     void microscopyFlipStateChanged(bool state);
550
551
552     void showHTMLDocumentation();
553
554
555     void updateSelection(int xmin, int xmax, int ymin, int ymax);
556
```

```
557
558     void clearSelection();
559
560
561     void resetSelection();
562
563
564     void updatePointsArea(int xmin, int xmax, int ymin, int ymax);
565
566
567     void layerSpinBoxChangedSlot(int alpha);
568
569
570     void layerZValueChangedSlot(int zvalue);
571
572
573     void layerCheckBoxChangedSlot(int state);
574
575
576     void activeLayerChangedSlot(bool checked);
577
578
579     void updateHistologySlot(int x, int y, int width, int height, double angle);
580
581
582     void updateMicroscopySlot(bool flipx, bool flipy, double x, double y, double width, double height,
      double angle);
583
584
585     void updateRulerSlot(double value);
586
587
588     void imageDoubleClickedSlot(int spectrumid);
589
590
591     void loadLayers();
592
593
594     void saveLayers();
595
596
597     void saveLayersAs();
598
599
600     void clearAllLayersSlot();
601
602
603     void microscopyNavigationChanged(int index);
604
605
606     void clearLayer();
607
608
609 signals:
610
611
619     void updateSummaryPeaksTableFilter(int xmin, int xmax, int ymin, int ymax);
620
621
630     void layerChanged(int layerid, bool checked, int alpha, int zvalue, bool redraw);
631
632
637     void activeLayerChanged(int layerid);
638
639
644     void doubleClickedSpectrumIDSignal(int spectrumid);
645
646
647 };
648
649 #endif
```

17.91 gui/cImageWindowImportDialog.h File Reference

The implementation of import dialog in image window.

```
#include <QDialog>
#include <QDesktopServices>
#include <QUrl>
#include <QKeyEvent>
#include <QDialogButtonBox>
#include <QTabWidget>
```

```
#include <QVBoxLayout>
#include <QMessageBox>
```

Classes

- class cImageWindowImportDialog

  Import dialog in image window.

17.91.1 Detailed Description

The implementation of import dialog in image window.

17.92 cImageWindowImportDialog.h

Go to the documentation of this file.

```
1
7 #ifndef _CIMAGEWINDOWIMPORTDIALOG_H
8 #define _CIMAGEWINDOWIMPORTDIALOG_H
9
10 #include <QDialog>
11 #include <QDesktopServices>
12 #include <QUrl>
13 #include <QKeyEvent>
14 #include <QDialogButtonBox>
15 #include <QTabWidget>
16 #include <QVBoxLayout>
17 #include <QMessageBox>
18
19
20 class cImageWindowTabElectron;
21 class cImageWindowTabOptical;
22
23
27 class cImageWindowImportDialog :  public QDialog
28 {
29     Q_OBJECT
30
31 public:
32
33
38     explicit cImageWindowImportDialog(QWidget *parent = 0);
39
40
46     void setFormValues(QString& lastdirmicroscopyimage, int numberoflayers);
47
48
57     int getFormValues(int& startfromindex, QStringList& tiffilenames, int& navigationindex, QString&
       liffilename);
58
59
63     ~cImageWindowImportDialog();
64
65
66 protected:
67
68
73     //void keyPressEvent(QKeyEvent *event);
74
75
76 private:
77
78     cImageWindowTabElectron* tabelectron;
79     cImageWindowTabOptical* taboptical;
80
81     QTabWidget *tabwidget;
82     QDialogButtonBox *buttonbox;
83
84     QVBoxLayout *mainlayout;
85
86
87 public slots:
88
92     void accept();
93
94 };
95
96
```

```
97 #endif
98
```

17.93 gui/cImageWindowTabElectron.h File Reference

The implementation of a tab used to import images from electron microscopy.

```
#include <QWidget>
#include <QGridLayout>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>
#include <QListWidget>
#include <QComboBox>
#include <QFileDialog>
```

Classes

- class cImageWindowTabElectron

  The class representing a tab used to import images from electron microscopy.

17.93.1 Detailed Description

The implementation of a tab used to import images from electron microscopy.

17.94 cImageWindowTabElectron.h

Go to the documentation of this file.

```
1
7 #ifndef _CIMAGEWINDOWTABELECTRON_H
8 #define _CIMAGEWINDOWTABELECTRON_H
9
10 #include <QWidget>
11 #include <QGridLayout>
12 #include <QHBoxLayout>
13 #include <QVBoxLayout>
14 #include <QLabel>
15 #include <QLineEdit>
16 #include <QPushButton>
17 #include <QListWidget>
18 #include <QComboBox>
19 #include <QFileDialog>
20
21
25 class cImageWindowTabElectron :  public QWidget
26 {
27     Q_OBJECT
28
29 public:
30
31
36     explicit cImageWindowTabElectron(QWidget *parent = 0);
37
38
44     void setFormValues(QString& lastdirmicroscopyimage, int numberoflayers);
45
46
52     void getFormValues(int& index, QStringList& filenames);
53
54
58     ~cImageWindowTabElectron();
59
60
61 private:
62
63     QString* lastdirmicroscopyimage;
64
65     QGridLayout* gridlayout;
```

```
66
67      QLabel* labelstartfrom;
68      QComboBox* comboboxstartfrom;
69
70      QLabel* labelfilenames;
71      QListWidget* listwidgetfilenames;
72      QPushButton* buttonfilenames;
73      QStringList imagenames;
74
75
76  private slots:
77
78      void browseButtonReleased();
79
80  };
81
82
83  #endif
84
```

17.95 gui/cImageWindowTabOptical.h File Reference

The implementation of a tab used to import images from light microscopy.

```
#include <QWidget>
#include <QGridLayout>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>
#include <QListWidget>
#include <QComboBox>
#include <QFileDialog>
#include <xercesc/framework/MemBufInputSource.hpp>
#include <xercesc/parsers/XercesDOMParser.hpp>
#include "core/utilities.h"
```

Classes

- struct cTileInfo

  Description of an image tile.
- struct cLifMetadata

  A structure to keep parsed lif metadata.
- class cImageWindowTabOptical

  The class representing a tab used to import images from light microscopy.

Enumerations

- enum eChannelType { **gray** = 0 , **red** = 1 , **green** = 2 , **blue** = 3 }

  Color channels.

Functions

- void readLifMetadata (string filename, vector< cLifMetadata > &parsedmetadata)

  Get metadata from a lif file.
- void parseLifMetadataSubTree (DOMElement *subtree, vector< cLifMetadata > &parsedmetadata)

  Parse metadata from a subtree of a lif file.
- void parseLifMetadata (string &metadata, vector< cLifMetadata > &parsedmetadata)

  Parse metadata from a lif file.

17.95.1 Detailed Description

The implementation of a tab used to import images from light microscopy.

17.95.2 Function Documentation

17.95.2.1 parseLifMetadata()

```
void parseLifMetadata (
            string & metadata,
            vector< cLifMetadata > & parsedmetadata )
```

Parse metadata from a lif file.

Parameters

metadata	input metadata
parsedmetadata	output vector of parsed metadata

17.95.2.2 parseLifMetadataSubTree()

```
void parseLifMetadataSubTree (
            DOMElement * subtree,
            vector< cLifMetadata > & parsedmetadata )
```

Parse metadata from a subtree of a lif file.

Parameters

subtree	root of a subtree
parsedmetadata	output vector of parsed metadata

17.95.2.3 readLifMetadata()

```
void readLifMetadata (
            string filename,
            vector< cLifMetadata > & parsedmetadata )
```

Get metadata from a lif file.

Parameters

filename	input lif file
parsedmetadata	output vector of parsed metadata

17.96 cImageWindowTabOptical.h

Go to the documentation of this file.

```
1
7 #ifndef _CIMAGEWINDOWTABOPTICAL_H
8 #define _CIMAGEWINDOWTABOPTICAL_H
9
10 #include <QWidget>
11 #include <QGridLayout>
12 #include <QHBoxLayout>
13 #include <QVBoxLayout>
```

```
14 #include <QLabel>
15 #include <QLineEdit>
16 #include <QPushButton>
17 #include <QListWidget>
18 #include <QComboBox>
19 #include <QFileDialog>
20 #include <xercesc/framework/MemBufInputSource.hpp>
21 #include <xercesc/parsers/XercesDOMParser.hpp>
22
23 #include "core/utilities.h"
24
25
26 using namespace xercesc;
27
28
32 enum eChannelType {
33     gray = 0,
34     red = 1,
35     green = 2,
36     blue = 3
37 };
38
39
43 struct cTileInfo {
44
48     int fieldx;
49
50
54     int fieldy;
55
56
60     double posx;
61
62
66     double posy;
67
68
72     cTileInfo() {
73         fieldx = 0;
74         fieldy = 0;
75         posx = 0;
76         posy = 0;
77     }
78
79
83     void print() {
84         cout « "fieldx:  " « fieldx « endl;
85         cout « "fieldy:  " « fieldy « endl;
86         cout « "posx:  " « posx « endl;
87         cout « "posy:  " « posy « endl;
88     }
89
90 };
91
92
96 struct cLifMetadata {
97
101     string imagename;
102
103
107     int microscopytype;
108
109
113     string microscopemodel;
114
115
119     int inversemicroscopemodel;
120
121
125     int magnification;
126
127
131     int flipx;
132
133
137     int flipy;
138
139
143     int dimx;
144
145
149     int dimy;
150
151
155     int dimz;
156
157
```

```
161     int dim10;
162
163
167     double width;
168
169
173     double height;
174
175
179     double stageposx;
180
181
185     double stageposy;
186
187
191     long long unsigned memoryblocksize;
192
193
197     string memoryblockid;
198
199
203     vector<long long unsigned> memoryblockoffsets;
204
205
209     vector<eChannelType> channels;
210
211
215     vector<string> luts;
216
217
221     int tilescaninfoflipx;
222
223
227     int tilescaninfoflipy;
228
229
233     vector<cTileInfo> tiles;
234
235
239     cLifMetadata() {
240         imagename.clear();
241         microscopytype = -1;
242         microscopemodel.clear();
243         inversemicroscopemodel = 0;
244         magnification = 0;
245         flipx = 0;
246         flipy = 0;
247         dimx = 0;
248         dimy = 0;
249         dimz = 0;
250         dim10 = 0;
251         width = 0;
252         height = 0;
253         stageposx = 0;
254         stageposy = 0;
255         memoryblocksize = 0;
256         memoryblockid = "";
257         memoryblockoffsets.clear();
258         channels.clear();
259         luts.clear();
260         tilescaninfoflipx = -1;
261         tilescaninfoflipy = -1;
262         tiles.clear();
263     }
264
265
269     void print() {
270         cout « "imagename:  " « imagename « endl;
271         cout « "microscopytype:  " « microscopytype « endl;
272         cout « "microscopemodel:  " « microscopemodel « endl;
273         cout « "inversemicroscopemodel:  " « inversemicroscopemodel « endl;
274         cout « "magnification:  " « magnification « endl;
275         cout « "flipx:  " « flipx « endl;
276         cout « "flipy:  " « flipy « endl;
277         cout « "dimx:  " « dimx « endl;
278         cout « "dimy:  " « dimy « endl;
279         cout « "dimz:  " « dimz « endl;
280         cout « "dim10:  " « dim10 « endl;
281         cout « "width:  " « width « endl;
282         cout « "height:  " « height « endl;
283         cout « "stageposx:  " « stageposx « endl;
284         cout « "stageposy:  " « stageposy « endl;
285         cout « "memoryblocksize:  " « memoryblocksize « endl;
286         cout « "memoryblockid:  " « memoryblockid « endl;
287
288         for (int i = 0; i < (int)memoryblockoffsets.size(); i++) {
289             cout « "memoryblockoffset " « i + 1 « ":  " « memoryblockoffsets[i] « endl;
```

```
290         }
291
292         for (int i = 0; i < (int)channels.size(); i++) {
293             cout « "channel " « i + 1 « ":  " « channels[i] « endl;
294         }
295
296         for (int i = 0; i < (int)luts.size(); i++) {
297             cout « "lut " « i + 1 « ":  " « luts[i] « endl;
298         }
299
300         cout « "tilescaninfoflipx:  " « tilescaninfoflipx « endl;
301         cout « "tilescaninfoflipy:  " « tilescaninfoflipy « endl;
302
303         double avgposx = 0;
304         double avgposy = 0;
305
306         for (auto& it :  tiles) {
307             it.print();
308
309             avgposx += it.posx;
310             avgposy += it.posy;
311         }
312
313         if (tiles.size() > 0) {
314             avgposx /= (double)tiles.size();
315             avgposy /= (double)tiles.size();
316
317             cout « "avgposx:  " « avgposx « endl;
318             cout « "avgpoxy:  " « avgposy « endl;
319         }
320
321         cout « endl;
322     }
323
324 };
325
326
332 void readLifMetadata(string filename, vector<cLifMetadata>& parsedmetadata);
333
334
340 void parseLifMetadataSubTree(DOMElement* subtree, vector<cLifMetadata>& parsedmetadata);
341
342
348 void parseLifMetadata(string& metadata, vector<cLifMetadata>& parsedmetadata);
349
350
354 class cImageWindowTabOptical :  public QWidget
355 {
356     Q_OBJECT
357
358 public:
359
360
365     explicit cImageWindowTabOptical(QWidget *parent = 0);
366
367
373     void setFormValues(QString& lastdirmicroscopyimage, int numberoflayers);
374
375
383     void getFormValues(int& startfromindex, QStringList& tiffilenames, int& navigationindex, QString&
      liffilename);
384
385
389     ~cImageWindowTabOptical();
390
391
392 private:
393
394     QString* lastdirmicroscopyimage;
395
396     QGridLayout* gridlayout;
397
398     QLabel* labelstartfrom;
399     QComboBox* comboboxstartfrom;
400
401     QLabel* labelusevaluesfrom;
402     QComboBox* comboboxusevaluesfrom;
403
404     QLabel* labellif;
405     QLineEdit* lineeditlif;
406     QPushButton* buttonlif;
407
408     QLabel* labelfilenames;
409     QListWidget* listwidgetfilenames;
410     QPushButton* buttonfilenames;
411     QStringList imagenames;
412
```

```
413
414 private slots:
415
416     void browseButtonReleased();
417
418     void lifButtonReleased();
419
420
421 };
422
423
424 #endif
425
```

17.97 gui/cImageWindowWidget.h File Reference

Image window widget.

```
#include <QWidget>
#include <QGraphicsView>
#include <QPixmap>
#include <QImage>
#include <unordered_set>
#include "core/utilities.h"
```

Classes

- struct layerInfo

  Paramaters of a layer.
- class cImageWindowWidget

  Image window widget.

Enumerations

- enum eCursorActivityType {
  cursoractivity_none = 0 , **cursoractivity_move** , **cursoractivity_resize_left** , **cursoractivity_resize_right** ,
  cursoractivity_resize_top , **cursoractivity_resize_bottom** , **cursoractivity_resize_top_left** , **cursoractivity_resize_top_right** ,
  cursoractivity_resize_bottom_left , **cursoractivity_resize_bottom_right** }

  Cursor activity.

Variables

- const int **numberoflayers** = 204

  The number of layers.

17.97.1 Detailed Description

Image window widget.

17.98 cImageWindowWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CIMAGEWINDOWWIDGET_H
8 #define _CIMAGEWINDOWWIDGET_H
9
10 #include <QWidget>
11 #include <QGraphicsView>
12 #include <QPixmap>
13 #include <QImage>
14 #include <unordered_set>
```

```
15 #include "core/utilities.h"
16
17
21 const int numberoflayers = 204;
22
23
27 enum eCursorActivityType {
28     cursoractivity_none = 0,
29     cursoractivity_move,
30     cursoractivity_resize_left,
31     cursoractivity_resize_right,
32     cursoractivity_resize_top,
33     cursoractivity_resize_bottom,
34     cursoractivity_resize_top_left,
35     cursoractivity_resize_top_right,
36     cursoractivity_resize_bottom_left,
37     cursoractivity_resize_bottom_right
38 };
39
40
44 struct layerInfo {
48     bool checked;
49
50
54     int alpha;
55
56
60     int zvalue;
61
62
66     bool ispixmapdefined;
67
68
72     bool fliphorizontally;
73
74
78     bool flipvertically;
79
80
84     double x;
85
86
90     double y;
91
92
96     double width;
97
98
102     double height;
103
104
108     double angle;
109
110
114     qreal lastx;
115
116
120     qreal lasty;
121
122
126     int navigationlayer;
127
128
132     QPixmap* pixmap;
133
134
138     layerInfo() {
139         pixmap = new QPixmap();
140         clear();
141     }
142
143
147     layerInfo(const layerInfo& layer) {
148         checked = layer.checked;
149         alpha = layer.alpha;
150         zvalue = layer.zvalue;
151         ispixmapdefined = layer.ispixmapdefined;
152         fliphorizontally = layer.fliphorizontally;
153         flipvertically = layer.flipvertically;
154         x = layer.x;
155         y = layer.y;
156         width = layer.width;
157         height = layer.height;
158         angle = layer.angle;
159         lastx = layer.lastx;
160         lasty = layer.lasty;
161         navigationlayer = layer.navigationlayer;
```

```
162         pixmap = new QPixmap();
163         *pixmap = *layer.pixmap;
164     }
165 
166 
170     void clear() {
171         checked = true;
172         alpha = 100;
173         zvalue = 0;
174         ispixmapdefined = false;
175         fliphorizontally = false;
176         flipvertically = false;
177         x = 0;
178         y = 0;
179         width = 0;
180         height = 0;
181         angle = 0;
182         lastx = 0;
183         lasty = 0;
184         navigationlayer = 3;
185 
186         delete pixmap;
187         pixmap = new QPixmap();
188     }
189 
190 
194     ~layerInfo() {
195         delete pixmap;
196     }
197 };
198 
199 
200 
204 class cImageWindowWidget :  public QGraphicsView
205 {
206     Q_OBJECT
207 
208 public:
209 
210 
214     cImageWindowWidget();
215 
216 
220     ~cImageWindowWidget();
221 
222 
227     void setOpticalImage(QImage* image);
228 
229 
234     void setHistologyImage(QImage* histologyimage);
235 
236 
242     void setMicroscopyImage(int layer, QImage* microscopyimage);
243 
244 
249     QImage getImage();
250 
251 
265     void setFilterOptions(vector<cCoordinateInfo>& coordinateinfo, bool operatortype, string&
    columnname1, string& comparatorname1, string& filterstring1, string& columnname2, string&
    comparatorname2, string& filterstring2, bool casesensitive, bool wholeword);
266 
267 
277     void setCorrelationValues(int leftshift, int topshift, int maxx, int maxy, int pixelsizex, int
    pixelsizey);
278 
279 
288     void setDefaultMaxXY(int defaultmaxx, int defaultmaxy, int defaultpixelsizex, int defaultpixelsizey,
    eVendorType vendor);
289 
290 
299     void setHistologyPosition(int x, int y, int width, int height, double angle);
300 
301 
313     void setMicroscopyPosition(int layer, bool flipx, bool flipy, double x, double y, double width,
    double height, double angle);
314 
315 
320     void goToMicroscopyPosition(int layer);
321 
322 
330     void setSelectedRegion(int xfrom, int xto, int yfrom, int yto);
331 
332 
336     void redraw();
337 
338 
```

```
342     void clearLayers();
343
344
348     void clearLayer(int layer);
349
350
355     void setKeepAspectRatio(bool state);
356
357
363     void setNavigationLayer(int layer, int navigation);
364
365
366 protected:
367
368
373     void wheelEvent(QWheelEvent *event);
374
375
380     void mousePressEvent(QMouseEvent *event);
381
382
387     void mouseMoveEvent(QMouseEvent *event);
388
389
394     void mouseReleaseEvent(QMouseEvent *event);
395
396
401     void mouseDoubleClickEvent(QMouseEvent *event);
402
403
408     void resizeEvent(QResizeEvent *event);
409
410
411 signals:
412
413
421     void updateFilter(int xmin, int xmax, int ymin, int ymax);
422
423
431     void updateCorrelation(int xmin, int xmax, int ymin, int ymax);
432
433
442     void updateHistologyPosition(int x, int y, int width, int height, double angle);
443
444
455     void updateMicroscopyPosition(bool flipx, bool flipy, double x, double y, double width, double
      height, double angle);
456
457
462     void updateRuler(double size);
463
464
469     void imageWidgetDoubleClicked(int spectrumid);
470
471
472 private:
473
474     QWidget* parent;
475     QGraphicsScene* scene;
476
477     vector<cCoordinateInfo> coordinateinfo;
478     bool operatortype;
479     string columnname1;
480     string comparatorname1;
481     string filterstring1;
482     string columnname2;
483     string comparatorname2;
484     string filterstring2;
485     bool casesensitive;
486     bool wholeword;
487
488     bool colorscale;
489     bool absoluteintensity;
490
491     vector<layerInfo> layersvector;
492     int activelayer;
493
494     int xfrom;
495     int xto;
496     int yfrom;
497     int yto;
498
499     int leftshift;
500     int topshift;
501     int maxx;
502     int maxy;
503     int pixelsizex;
```

```
504     int pixelsizey;
505
506     int defaultmaxx;
507     int defaultmaxy;
508     int defaultpixelsizex;
509     int defaultpixelsizey;
510     eVendorType vendor;
511
512     qreal currentscale;
513     qreal factor;
514
515     int rulersize;
516     double rulervalue;
517
518     int pressedx;
519     int pressedy;
520     int currentx;
521     int currenty;
522     int currentwidth;
523     int currentheight;
524
525     bool showselection;
526     bool keepaspectratio;
527
528     qreal microscopynavigationcenterx;
529     qreal microscopynavigationcentery;
530
531     QGraphicsRectItem* selectionrect;
532     QGraphicsSimpleTextItem* selectionsimpletextitem;
533
534     QGraphicsRectItem* currentrect;
535     eCursorActivityType cursoractivity;
536
537     void redrawScene();
538
539     void drawMicroscopyImage(int layer, QRectF& rect_scene, int currentwidth, int currentheight);
540
541     void updateSelectionGroup();
542
543     void getRectanglePoints(qreal& rx1, qreal& ry1, qreal& rx2, qreal& ry2);
544
545     double prepareTransformation(QRectF rect, QTransform& transform, bool testedfliphorizontally, bool
    testedflipvertically, double testedangle, bool flipaxes);
546
547     void fixRectCenter(qreal& rx1, qreal& ry1, qreal& rx2, qreal& ry2, bool testedfliphorizontally, bool
    testedflipvertically, double testedangle);
548
549     double recalculateRulerValue(qreal currentscale);
550
551
552 private slots:
553
554
555     void zoomIn();
556
557
558     void zoomOut();
559
560
561     void normalSize();
562
563
564     void setRulerValue(double value);
565
566
567     void scrollbarValueChanged(int value);
568
569
570     void absoluteIntensityStateChanged(bool state);
571
572
573     void colorScaleStateChanged(bool state);
574
575
576     void changeLayer(int layerid, bool checked, int alpha, int zvalue, bool redraw);
577
578
579     void changeActiveLayer(int layerid);
580
581
582     void flipHistologyHorizontallyStateChanged(bool state);
583
584
585     void flipHistologyVerticallyStateChanged(bool state);
586
587
588     void showSelectionStateChanged(bool state);
```

```
589
590 };
591
592 #endif
```

17.99 gui/cLinearWidget.h File Reference

Visualization of a linear peptide.

```
#include <QWidget>
#include "core/cTheoreticalSpectrum.h"
```

Classes

- struct clonLabel

  Auxiliary structure for visualization of labels of fragment ions.
- struct hash_clonLabel

  The structure defining a hash function of clonLabel.
- class cLinearWidget

  Visualization of a linear peptide.

Functions

- bool operator== (clonLabel const &a, clonLabel const &b)

  Overloaded operator ==.
- void insertLabel (unordered_set< clonLabel, hash_clonLabel > &labels, int x, int y, string &description, bool alignright)

  Insert a fragment ion label into an unordered set of labels and concatenate descriptions of labels on the same coordinates.
- void drawLabels (QPainter &painter, unordered_set< clonLabel, hash_clonLabel > &labels, int width)

  Draw labels of fragment ions.

17.99.1 Detailed Description

Visualization of a linear peptide.

17.99.2 Function Documentation

17.99.2.1 drawLabels()

```
void drawLabels (
            QPainter & painter,
            unordered_set< cIonLabel, hash_cIonLabel > & labels,
            int width )
```

Draw labels of fragment ions.

Parameters

painter	reference to a painter
labels	unordered set of labels
width	widget width

17.99.2.2 insertLabel()

```
void insertLabel (
            unordered_set< cIonLabel, hash_cIonLabel > & labels,
            int x,
            int y,
            string & description,
            bool alignright )
```

Insert a fragment ion label into an unordered set of labels and concatenate descriptions of labels on the same coordinates.

Parameters

labels	unordered set of labels
x	x coordinate
y	y coordinate
description	a description of a fragment ion
alignright	true when the description is aligned right

17.99.2.3 operator==()

```
bool operator== (
            cIonLabel const & a,
            cIonLabel const & b )
```

Overloaded operator ==.

Parameters

a	first ion label
b	second ion label

Return values

bool	true when labels have equal coordinates

17.100 cLinearWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CLINEARWIDGET_H
8 #define _CLINEARWIDGET_H
9
10 #include <QWidget>
11
12 #include "core/cTheoreticalSpectrum.h"
13
14
15 // forward declaration
16 class QPaintEvent;
17
18
22 struct cIonLabel {
23
27     int x;
28
29
33     int y;
34
35
39     set<string> labelset;
```

```
40
41
45      bool alignright;
46
47
51      cIonLabel() {
52          clear();
53      }
54
55
59      void clear() {
60          x = 0;
61          y = 0;
62          labelset.clear();
63          alignright = false;
64      }
65
66  };
67
68
72  struct hash_cIonLabel {
73
79      size_t operator()(const cIonLabel& label)const {
80          return std::hash<string>()(to_string(label.x) + to_string(label.y));
81      }
82
83  };
84
85
92  bool operator == (cIonLabel const& a, cIonLabel const& b);
93
94
103 void insertLabel(unordered_set<cIonLabel, hash_cIonLabel>& labels, int x, int y, string& description,
      bool alignright);
104
105
112 void drawLabels(QPainter& painter, unordered_set<cIonLabel, hash_cIonLabel>& labels, int width);
113
114
118 class cLinearWidget :  public QWidget
119 {
120     Q_OBJECT
121
122 public:
123
124
128     cLinearWidget();
129
130
136     void initialize(cParameters* parameters, cTheoreticalSpectrum* theoreticalspectrum);
137
138
144     void exportToPDF(QString filename, bool postscript);
145
146
151     void exportToPNG(QString filename);
152
153
158     void exportToSVG(QString filename);
159
160
165     void setReportIsomers(bool reportisomers);
166
167
172     void hidePeakLabels(bool state);
173
174
175 protected:
176
177
182     void paintEvent(QPaintEvent *event);
183
184
185 private:
186
187     void paint(QPainter& painter);
188
189     cParameters* parameters;
190     cTheoreticalSpectrum* theoreticalspectrum;
191
192     string visibleionseries;
193     string visibleneutralloss;
194
195     bool reportisomers;
196     bool hidelabels;
197
198
```

```
199 private slots:
200
201     void ionSeriesChanged(QString text);
202
203     void neutralLossChanged(QString text);
204
205 };
206
207 #endif
```

17.101 gui/cMainThread.h File Reference

The thread launched by the command 'Search->Run'.

```
#include <iostream>
#include <QThread>
#include <QThreadPool>
#include <QTime>
#include <QMutex>
#include "core/utilities.h"
#include "core/cParameters.h"
#include "core/cDeNovoGraph.h"
#include "core/cTheoreticalSpectrum.h"
#include "core/cIsotopePatternCache.h"
```

Classes

- class cMainThread

  The class representing a thread launched by the command 'Search->Run'.

Variables

- cIsotopePatternCache **isotopepatterncache**

  Global isotope pattern cache.

17.101.1 Detailed Description

The thread launched by the command 'Search->Run'.

17.102 cMainThread.h

Go to the documentation of this file.

```
1
7 #ifndef _CMAINTHREAD_H
8 #define _CMAINTHREAD_H
9
10 #include <iostream>
11 #include <QThread>
12 #include <QThreadPool>
13 #include <QTime>
14 #include <QMutex>
15
16 #include "core/utilities.h"
17 #include "core/cParameters.h"
18 #include "core/cDeNovoGraph.h"
19 #include "core/cTheoreticalSpectrum.h"
20 #include "core/cIsotopePatternCache.h"
21
22 using namespace std;
23
24
28 extern cIsotopePatternCache isotopepatterncache;
29
30
34 class cMainThread :  public QThread {
35
36     Q_OBJECT
37
```

```
38 private:
39
40     bool enablelogwindow;
41     bool enablestdout;
42     bool terminatecomputation;
43     bool graphreaderisworking;
44
45     cDeNovoGraph graph;
46     cTheoreticalSpectrumList* listoftheoreticalspectra;
47     cParameters parameters;
48
49     QMutex mutex;
50
51     int paralleloutputstate;
52     int paralleloutputsize;
53
54     string getCurrentTime(QTime& time);
55
56     void singleThreadMS1(cMainThread* os, QTime& time, cTheoreticalSpectrum* testspectrum = 0, vector<
       vector< vector<int> > >* testhintsindexvector = 0, cTheoreticalSpectrumList*
       testlistoftheoreticalspectra = 0, vector< vector<cPeaksList> >* testunmatchedpeaksvector = 0);
57
58     void multiThreadMS1(cMainThread* os, QTime& time, cTheoreticalSpectrum* testspectrum = 0, vector<
       vector< vector<int> > >* testhintsindexvector = 0, cTheoreticalSpectrumList*
       testlistoftheoreticalspectra = 0, vector< vector<cPeaksList> >* testunmatchedpeaksvector = 0);
59
60     void initParallelOutputState(int size);
61
62     void serializeSpectrumPool(vector<cTheoreticalSpectrum*>& spectrumpool, cTheoreticalSpectrum& ts);
63
64     void serializeHintsIndexPool(vector< vector< vector<int> > >& hintsindexpool, vector< vector<int> >&
       hintsindex);
65
66     void serializeUnmatchedPeaksPool(int peaklistseriesvectorid, vector< vector<cPeaksList> >&
       unmatchedpeakspool, vector< vector<cPeaksList> >& unmatchedpeaks);
67
68     void serializeTheoreticalSpectrumListPool(int peaklistseriesvectorid,
       vector<cTheoreticalSpectrumList*>& theoreticalspectrumlistpool, cTheoreticalSpectrumList&
       theoreticalspectrumlist);
69
70     void printMatrix(cPeaksList& peaklist);
71
72     void printIonEstimations(cPeaksList& peaklist);
73
74
75 public:
76
77
85     cMainThread(cParameters& parameters, cTheoreticalSpectrumList& listoftheoreticalspectra, bool
       enablelogwindow = true, bool enablestdout = true);
86
87
92     void setGraphReaderIsWorking(bool working);
93
94
99     bool isGraphReaderWorking();
100
101
105     typedef std::basic_ostream<char, std::char_traits<char> > CoutType;
106
107
111     typedef CoutType& (*StandardEndLine)(CoutType&);
112
113
117     template<class T> cMainThread& operator<<(const T& x) {
118         if (enablelogwindow) {
119             emit message(to_string(x).c_str());
120         }
121
122         if (enablestdout) {
123             cout << x;
124         }
125
126         return *this;
127     }
128
129
133     cMainThread& operator<<(const char* x);
134
135
139     cMainThread& operator<<(const string& x);
140
141
145     cMainThread& operator<<(StandardEndLine manip);
146
147
152     void addToParallelOutputState(int value);
```

```
153
154
155 protected:
156
157
161     void run();
162
163
167     void emitEndSignals();
168
169
170 signals:
171
172
177     void message(QString s);
178
179
184     void enableRunButtonAndSettings(bool enable);
185
186
191     void enableStopButton(bool enable);
192
193
198     void enableButtonsHandlingResults(bool enable);
199
200
205     void sendParameters(cParameters parameters);
206
207
211     void reportSpectra();
212
213
218     void setGraph(string s);
219
220
221 private slots:
222
223     void stopComputation();
224
225     void graphReaderFinished();
226
227 };
228
229
230 #endif
231
```

17.103 gui/cMainWindow.h File Reference

The main window of the application.

```
#include <QMainWindow>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include <QFile>
#include <QTableView>
#include <QStandardItemModel>
#include <QStandardItem>
#include <QItemDelegate>
#include <QComboBox>
#include "core/utilities.h"
#include "core/cTheoreticalSpectrum.h"
#include "gui/cAboutWidget.h"
#include "gui/cGraphWidget.h"
#include "gui/cParametersWidget.h"
#include "gui/cSpectrumDetailWidget.h"
#include "gui/cBricksDatabaseWidget.h"
#include "gui/cSequenceDatabaseWidget.h"
#include "gui/cModificationsWidget.h"
#include "gui/cDrawPeptideWidget.h"
#include "gui/cSummaryPeaksTableWidget.h"
#include "gui/cMainThread.h"
#include "gui/cHTMLExportDialog.h"
```

```
#include "gui/cImageWindow.h"
#include "gui/cChromatogramWindow.h"
#include "gui/cMainWindowProxyModel.h"
#include "gui/cPreferencesDialog.h"
#include "gui/cPatternSimulatorWidget.h"
#include "gui/cFormulaPredictor.h"
#include "gui/cMultipleDatasetsTableWidget.h"
```

Classes

- class cMainWindow

  The class representing the main window of the application.

Functions

- **Q_DECLARE_METATYPE** (string)

  Register string by Qt.

17.103.1 Detailed Description

The main window of the application.

17.104 cMainWindow.h

Go to the documentation of this file.

```
1
7 #ifndef _MAINWINDOW_H
8 #define _MAINWINDOW_H
9
10 #include <QMainWindow>
11 #include <QDesktopServices>
12 #include <QUrl>
13 #include <QFileInfo>
14 #include <QFile>
15 #include <QTableView>
16 #include <QStandardItemModel>
17 #include <QStandardItem>
18 #include <QItemDelegate>
19 #include <QComboBox>
20 #include "core/utilities.h"
21 #include "core/cTheoreticalSpectrum.h"
22 #include "gui/cAboutWidget.h"
23 #include "gui/cGraphWidget.h"
24 #include "gui/cParametersWidget.h"
25 #include "gui/cSpectrumDetailWidget.h"
26 #include "gui/cBricksDatabaseWidget.h"
27 #include "gui/cSequenceDatabaseWidget.h"
28 #include "gui/cModificationsWidget.h"
29 #include "gui/cDrawPeptideWidget.h"
30 #include "gui/cSummaryPeaksTableWidget.h"
31 #include "gui/cMainThread.h"
32 #include "gui/cHTMLExportDialog.h"
33 #include "gui/cImageWindow.h"
34 #include "gui/cChromatogramWindow.h"
35 #include "gui/cMainWindowProxyModel.h"
36 #include "gui/cPreferencesDialog.h"
37 #include "gui/cPatternSimulatorWidget.h"
38 #include "gui/cFormulaPredictor.h"
39 #include "gui/cMultipleDatasetsTableWidget.h"
40
41
42 // forward declaration
43 class QTextEdit;
44 class QSplitter;
45 class QTableWidget;
46
47
51 class cMainWindow :  public QMainWindow
52 {
53     Q_OBJECT
54
55 public:
```

```
56
57
61      cMainWindow();
62
63
67      ~cMainWindow();
68
69
70  protected:
71
76      void keyPressEvent(QKeyEvent *event);
77
78
79  private:
80
81      QMenuBar* menuBar;
82
83      QMenu* menuFile;
84      QMenu* menuSearch;
85      QMenu* menuTools;
86      QMenu* menuBookmarks;
87      QMenu* menuView;
88      QMenu* menuHelp;
89
90      QToolBar* toolbarFile;
91      QToolBar* toolbarSearch;
92      QToolBar* toolbarTools;
93      QToolBar* toolbarBookmarks;
94      QToolBar* toolbarView;
95      QToolBar* toolbarHelp;
96      QToolBar* toolbarInputFilter;
97      QToolBar* toolbarRowsFilter;
98
99      QAction* actionOpenResults;
100      QAction* actionSaveResults;
101      QAction* actionExportToCsv;
102      QAction* actionExportToHTML;
103      QAction* actionPreferences;
104      QAction* actionQuit;
105      QAction* actionRun;
106      QAction* actionStop;
107      QAction* actionProperties;
108      QAction* actionBricksDatabase;
109      QAction* actionSequenceDatabase;
110      QAction* actionModifications;
111      QAction *actionDrawPeptide;
112      QAction *actionBookMark1;
113      QAction *actionBookMark2;
114      QAction *actionBookMark3;
115      QAction *actionBookMark4;
116      QAction *actionBookMark5;
117      QAction *actionBookMark6;
118      QAction *actionBookMark7;
119      QAction *actionBookMark8;
120      QAction *actionBookMark9;
121      QAction *actionBookMark10;
122      QAction* actionShowIsomers;
123      QAction* actionGraph;
124      QAction* actionSummaryTableOfMatchedPeaks;
125      QAction* actionMultipleDatasets;
126      QAction* actionChromatogramWindow;
127      QAction* actionImageWindow;
128      QAction* actionPatternSimulatorWindow;
129      QAction* actionFormulaPredictor;
130      QAction* actionLog;
131      QAction* actionHTMLDocumentation;
132      QAction* actionPDFManual;
133      QAction* actionAbout;
134
135      QWidget* inputfilterwidget;
136      QHBoxLayout* inputfilterhbox;
137      QComboBox* inputfiltercombobox;
138      QPushButton* inputfilterbutton;
139
140      QWidget* rowsfilterwidget;
141      QHBoxLayout* rowsfilterhbox;
142      QComboBox* rowsfilteroperator;
143      QComboBox* rowsfiltercombobox1;
144      QComboBox* rowsfiltercomparatorcombobox1;
145      QLineEdit* rowsfilterline1;
146      QComboBox* rowsfiltercombobox2;
147      QComboBox* rowsfiltercomparatorcombobox2;
148      QLineEdit* rowsfilterline2;
149      QCheckBox* rowsfiltercasesensitive;
150      QCheckBox* rowsfilterwholeword;
151      QPushButton* rowsfilterbutton;
152      QPushButton* rowsfilterclearbutton;
```

```
153
154     QTableView* results;
155     QStandardItemModel* resultsmodel;
156     cMainWindowProxyModel* resultsproxymodel;
157
158     cTheoreticalSpectrumList listoftheoreticalspectra;
159     cPeakListSeries eicchromatograms;
160     vector<cSpectrumDetailWidget> spectradetails;
161     vector<cSpectrumDetailWidget> avgspectra;
162
163     cGlobalPreferences globalpreferences;
164     cParameters parameters;
165
166     cPeakListSeries rawdata;
167     vector<cImzMLItem> imzmlprofilemetadata;
168     bool profilemz64precision;
169     bool profileintensity64precision;
170
171     QTextEdit* logWindow;
172     QSplitter* splitter;
173
174     cAboutWidget* about;
175     cGraphWidget* graph;
176     cBricksDatabaseWidget* bricksdatabasewidget;
177     cSequenceDatabaseWidget* sequencedatabasewidget;
178     cModificationsWidget* modificationswidget;
179     cDrawPeptideWidget* drawpeptidewidget;
180     cSummaryPeaksTableWidget* summarytableofmatchedpeaks;
181     cImageWindow* imagewindow;
182     cChromatogramWindow* chromatogramwindow;
183     cParametersWidget* parameterswidget;
184     cHTMLExportDialog* htmlexportdialog;
185     cPreferencesDialog* preferencesdialog;
186     cMultipleDatasetsTableWidget* multipledatasetswidget;
187     cPatternSimulatorWidget* patternsimulatorwidget;
188     cFormulaPredictor* formulapredictor;
189
190     int resultsbasecolumncount;
191     int resultsspecificcolumncount;
192     int searchspecificcolumncount;
193
194     QString lastdirexporttocsv;
195     QString lastdirexporttohtml;
196
197     QString lastdirsaveresults;
198     QString lastdiropenresults;
199
200     bool summarytableisprepared;
201     bool multipledatasetsisprepared;
202
203     int lastactivedetail;
204
205     QRect othernormalgeometry;
206     bool otherismaximized;
207     bool otherprofilespectrum;
208     bool otherabsoluteintensity;
209     bool othermatchedpeaks;
210     bool otherunmatchedpeaks;
211     bool otherscrambledpeaks;
212     bool otherlabelspeaks;
213     bool othermzselection;
214
215     int activefileid;
216
217     int avgspectrumused;
218     bool avgspectrumabsoluteintensityenabled;
219     QRect avgspectrumgeometry;
220     bool avgspectrummaximized;
221
222     bool quitapp;
223
224     void closeEvent(QCloseEvent *event);
225
226     void preparePeptideSequence(int row, string& peptidesequence, bool reportisomers);
227
228     void reportSpectrum(int row, cTheoreticalSpectrum& theoreticalspectrum, bool reportisomers);
229
230     void deleteResults();
231
232     void applyGlobalPreferences();
233
234     void openDetail(int rowid);
235
236     void calculateAvgSpectrum(int minscan, int maxscan);
237
238
239 private slots:
```

```
240
241     void showHideLog();
242
243     void showAbout();
244
245     void showBricksDatabase();
246
247     void showSequenceDatabase();
248
249     void showModifications();
250
251     void showDrawPeptideWidget();
252
253     void setAndShowDrawPeptideWidget(int peptidetypeindex, QString sequence);
254
255     bool prepareSummaryTableOfMatchedPeaks();
256
257     void showSummaryTableOfMatchedPeaks();
258
259     void showImageWindow();
260
261     void showChromatogramWindow();
262
263     void showGraph();
264
265     void showProperties();
266
267     void showMultipleDatasetsOverview();
268
269     void showPatternSimulatorWindow();
270
271     void showFormulaPredictor();
272
273     void run();
274
275     void stop();
276
277     void updateLog(QString qstr);
278
279     void enableRunButtonAndSettings(bool enable);
280
281     void enableStopButton(bool enable);
282
283     void enableButtonsHandlingResults(bool enable);
284
285     void sendParameters(cParameters parameters);
286
287     void reportSpectra();
288
289     void showIsomersStateChanged();
290
291     void rowDoubleClicked(const QModelIndex& item);
292
293     void setGraph(string s);
294
295     void exportToCsv();
296
297     void exportToHTML();
298
299     void showPreferences();
300
301     void showHTMLDocumentation();
302
303     void showPDFManual();
304
305     void saveResultsFile();
306
307     void openResultsFile();
308
309     void setSequence(int peptidetypeindex, QString sequence);
310
311     void setTag(int peptidetypeindex, QString tag);
312
313     void insertSequence(int peptidetypeindex, QString tag);
314
315     void quitApplication();
316
317     void filterResults();
318
319     void resetFilter();
320
321     void lastActiveDetailSlot(int rowid);
322
323     void gotoBookMark1();
324
325     void gotoBookMark2();
326
```

```
327     void gotoBookMark3();
328
329     void gotoBookMark4();
330
331     void gotoBookMark5();
332
333     void gotoBookMark6();
334
335     void gotoBookMark7();
336
337     void gotoBookMark8();
338
339     void gotoBookMark9();
340
341     void gotoBookMark10();
342
343     void summaryPeaksTableCancelled();
344
345     void multipleDatasetsTableCancelled();
346
347     void summaryPeaksTableRowDoubleClicked(int rowid, double experimentalmz);
348
349     void multipleDatasetsTableRowClicked(int datasetid, int spectrumid, int peakid, int datatypeview,
    int peakshape, string name, string iontype, string mzstr);
350
351     void updateSummaryPeaksTableFilterSlot(int xmin, int xmax, int ymin, int ymax);
352
353     void chromatogramDoubleClicked(int scanid);
354
355     void imageWindowDoubleClicked(int spectrumid);
356
357     void inputFilterButtonReleased();
358
359     void filterSummaryTableIfPrepared();
360
361     void showAvgSpectrum(int minscan, int maxscan);
362
363     void avgSpectrumAbsoluteIntensityStateChanged(bool state);
364
365     //void showContextMenu(const QPoint &pt);
366
367 signals:
368
372     void stopComputation();
373
374 };
375
376
380 Q_DECLARE_METATYPE(string);
381
382
383 #endif
```

17.105 gui/cMainWindowProxyModel.h File Reference

A table proxy model.

```
#include <QString>
#include <QVariant>
#include <QComboBox>
#include <QSortFilterProxyModel>
#include <QLineEdit>
#include "core/utilities.h"
```

Classes

- class cMainWindowProxyModel

  A table proxy model.

17.105.1 Detailed Description

A table proxy model.

17.106 cMainWindowProxyModel.h

Go to the documentation of this file.

```
1
7 #ifndef _CMAINWINDOWPROXYMODEL_H
8 #define _CMAINWINDOWPROXYMODEL_H
9
10 #include <QString>
11 #include <QVariant>
12 #include <QComboBox>
13 #include <QSortFilterProxyModel>
14 #include <QLineEdit>
15 #include "core/utilities.h"
16
17
21 class cMainWindowProxyModel :  public QSortFilterProxyModel {
22
23     Q_OBJECT
24
25 private:
26
27     QComboBox* filteroperator;
28     QComboBox* filtercombobox1;
29     QComboBox* filtercomparatorcombobox1;
30     QLineEdit* filterline1;
31     QComboBox* filtercombobox2;
32     QComboBox* filtercomparatorcombobox2;
33     QLineEdit* filterline2;
34     bool wholeword;
35
36 public:
37
38
43     cMainWindowProxyModel(QObject *parent = 0);
44
45
56     void initialize(QComboBox* rowsfilteroperator, QComboBox* rowsfiltercombobox1, QComboBox*
       rowsfiltercomparatorcombobox1, QLineEdit* rowsfilterline1, QComboBox* rowsfiltercombobox2, QComboBox*
       rowsfiltercomparatorcombobox2, QLineEdit* rowsfilterline2);
57
58
63     void setWholeWord(bool wholeword);
64
65
66 protected:
67
68
75     bool filterAcceptsRow(int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE;
76
77
84     bool lessThan(const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE;
85
86
87 };
88
89
90 #endif
91
```

17.107 gui/cModificationsProxyModel.h File Reference

A proxy model for editor of database of modifications.

```
#include <QString>
#include <QVariant>
#include <QComboBox>
#include <QSortFilterProxyModel>
#include <QLineEdit>
#include "core/utilities.h"
```

Classes

- class cModificationsProxyModel

  A proxy model for editor of database of modifications.

17.107.1 Detailed Description

A proxy model for editor of database of modifications.

17.108 cModificationsProxyModel.h

Go to the documentation of this file.

```
1
7 #ifndef _CMODIFICATIONSPROXYMODEL_H
8 #define _CMODIFICATIONSPROXYMODEL_H
9
10 #include <QString>
11 #include <QVariant>
12 #include <QComboBox>
13 #include <QSortFilterProxyModel>
14 #include <QLineEdit>
15 #include "core/utilities.h"
16
17
21 class cModificationsProxyModel :  public QSortFilterProxyModel {
22
23     Q_OBJECT
24
25 private:
26
27     QComboBox* filteroperator;
28     QComboBox* filtercombobox1;
29     QComboBox* filtercomparatorcombobox1;
30     QLineEdit* filterline1;
31     QComboBox* filtercombobox2;
32     QComboBox* filtercomparatorcombobox2;
33     QLineEdit* filterline2;
34     bool wholeword;
35
36 public:
37
38
43     cModificationsProxyModel(QObject *parent = 0);
44
45
56     void initialize(QComboBox* rowsfilteroperator, QComboBox* rowsfiltercombobox1, QComboBox*
      rowsfiltercomparatorcombobox1, QLineEdit* rowsfilterline1, QComboBox* rowsfiltercombobox2, QComboBox*
      rowsfiltercomparatorcombobox2, QLineEdit* rowsfilterline2);
57
58
63     void setWholeWord(bool wholeword);
64
65
66 protected:
67
68
75     bool filterAcceptsRow(int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE;
76
77
84     bool lessThan(const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE;
85
86
87 };
88
89
90 #endif
91
```

17.109 gui/cModificationsWidget.h File Reference

An editor of N-/C-terminal modifications database.

```
#include <QWidget>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include <QMainWindow>
#include <QToolBar>
#include <QAction>
#include <QTableView>
#include <QStandardItemModel>
#include <QStandardItem>
```

```
#include <QComboBox>
#include <fstream>
#include "core/utilities.h"
#include "core/cGlobalPreferences.h"
#include "core/cFragmentIons.h"
#include "core/cSummaryFormula.h"
#include "gui/cModificationsProxyModel.h"
#include "gui/cCheckBoxDelegate.h"
```

Classes

- class cModificationsWidget

  An editor of N-/C-terminal modifications database.

17.109.1 Detailed Description

An editor of N-/C-terminal modifications database.

17.110 cModificationsWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CMODIFICATIONSWIDGET_H
8 #define _CMODIFICATIONSWIDGET_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QUrl>
13 #include <QFileInfo>
14 #include <QMainWindow>
15 #include <QToolBar>
16 #include <QAction>
17 #include <QTableView>
18 #include <QStandardItemModel>
19 #include <QStandardItem>
20 #include <QComboBox>
21 #include <fstream>
22 #include "core/utilities.h"
23 #include "core/cGlobalPreferences.h"
24 #include "core/cFragmentIons.h"
25 #include "core/cSummaryFormula.h"
26 #include "gui/cModificationsProxyModel.h"
27 #include "gui/cCheckBoxDelegate.h"
28
29
30 // forward declaration
31 class QHBoxLayout;
32 class QVBoxLayout;
33 class QPushButton;
34 class QLineEdit;
35 class QMenuBar;
36 class QMenu;
37
38
42 class cModificationsWidget :  public QMainWindow
43 {
44     Q_OBJECT
45
46 public:
47
48
54     cModificationsWidget(cGlobalPreferences* globalpreferences, QWidget* parent = (QWidget *)0);
55
56
60     ~cModificationsWidget();
61
62
67     void closeEvent(QCloseEvent *event);
68
69
74     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
75
76
77 private:
```

```
78
79      QString editorname;
80
81      QWidget* parent;
82
83      cGlobalPreferences* globalpreferences;
84
85      QMenuBar* menuBar;
86      QMenu* menuFile;
87      QMenu* menuEdit;
88      QMenu* menuHelp;
89
90      QToolBar* toolbarFile;
91      QAction* actionNewDatabase;
92      QAction* actionOpenDatabase;
93      QAction* actionSaveDatabase;
94      QAction* actionSaveDatabaseAs;
95      QAction* actionImportDatabase;
96      QAction* actionCloseWindow;
97
98      QToolBar* toolbarEdit;
99      QAction* actionAddRow;
100      QAction* actionRemoveSelectedRows;
101      QAction* actionSelectAll;
102      QAction* actionUnselectAll;
103
104      QToolBar* toolbarHelp;
105      QAction* actionHTMLDocumentation;
106
107      QToolBar* toolbarFilter;
108      QWidget* rowsfilterwidget;
109      QHBoxLayout* rowsfilterhbox;
110      QComboBox* rowsfilteroperator;
111      QComboBox* rowsfiltercombobox1;
112      QComboBox* rowsfiltercomparatorcombobox1;
113      QLineEdit* rowsfilterline1;
114      QComboBox* rowsfiltercombobox2;
115      QComboBox* rowsfiltercomparatorcombobox2;
116      QLineEdit* rowsfilterline2;
117      QCheckBox* rowsfiltercasesensitive;
118      QCheckBox* rowsfilterwholeword;
119      QPushButton* rowsfilterbutton;
120      QPushButton* rowsfilterclearbutton;
121
122      QTableView* database;
123      QStandardItemModel* databasemodel;
124      cModificationsProxyModel* proxymodel;
125      QVBoxLayout* mainlayout;
126      QWidget* mainwidget;
127
128      QString databasefile;
129      QString lastdir;
130      ifstream inputstream;
131      ofstream outputstream;
132      vector<cFragmentIonType> modifications;
133
134      bool datamodified;
135
136      void deleteTable();
137
138      void resetHeader();
139
140      bool checkTable();
141
142      bool checkFormula(int row, const string& summary);
143
144      void setDataModified(bool datamodified);
145
146  protected:
147
148
153      void keyPressEvent(QKeyEvent *event);
154
155
156  private slots:
157
158      void closeWindow();
159
160      void openDatabase();
161
162      bool saveDatabase();
163
164      bool saveDatabaseAs();
165
166      void saveDatabaseCheck();
167
168      void addRow();
```

```
169
170     void removeSelectedRows();
171
172     void itemChanged(QStandardItem* item);
173
174     void headerItemClicked(int index);
175
176     void filterRows();
177
178     void resetFilter();
179
180     void createNewDatabase();
181
182     void importDatabase();
183
184     void selectAll();
185
186     void unselectAll();
187
188     void showHTMLDocumentation();
189
190 };
191
192 #endif
```

17.111 gui/cMultipleButtonDelegate.h File Reference

A delegate to create multiple buttons.

```
#include <QString>
#include <QLocale>
#include <QVariant>
#include <QStyledItemDelegate>
#include <QPainter>
#include <QModelIndex>
#include <QEvent>
#include <QAbstractItemModel>
#include "core/utilities.h"
```

Classes

- class cMultipleButtonDelegate

  A delegate to create multiple buttons.

17.111.1 Detailed Description

A delegate to create multiple buttons.

17.112 cMultipleButtonDelegate.h

Go to the documentation of this file.

```
1
7 #ifndef _CMULTIPLEBUTTONDELEGATE_H
8 #define _CMULTIPLEBUTTONDELEGATE_H
9
10 #include <QString>
11 #include <QLocale>
12 #include <QVariant>
13 #include <QStyledItemDelegate>
14 #include <QPainter>
15 #include <QModelIndex>
16 #include <QEvent>
17 #include <QAbstractItemModel>
18 #include "core/utilities.h"
19
23 class cMultipleButtonDelegate :  public QStyledItemDelegate {
24
25     Q_OBJECT
26
27 public:
28
```

```
29
34      cMultipleButtonDelegate(QObject* parent = (QObject *)0);
35
36
43      QString displayText(const QVariant& value, const QLocale& locale) const;
44
45
52      void paint(QPainter* painter, const QStyleOptionViewItem& option, const QModelIndex& index) const;
53
54
55  protected:
56
57
65      bool editorEvent(QEvent* event, QAbstractItemModel* model, const QStyleOptionViewItem& option, const
        QModelIndex& index);
66
67  };
68
69
70  #endif
71
```

17.113 gui/cMultipleDatasetsTableProxyModel.h File Reference

A table proxy model.

```
#include <QString>
#include <QVariant>
#include <QSortFilterProxyModel>
#include <QComboBox>
#include <QLineEdit>
#include "core/cParameters.h"
```

Classes

- class cMultipleDatasetsTableProxyModel

  A table proxy model.

17.113.1 Detailed Description

A table proxy model.

17.114 cMultipleDatasetsTableProxyModel.h

Go to the documentation of this file.

```
1
7   #ifndef _CMULTIPLEDATASETSTABLEPROXYMODEL_H
8   #define _CMULTIPLEDATASETSTABLEPROXYMODEL_H
9
10  #include <QString>
11  #include <QVariant>
12  #include <QSortFilterProxyModel>
13  #include <QComboBox>
14  #include <QLineEdit>
15  #include "core/cParameters.h"
16
17
21  class cMultipleDatasetsTableProxyModel :  public QSortFilterProxyModel {
22
23      Q_OBJECT
24
25  public:
26
31      cMultipleDatasetsTableProxyModel(QObject *parent = 0);
32
33  protected:
34
41      bool lessThan(const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE;
42
43  };
44
45
```

```
46 #endif
47
```

17.115 gui/cMultipleDatasetsTableWidget.h File Reference

Aggregated results from multiple datasets.

```
#include <QDesktopServices>
#include <QUrl>
#include <QMainWindow>
#include <QToolBar>
#include <QAction>
#include <QTableView>
#include <QStandardItemModel>
#include <QStandardItem>
#include <QItemDelegate>
#include <QMessageBox>
#include <QApplication>
#include <QComboBox>
#include <QPushButton>
#include <QLabel>
#include <QSpinBox>
#include <QDoubleSpinBox>
#include <QHBoxLayout>
#include "core/utilities.h"
#include "core/cGlobalPreferences.h"
#include "gui/cMultipleDatasetsTableProxyModel.h"
#include "gui/cPubChemSearchWidget.h"
#include "gui/cEventFilter.h"
#include "gui/cChromatogramWindowWidget.h"
#include "gui/cDefineCalibrationWidget.h"
#include "gui/cCalibrationChart.h"
```

Classes

- class cMultipleDatasetsTableWidget

  Aggregated results from multiple datasets.

Functions

- void calculateLineParameterA (vector< double > &x, vector< double > &y, double &a)

  Calculate parameter a of a line.
- void calculateLineParameters (vector< double > &x, vector< double > &y, double &a, double &b)

  Calculate parameters a and b of a line.
- void calculateLineParametersWithIterations (vector< double > &x, vector< double > &y, int iterations, double &a, double &b)

  Calculate parameters a and b of a line.

17.115.1 Detailed Description

Aggregated results from multiple datasets.

17.115.2 Function Documentation

17.115.2.1 calculateLineParameterA()

```
void calculateLineParameterA (
            vector< double > & x,
            vector< double > & y,
            double & a )
```

Calculate parameter a of a line.

Parameters

x	input vector of x-coordinates
y	input vector of y-coordinates
a	output value a

17.115.2.2 calculateLineParameters()

```
void calculateLineParameters (
            vector< double > & x,
            vector< double > & y,
            double & a,
            double & b )
```

Calculate parameters a and b of a line.

Parameters

x	input vector of x-coordinates
y	input vector of y-coordinates
a	output value a
b	output value b

17.115.2.3 calculateLineParametersWithIterations()

```
void calculateLineParametersWithIterations (
            vector< double > & x,
            vector< double > & y,
            int iterations,
            double & a,
            double & b )
```

Calculate parameters a and b of a line.

Parameters

x	input vector of x-coordinates
y	input vector of y-coordinates
iterations	number of iterations
a	output value a
b	output value b

17.116 cMultipleDatasetsTableWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CMULTIPLEDATASETSTABLEWIDGET_H
8 #define _CMULTIPLEDATASETSTABLEWIDGET_H
9
10 #include <QDesktopServices>
11 #include <QUrl>
12 #include <QMainWindow>
13 #include <QToolBar>
14 #include <QAction>
15 #include <QTableView>
16 #include <QStandardItemModel>
17 #include <QStandardItem>
18 #include <QItemDelegate>
19 #include <QMessageBox>
20 #include <QApplication>
21 #include <QComboBox>
22 #include <QPushButton>
23 #include <QLabel>
24 #include <QSpinBox>
25 #include <QDoubleSpinBox>
26 #include <QHBoxLayout>
27
28 #include "core/utilities.h"
29 #include "core/cGlobalPreferences.h"
30 #include "gui/cMultipleDatasetsTableProxyModel.h"
31 #include "gui/cPubChemSearchWidget.h"
32 #include "gui/cEventFilter.h"
33 #include "gui/cChromatogramWindowWidget.h"
34 #include "gui/cDefineCalibrationWidget.h"
35 #include "gui/cCalibrationChart.h"
36
37
38 // forward declarations
39 class cParameters;
40 class cTheoreticalSpectrumList;
41 class QVBoxLayout;
42 class QMenuBar;
43 class QMenu;
44
45
52 void calculateLineParameterA(vector<double>& x, vector<double>& y, double& a);
53
54
62 void calculateLineParameters(vector<double>& x, vector<double>& y, double& a, double& b);
63
64
73 void calculateLineParametersWithIterations(vector<double>& x, vector<double>& y, int iterations, double&
    a, double& b);
74
75
79 class cMultipleDatasetsTableWidget :  public QMainWindow {
80
81     Q_OBJECT
82
83 public:
84
85
91     cMultipleDatasetsTableWidget(cGlobalPreferences* globalpreferences, QWidget* parent = (QWidget *)0);
92
93
97     ~cMultipleDatasetsTableWidget();
98
99
104     void closeEvent(QCloseEvent *event);
105
106
114     bool prepareToShow(cParameters* parameters, cTheoreticalSpectrumList* listoftheoreticalspectra, int
    activefileid);
115
116
120     void updateTable();
121
122
126     void resetStandard();
127
128
132     void updateListofStandards();
133
134
138     void deleteTable();
139
140
145     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
146
147
152     cPubChemSearchWidget* getPubChemSearchWidget();
153
```

```
154
159     cDefineCalibrationWidget* getDefineCalibrationWidget();
160
161
166     cCalibrationChart* getCalibrationChartWidget();
167
168
169 private:
170
171     QString title;
172
173     cGlobalPreferences* globalpreferences;
174     QWidget* parent;
175
176     cParameters* parameters;
177     cTheoreticalSpectrumList* listoftheoreticalspectra;
178
179     cPubChemSearchWidget* pubchemsearchwidget;
180     cDefineCalibrationWidget* definecalibrationwidget;
181     cCalibrationChart* calibrationchartwidget;
182
183     QMenuBar* menuBar;
184     QMenu* menuFile;
185     QMenu* menuSearch;
186     QMenu* menuHelp;
187
188     QToolBar* toolbarFile;
189     QAction* actionExportCSV;
190     QAction* actionExportDatabase;
191     QAction* actionCloseWindow;
192
193     QToolBar* toolbarView;
194
195     QToolBar* toolbarCalibration;
196
197     QToolBar* toolbarSearch;
198     QAction* actionSearchPubChem;
199
200     QToolBar* toolbarHelp;
201     QAction* actionHTMLDocumentation;
202
203     QLabel* datatypelabel;
204     QComboBox* comboboxdatatype;
205     QHBoxLayout* datatypehboxlayout;
206     QWidget* datatypewidget;
207
208     QLabel* peakshapelabel;
209     QComboBox* comboboxpeakshape;
210     QHBoxLayout* peakshapehboxlayout;
211     QWidget* peakshapewidget;
212
213     QLabel* standardlabel;
214     QComboBox* comboboxstandard;
215     QHBoxLayout* standardhboxlayout;
216     QWidget* standardwidget;
217
218     QLabel* calibrationlabely;
219     QDoubleSpinBox* calibrationspinboxa;
220     QLabel* calibrationlabelx;
221     QDoubleSpinBox* calibrationspinboxb;
222     QPushButton* calibrationsetbutton;
223     QPushButton* calibrationdefinebutton;
224     QPushButton* calibrationviewbutton;
225     QHBoxLayout* calibrationhboxlayout;
226     QWidget* calibrationwidget;
227
228     QLabel* columnsizelabel;
229     QSpinBox* columnsizespinbox;
230     QPushButton* columnsizesetbutton;
231     QPushButton* columnsizeresetbutton;
232     QHBoxLayout* columnsizehboxlayout;
233     QWidget* columnsizewidget;
234
235     QTableView* database;
236     QStandardItemModel* databasemodel;
237     cMultipleDatasetsTableProxyModel* proxymodel;
238     QVBoxLayout* mainlayout;
239     QWidget* mainwidget;
240
241     QString lastdirexporttocsv;
242     QString lastdirexportdatabase;
243
244     map<string, map<int, pair<int, int> > > identifieditems;
245     map<string, multimap<int, pair<int, int> > > identifieditemswithmultimap;
246
247     int activefileid;
248     int datatypeview;
```

```
249     int peakshape;
250     int standard;
251     double calibrationa;
252     double calibrationb;
253     int systemcolumns;
254 
255     void getEICFromMultiMap(int datafileid, multimap<int, pair<int, int> >& compoundmultimap,
      cPeaksList& eicchromatogram, bool highestpeaks);
256 
257 
258 protected:
259 
260 
265     void keyPressEvent(QKeyEvent *event);
266 
267 
268 private slots:
269 
270     void closeWindow();
271 
272 
273     void exportToCsv();
274 
275 
276     void exportToDatabase();
277 
278 
279     void dataTypeViewChanged(int index);
280 
281 
282     void peakShapeChanged(int index);
283 
284 
285     void standardChanged(int index);
286 
287 
288     void calibrationSetButtonReleased();
289 
290 
291     void calibrationDefineButtonReleased();
292 
293 
294     void calibrationViewButtonReleased();
295 
296 
297     void calibrationCurveRedefined();
298 
299 
300     void showHTMLDocumentation();
301 
302 
303     void rowClicked(const QModelIndex& item);
304 
305 
306     void searchPubChem();
307 
308 
309     void setColumnSize();
310 
311 
312     void resetColumnSize();
313 
314 
315 signals:
316 
317 
321     void tableCancelled();
322 
323 
327     void filterSummaryTable();
328 
329 
341     void multipleDatasetsTableRowClicked(int datasetid, int spectrumid, int peakid, int datatypeview,
      int peakshape, string name, string iontype, string mzstr);
342 
343 };
344 
345 #endif
346 
```

17.117 gui/cNeutralLossesListWidget.h File Reference

The implementation of a widget where types of neutral losses generated in theoretical spectra are selected.

```
#include <string>
#include <QListWidget>
#include <QHBoxLayout>
#include <QVBoxLayout>
```

Classes

- class cNeutralLossesListWidget

  The widget for selection of types of neutral losses.

17.117.1 Detailed Description

The implementation of a widget where types of neutral losses generated in theoretical spectra are selected.

17.118 cNeutralLossesListWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CNEUTRALLOSSESLISTWIDGET_H
8 #define _CNEUTRALLOSSESLISTWIDGET_H
9
10 #include <string>
11 #include <QListWidget>
12 #include <QHBoxLayout>
13 #include <QVBoxLayout>
14
15
16 // forward declaration
17 class QPushButton;
18 class QHBoxLayout;
19 class QVBoxLayout;
20
21
25 class cNeutralLossesListWidget :  public QWidget
26 {
27     Q_OBJECT
28
29 public:
30
31
38     cNeutralLossesListWidget(bool showdefaultbutton, bool showhconbutton, QObject* parent);
39
40
44     ~cNeutralLossesListWidget();
45
46
51     QListWidget* getList();
52
53
58     void addItem(QString formula);
59
60
61 private:
62     QObject* parent;
63
64     bool showdefaultbutton;
65     bool showhconbutton;
66
67     QListWidget* list;
68     QPushButton* selectallbutton;
69     QPushButton* clearallbutton;
70     QPushButton* addbutton;
71     QPushButton* removebutton;
72     QPushButton* defaultbutton;
73     QPushButton* hconbutton;
74
75     QHBoxLayout* hbox;
76     QVBoxLayout* vbox1;
77     QVBoxLayout* vbox2;
78
79
80 private slots:
81
82     void selectAllItems();
83
84     void clearAllItems();
```

```
85
86      void addEmptyItem();
87
88      void removeItem();
89
90      void setDefaultItems();
91
92      void setHCON();
93
94 };
95
96 #endif
```

17.119 gui/cParametersWidget.h File Reference

The widget for editing parameters of the application opened using the command 'Search->Settings'.

```
#include <iostream>
#include <string>
#include <QWidget>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include <QLineEdit>
#include <QPushButton>
#include "core/utilities.h"
#include "core/cGlobalPreferences.h"
#include "core/cSummaryFormula.h"
#include "core/cParameters.h"
#include "gui/cFragmentIonsListWidget.h"
#include "gui/cNeutralLossesListWidget.h"
```

Classes

- class cParametersWidget

  The class representing a widget for editing parameters of the application.

17.119.1 Detailed Description

The widget for editing parameters of the application opened using the command 'Search->Settings'.

17.120 cParametersWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CPARAMETERSWIDGET_H
8 #define _CPARAMETERSWIDGET_H
9
10 #include <iostream>
11 #include <string>
12
13 #include <QWidget>
14 #include <QDesktopServices>
15 #include <QUrl>
16 #include <QFileInfo>
17 #include <QLineEdit>
18 #include <QPushButton>
19
20 #include "core/utilities.h"
21 #include "core/cGlobalPreferences.h"
22 #include "core/cSummaryFormula.h"
23 #include "core/cParameters.h"
24 #include "gui/cFragmentIonsListWidget.h"
25 #include "gui/cNeutralLossesListWidget.h"
26
27
28 using namespace std;
29
```

```
30
31 // forward declarations
32 class QFormLayout;
33 class QGridLayout;
34 class QTextEdit;
35 class QVBoxLayout;
36 class QHBoxLayout;
37 class QComboBox;
38 class QSpinBox;
39 class QDoubleSpinBox;
40 class QDialogButtonBox;
41 class QGroupBox;
42 class QCheckBox;
43 class QTableWidget;
44 class QListWidget;
45 class QMessageBox;
46 class QScrollArea;
47 class QLabel;
48
49
53 class cParametersWidget :  public QWidget
54 {
55     Q_OBJECT
56
57 public:
58
59
65     cParametersWidget(cGlobalPreferences* globalpreferences, QWidget* parent = (QWidget *)0);
66
67
71     ~cParametersWidget();
72
73
78     cParameters& getParameters();
79
80
85     void setAndRestoreParameters(cParameters& parameters);
86
87
92     void closeEvent(QCloseEvent *event);
93
94
100     void setSequence(int peptidetypeindex, QString sequence);
101
102
108     void setTag(int peptidetypeindex, QString tag);
109
110
115     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
116
117
118 private:
119
120     bool hideunusedparameters;
121
122     QWidget* parent;
123
124     cGlobalPreferences* globalpreferences;
125     cParameters parameters;
126
127     QVBoxLayout* vlayout1;
128     QVBoxLayout* vlayout2;
129     QVBoxLayout* vlayout;
130     QHBoxLayout* hlayout;
131     QWidget* hlayoutwidget;
132     QScrollArea* hlayoutscroll;
133
134     QString settingsfile;
135     QString oldsettingsfile;
136
137     QDialogButtonBox* stdbuttons;
138     QHBoxLayout* buttons;
139     QPushButton* load;
140     QPushButton* save;
141     QPushButton* saveas;
142
143     QGroupBox* searchgroupbox;
144     QGridLayout* searchgridlayout;
145     QLabel* modelabel;
146     QComboBox* mode;
147     eModeType oldmodetype;
148     QSpinBox* maximumnumberofthreads;
149     QLabel* maximumnumberofthreadslabel;
150
151     QGroupBox* experimentalspectragroupbox;
152     QGridLayout* experimentalspectragridlayout;
153     QLabel* peptidetypelabel;
```

```
154     QComboBox* peptidetype;
155     QLabel* peaklistlabel;
156     QLineEdit* peaklistline;
157     QCheckBox* useprofiledata;
158     QPushButton* peaklistbutton;
159     QHBoxLayout* peaklistlayout;
160     QWidget* peaklistwidget;
161     QLabel* scannumberlabel;
162     QSpinBox* scannumber;
163     QLabel* precursormasslabel;
164     QDoubleSpinBox* precursormass;
165     QLabel* precursoradductlabel;
166     QLineEdit* precursoradduct;
167     QLabel* precursorchargelabel;
168     QSpinBox* precursorcharge;
169     QLabel* precursormasserrortolerancelabel;
170     QDoubleSpinBox* precursormasserrortolerance;
171     QLabel* fragmentmasserrortolerancelabel;
172     QDoubleSpinBox* fragmentmasserrortolerance;
173     QLabel* minimumrelativeintensitythresholdlabel;
174     QDoubleSpinBox* minimumrelativeintensitythreshold;
175     QLabel* minimumabsoluteintensitythresholdlabel;
176     QSpinBox* minimumabsoluteintensitythreshold;
177     QLabel* mzratiolabel;
178     QDoubleSpinBox* minimummz;
179     QDoubleSpinBox* maximummz;
180     QHBoxLayout* mzratiolayout;
181     QWidget* mzratiowidget;
182     QLabel* rtlabel;
183     QDoubleSpinBox* minimumrt;
184     QDoubleSpinBox* maximumrt;
185     QHBoxLayout* rtlayout;
186     QWidget* rtwidget;
187     QLabel* fwhmlabel;
188     QDoubleSpinBox* fwhm;
189
190     QGroupBox* isotoperatiosgroupbox;
191     QGridLayout* isotoperatiosgridlayout;
192
193     QLabel* ratio54Fe56FeLabel;
194     QCheckBox* ratio54Fe56FeEnabled;
195     QDoubleSpinBox* minratio54Fe56Fe;
196     QDoubleSpinBox* maxratio54Fe56Fe;
197     QHBoxLayout* ratio54Fe56FeLayout;
198     QWidget* ratio54Fe56FeWidget;
199
200     QLabel* ratio60Ni58NiLabel;
201     QCheckBox* ratio60Ni58NiEnabled;
202     QDoubleSpinBox* minratio60Ni58Ni;
203     QDoubleSpinBox* maxratio60Ni58Ni;
204     QHBoxLayout* ratio60Ni58NiLayout;
205     QWidget* ratio60Ni58NiWidget;
206
207     QLabel* ratio62Ni58NiLabel;
208     QCheckBox* ratio62Ni58NiEnabled;
209     QDoubleSpinBox* minratio62Ni58Ni;
210     QDoubleSpinBox* maxratio62Ni58Ni;
211     QHBoxLayout* ratio62Ni58NiLayout;
212     QWidget* ratio62Ni58NiWidget;
213
214     QLabel* ratio65Cu63CuLabel;
215     QCheckBox* ratio65Cu63CuEnabled;
216     QDoubleSpinBox* minratio65Cu63Cu;
217     QDoubleSpinBox* maxratio65Cu63Cu;
218     QHBoxLayout* ratio65Cu63CuLayout;
219     QWidget* ratio65Cu63CuWidget;
220
221     QLabel* ratio66Zn64ZnLabel;
222     QCheckBox* ratio66Zn64ZnEnabled;
223     QDoubleSpinBox* minratio66Zn64Zn;
224     QDoubleSpinBox* maxratio66Zn64Zn;
225     QHBoxLayout* ratio66Zn64ZnLayout;
226     QWidget* ratio66Zn64ZnWidget;
227
228     QLabel* ratio67Zn64ZnLabel;
229     QCheckBox* ratio67Zn64ZnEnabled;
230     QDoubleSpinBox* minratio67Zn64Zn;
231     QDoubleSpinBox* maxratio67Zn64Zn;
232     QHBoxLayout* ratio67Zn64ZnLayout;
233     QWidget* ratio67Zn64ZnWidget;
234
235     QLabel* ratio68Zn64ZnLabel;
236     QCheckBox* ratio68Zn64ZnEnabled;
237     QDoubleSpinBox* minratio68Zn64Zn;
238     QDoubleSpinBox* maxratio68Zn64Zn;
239     QHBoxLayout* ratio68Zn64ZnLayout;
240     QWidget* ratio68Zn64ZnWidget;
```

```
241
242     QGroupBox* brickdatabasegroupbox;
243     QGridLayout* brickdatabasegridlayout;
244     QLabel* brickdatabaselabel;
245     QLineEdit* brickdatabaseline;
246     QPushButton* brickdatabasebutton;
247     QHBoxLayout* brickdatabaselayout;
248     QWidget* brickdatabasewidget;
249     QLabel* maximumbricksincombinatiolabel;
250     QSpinBox* maximumbricksincombinationbegin;
251     QSpinBox* maximumbricksincombinationmiddle;
252     QSpinBox* maximumbricksincombinationend;
253     QHBoxLayout* maximumbricksincombinationlayout;
254     QWidget* maximumbricksincombinationwidget;
255     QLabel* maximumcumulativemasslabel;
256     QDoubleSpinBox* maximumcumulativemass;
257     QLabel* modificationslabel;
258     QLineEdit* modificationsline;
259     QPushButton* modificationsbutton;
260     QHBoxLayout* modificationslayout;
261     QWidget* modificationswidget;
262
263     QGroupBox* miscgroupbox;
264     QGridLayout* miscgridlayout;
265     QLabel* blindedgeslabel;
266     QComboBox* blindedges;
267     QLabel* cyclicnterminuslabel;
268     QCheckBox* cyclicnterminus;
269     QLabel* cyclicctterminuslabel;
270     QCheckBox* cyclicctterminus;
271     QLabel* internalfragmentslabel;
272     QCheckBox* internalfragments;
273     QLabel* enablescramblinglabel;
274     QCheckBox* enablescrambling;
275     QLabel* similaritysearchlabel;
276     QCheckBox* similaritysearch;
277     QLabel* regularblocksorderlabel;
278     QCheckBox* regularblocksorder;
279
280     QGroupBox* theoreticalspectragroupbox;
281     QGridLayout* theoreticalspectragridlayout;
282     QLabel* sequencedatabaselabel;
283     QLineEdit* sequencedatabaseline;
284     QPushButton* sequencedatabasebutton;
285     QHBoxLayout* sequencedatabaselayout;
286     QWidget* sequencedatabasewidget;
287     QLabel* scoretypelabel;
288     QComboBox* scoretype;
289     QLabel* hitsreportedlabel;
290     QSpinBox* hitsreported;
291     QLabel* sequencetaglabel;
292     QLineEdit* sequencetag;
293     QLabel* iontypeslabel;
294     cFragmentIonsListWidget* iontypes;
295     QLabel* neutrallosstypeslabel;
296     cNeutralLossesListWidget* neutrallosstypes;
297     QLabel* maximumcombinedlosseslabel;
298     QSpinBox* maximumcombinedlosses;
299     QLabel* elementstypeslabel;
300     cNeutralLossesListWidget* elementstypes;
301     QLabel* maximumcombinedelementslabel;
302     QSpinBox* maximumcombinedelements;
303     //QCheckBox* clearhitswithoutparent;
304     QLabel* reportunmatchedtheoreticalpeakslabel;
305     QCheckBox* reportunmatchedtheoreticalpeaks;
306     QLabel* generateisotopepatternlabel;
307     QCheckBox* generateisotopepattern;
308     QLabel* minimumpatternsizelabel;
309     QSpinBox* minimumpatternsize;
310     QLabel* minimumfeaturesizelabel;
311     QSpinBox* minimumfeaturesize;
312     QLabel* minimumiontypeslabel;
313     QSpinBox* minimumiontypes;
314     QLabel* basicformulachecklabel;
315     QCheckBox* basicformulacheck;
316     QLabel* advancedformulachecklabel;
317     QCheckBox* advancedformulacheck;
318     QLabel* noratiochecklabel;
319     QCheckBox* noratiocheck;
320     QLabel* calculatefdrslabel;
321     QCheckBox* calculatefdrs;
322
323     QLabel* minimumannotationintensitylabel;
324     QDoubleSpinBox* minimumannotationintensityrelative;
325     QSpinBox* minimumannotationintensityabsolute;
326     QHBoxLayout* minimumannotationintensitylayout;
327     QWidget* minimumannotationintensitywidget;
```

```
328
329     QLabel* mzdifftolerancelabel;
330     QDoubleSpinBox* mzdifftolerance;
331     QLabel* intensitytolerancelabel;
332     QDoubleSpinBox* intensitytolerance;
333
334     QGroupBox* searchedsequencegroupbox;
335     QGridLayout* searchedsequencegridlayout;
336     QLabel* searchedsequencelabel;
337     QLineEdit* searchedsequenceline;
338     QPushButton* searchedsequencebutton;
339     QHBoxLayout* searchedsequencelayout;
340     QWidget* searchedsequencewidget;
341     QLabel* searchedsequencemodiflabel;
342     QLineEdit* searchedsequenceNtermmodif;
343     QLineEdit* searchedsequenceCtermmodif;
344     QLineEdit* searchedsequenceTmodif;
345     QHBoxLayout* searchedsequencemodiflayout;
346     QWidget* searchedsequencemodifwidget;
347     QLabel* searchedsequenceformulalabel;
348     QLineEdit* searchedsequenceformula;
349
350     QString lastdirloadsettings;
351     QString lastdirsavesettings;
352
353     QString defaultdirselectpeaklist;
354     QString defaultdirselectbricksdatabase;
355     QString defaultdirselectmodifications;
356     QString defaultdirselectsequencedatabase;
357
358
359 protected:
360
361
366     void keyPressEvent(QKeyEvent *event);
367
368
369 private slots:
370
371     void loadSettings();
372
373     void saveSettings();
374
375     void saveSettingsAs();
376
377     void peaklistButtonReleased();
378
379     void brickDatabaseButtonReleased();
380
381     void modificationsButtonReleased();
382
383     void sequenceDatabaseButtonReleased();
384
385     bool updateParameters();
386
387     void updateParametersAndHide();
388
389     void restoreParameters();
390
391     void updateSettingsWhenPeptideTypeChanged(int index);
392
393     void updateSettingsWhenModeChanged(int index);
394
395     void resetFragmentIonTypes();
396
397     void drawPeptideButtonReleased();
398
399
400 signals:
401
407     void sendSequenceLine(int peptidetypeindex, QString sequence);
408
409
413     void setHCON();
414
415
419     void clearIonSelection();
420
421 };
422
423 #endif
```

17.121 gui/cPatternSimulatorSceneWidget.h File Reference

The implementation of an isotopic pattern simulator (graphics).

```
#include <iostream>
#include <QWidget>
#include <QGraphicsView>
#include "core/cTheoreticalSpectrum.h"
```

Classes

- class cPatternSimulatorSceneWidget

  The implementation of an isotopic pattern simulator (graphics).

Functions

- double computeGaussFunction (double x, double a, double ny, double sigma)

  Compute value of Gaussian function.
- double computeGaussHeight (double x, double y, double ny, double sigma)

  Compute height of Gaussian function (a).

17.121.1 Detailed Description

The implementation of an isotopic pattern simulator (graphics).

17.121.2 Function Documentation

17.121.2.1 computeGaussFunction()

```
double computeGaussFunction (
            double x,
            double a,
            double ny,
            double sigma )
```

Compute value of Gaussian function.

Parameters

x	x
a	a
ny	ny
sigma	sigma

Return values

double	f(x)

17.121.2.2 computeGaussHeight()

```
double computeGaussHeight (
            double x,
            double y,
```

```
            double ny,
            double sigma )
```

Compute height of Gaussian function (a).

Parameters

x	x
y	y
ny	ny
sigma	sigma

Return values

double	a

17.122 cPatternSimulatorSceneWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CPATTERNSIMULATORSCENEWIDGET_H
8 #define _CPATTERNSIMULATORSCENEWIDGET_H
9
10 #include <iostream>
11 #include <QWidget>
12 #include <QGraphicsView>
13
14 #include "core/cTheoreticalSpectrum.h"
15
16
17 using namespace std;
18
19
20 // forward declaration
21 class QPaintEvent;
22 class QGraphicsScene;
23 class QMatrix;
24 class QGraphicsItemGroup;
25 class QGraphicsRectItem;
26 class QGraphicsSimpleTextItem;
27
28
37 double computeGaussFunction(double x, double a, double ny, double sigma);
38
39
48 double computeGaussHeight(double x, double y, double ny, double sigma);
49
50
54 class cPatternSimulatorSceneWidget :  public QGraphicsView
55 {
56     Q_OBJECT
57
58 public:
59
60
65     cPatternSimulatorSceneWidget(QWidget* parent = 0);
66
67
71     ~cPatternSimulatorSceneWidget();
72
73
79     void initialize(cPeaksList isotopicpattern, double fwhm);
80
81
87     void exportToPDF(QString filename, bool postscript);
88
89
94     void exportToSVG(QString filename);
95
96
101     void exportToPNG(QString filename);
102
103
108     void hidePeakLabels(bool state);
109
```

```
110
111 protected:
112
113
118     void wheelEvent(QWheelEvent *event);
119
120
125     void mouseMoveEvent(QMouseEvent *event);
126
127
132     void mouseReleaseEvent(QMouseEvent *event);
133
134
139     void mousePressEvent(QMouseEvent *event);
140
141
146     void showEvent(QShowEvent *event);
147
148
153     void resizeEvent(QResizeEvent *event);
154
155
156 private:
157
158     QWidget* parent;
159     QGraphicsScene* scene;
160     QMatrix originalmatrix;
161
162     QGraphicsItemGroup* zoomgroup;
163     QGraphicsRectItem* zoomrect;
164     QGraphicsSimpleTextItem* zoomsimpletextitem;
165
166     QGraphicsSimpleTextItem* cursorsimpletextitem;
167
168     cPeaksList isotopicpattern;
169
170     qreal currentscale;
171     qreal factor;
172
173     int origwidth;
174     int origheight;
175
176     bool calledbyresizeevent;
177     deque<int> oldwidth;
178     deque<int> oldheight;
179
180     bool absoluteintensity;
181     bool rawdatastate;
182     bool hidelabels;
183
184     int pressedx;
185     int pressedy;
186
187     int currentx;
188     int currenty;
189
190     double minmzratio;
191     double maxmzratio;
192
193     int topmargin;
194     int bottommargin;
195     int leftmargin;
196     int rightmargin;
197
198     bool firstshow;
199     bool enablemousemzselection;
200
201     double mzoverhead;
202     double fwhm;
203
204
205     double getMZRatioFromXPosition(int x, int w);
206
207
208     int getXPositionFromMZRatio(double mzratio, int w);
209
210
211     double getIntensityFromYPosition(int y, int h);
212
213
214     int getYPositionFromIntensity(double intensity, int h);
215
216
217     void redrawScene();
218
219
220     void updateZoomGroup();
```

```
221
222
223     void calculateMinMaxMZ();
224
225
226     double getMaximumIntensity();
227
228
229 signals:
230
231
237     void updateMZInterval(double minmz, double maxmz);
238
239
240 private slots:
241
242
243     void zoomIn();
244
245
246     void zoomOut();
247
248
249     void normalSize();
250
251
252     void absoluteIntensityStateChanged(bool state);
253
254
255     void rawDataStateChanged(bool state);
256
257
258     void setMZInterval(double minmz, double maxmz);
259
260
261     void resetMZInterval();
262
263
264     void enableMouseMzSelectionTool(bool enable);
265
266 };
267
268 #endif
```

17.123 gui/cPatternSimulatorWidget.h File Reference

The implementation of an isotopic pattern simulator (window).

```
#include <QWidget>
#include <QDesktopServices>
#include <QToolBar>
#include <QMenuBar>
#include <QMenu>
#include <QUrl>
#include <QFileInfo>
#include <QMainWindow>
#include <QTextDocument>
#include <QTableView>
#include <QStandardItemModel>
#include <QStandardItem>
#include <QHeaderView>
#include <QProgressDialog>
#include <QTextStream>
#include "core/cGlobalPreferences.h"
#include "gui/cPatternSimulatorSceneWidget.h"
#include "gui/cSpectrumDetailProxyModel.h"
#include "gui/cExportDialog.h"
#include "gui/cEventFilter.h"
#include "gui/cHTMLDelegate.h"
```

Classes

- class cPatternSimulatorWidget

  The implementation of an isotopic pattern simulator (window).

17.123.1 Detailed Description

The implementation of an isotopic pattern simulator (window).

17.124 cPatternSimulatorWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CPATTERNSIMULATORWIDGET_H
8 #define _CPATTERNSIMULATORWIDGET_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QToolBar>
13 #include <QMenuBar>
14 #include <QMenu>
15 #include <QUrl>
16 #include <QFileInfo>
17 #include <QMainWindow>
18 #include <QTextDocument>
19 #include <QTableView>
20 #include <QStandardItemModel>
21 #include <QStandardItem>
22 #include <QHeaderView>
23 #include <QProgressDialog>
24 #include <QTextStream>
25 #include "core/cGlobalPreferences.h"
26 #include "gui/cPatternSimulatorSceneWidget.h"
27 #include "gui/cSpectrumDetailProxyModel.h"
28 #include "gui/cExportDialog.h"
29 #include "gui/cEventFilter.h"
30 #include "gui/cHTMLDelegate.h"
31
32
33 // forward declaration
34 class QTextBrowser;
35 class QHBoxLayout;
36 class QVBoxLayout;
37 class QFormLayout;
38 class QCheckBox;
39 class QPushButton;
40 class QSplitter;
41 class QComboBox;
42 class QDoubleSpinBox;
43 class QSpinBox;
44 class QAction;
45 class QLabel;
46
47
51 class cPatternSimulatorWidget :  public QMainWindow
52 {
53     Q_OBJECT
54
55 public:
56
57
63     cPatternSimulatorWidget(cGlobalPreferences* globalpreferences, QWidget *parent);
64
65
69     ~cPatternSimulatorWidget();
70
71
76     void closeEvent(QCloseEvent *event);
77
78
83     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
84
85
86 protected:
87
88
93     void keyPressEvent(QKeyEvent *event);
94
95
96 private:
97
```

```
98      cGlobalPreferences* globalpreferences;
99      QWidget* parent;
100
101      QMenuBar* menuBar;
102      QMenu* menuFile;
103      QMenu* menuView;
104      QMenu* menuHelp;
105
106      QToolBar* toolbarFile;
107      QToolBar* toolbarView;
108      QToolBar* toolbarHelp;
109      QToolBar* toolbarMz;
110      QToolBar* toolbarInput;
111
112      QAction* actionExportTable;
113      QAction* actionExportSpectrum;
114      QAction* actionCloseWindow;
115      QAction* actionZoomIn;
116      QAction* actionZoomOut;
117      QAction* actionZoomReset;
118      QAction* actionAbsoluteIntensity;
119      QAction* actionRawData;
120      QAction* actionHideLabels;
121      QAction* actionMouseMzSelection;
122      QAction* actionHTMLDocumentation;
123      QSplitter* hsplitter1;
124      QSplitter* hsplitter2;
125      QSplitter* vsplitter;
126
127      QWidget* widgetmz;
128      QHBoxLayout* hboxmz;
129      QLabel* labelmz;
130      QDoubleSpinBox* minmz;
131      QLabel* labelseparator;
132      QDoubleSpinBox* maxmz;
133      QPushButton* setmzinterval;
134      QPushButton* resetmzinterval;
135
136      QWidget* widgetiontype;
137      QHBoxLayout* hboxiontype;
138      QLabel* labeliontype;
139      QComboBox* comboboxiontype;
140
141      QWidget* widgetformula;
142      QHBoxLayout* hboxformula;
143      QLabel* labelformula;
144      QLineEdit* lineeditformula;
145
146      QWidget* widgetfwhm;
147      QHBoxLayout* hboxfwhm;
148      QLabel* labelfwhm;
149      QDoubleSpinBox* doublespinboxfwhm;
150
151      QWidget* widgetcharge;
152      QHBoxLayout* hboxcharge;
153      QLabel* labelcharge;
154      QSpinBox* spinboxcharge;
155
156      QWidget* widgetabsoluteintensity;
157      QHBoxLayout* hboxabsoluteintensity;
158      QLabel* labelabsoluteintensity;
159      QDoubleSpinBox* doublespinboxabsoluteintensity;
160
161      QPushButton* simulatebutton;
162
163      QTableView* peakstable;
164      QStandardItemModel* peakstablemodel;
165      cSpectrumDetailProxyModel* proxymodel;
166
167      cPatternSimulatorSceneWidget* spectrumscene;
168
169      cExportDialog* exportdialog;
170
171
172      void preparePeaksTable(cPeaksList& isotopepattern);
173
174
175      void deleteTable();
176
177
178  signals:
179
180
186      void emitMZInterval(double minmz, double maxmz);
187
188
193      void rawDataStateChangedSignal(bool state);
```

```
194
195
196 private slots:
197
198
199     void simulateButtonReleased();
200
201
202     void updateMZInterval(double minmz, double maxmz);
203
204
205     void setMZInterval();
206
207
208     void exportSpectrum();
209
210
211     void exportTableToCSV();
212
213
214     void rawDataStateChanged(bool state);
215
216
217     void hidePeakLabels(bool hide);
218
219
220     void showHTMLDocumentation();
221
222
223     void closeWindow();
224
225 };
226
227 #endif
228
```

17.125 gui/cPreferencesDialog.h File Reference

The implementation of preferences dialog.

```
#include "core/cGlobalPreferences.h"
#include <QDialog>
#include <QDesktopServices>
#include <QUrl>
#include <QKeyEvent>
#include <QDialogButtonBox>
#include <QTabWidget>
#include <QVBoxLayout>
```

Classes

- class cPreferencesDialog

  Preferences dialog.

17.125.1 Detailed Description

The implementation of preferences dialog.

17.126 cPreferencesDialog.h

Go to the documentation of this file.

```
1
7 #ifndef _CPREFERENCESDIALOG_H
8 #define _CPREFERENCESDIALOG_H
9
10 #include "core/cGlobalPreferences.h"
11
12 #include <QDialog>
13 #include <QDesktopServices>
14 #include <QUrl>
15 #include <QKeyEvent>
```

```
16 #include <QDialogButtonBox>
17 #include <QTabWidget>
18 #include <QVBoxLayout>
19
20
21 class cPreferencesTabIonTypes;
22 class cPreferencesTabDataProcessing;
23 class cPreferencesTabDirectories;
24 class cPreferencesTabURLs;
25
26
30 class cPreferencesDialog :  public QDialog
31 {
32     Q_OBJECT
33
34 public:
35
36
42     explicit cPreferencesDialog(cGlobalPreferences& globalpreferences, QWidget *parent = 0);
43
44
49     void setPreferences(cGlobalPreferences& globalpreferences);
50
51
56     void resetPreferences(cGlobalPreferences& globalpreferences);
57
58
62     ~cPreferencesDialog();
63
64
65 protected:
66
67
72     void keyPressEvent(QKeyEvent *event);
73
74
75 private:
76
77     cPreferencesTabIonTypes* tabiontypes;
78     cPreferencesTabDataProcessing* tabdataprocessing;
79     cPreferencesTabDirectories* tabdirectories;
80     cPreferencesTabURLs* taburls;
81
82     QTabWidget *tabwidget;
83     QDialogButtonBox *buttonbox;
84
85     QVBoxLayout *mainlayout;
86
87 };
88
89
90 #endif
91
```

17.127 gui/cPreferencesIonTypeDialog.h File Reference

The implementation of ion type dialog.

```
#include "core/cSummaryFormula.h"
#include "core/cGlobalPreferences.h"
#include <QDialog>
#include <QDialogButtonBox>
#include <QGridLayout>
#include <QVBoxLayout>
#include <QLabel>
#include <QLineEdit>
#include <QComboBox>
#include <QSpinBox>
#include <QString>
```

Classes

- class cPreferencesIonTypeDialog

  Preferences dialog.

17.127.1 Detailed Description

The implementation of ion type dialog.

17.128 cPreferencesIonTypeDialog.h

Go to the documentation of this file.

```
1
7 #ifndef _CPREFERENCESIONTYPEDIALOG_H
8 #define _CPREFERENCESIONTYPEDIALOG_H
9
10 #include "core/cSummaryFormula.h"
11 #include "core/cGlobalPreferences.h"
12
13 #include <QDialog>
14 #include <QDialogButtonBox>
15 #include <QGridLayout>
16 #include <QVBoxLayout>
17 #include <QLabel>
18 #include <QLineEdit>
19 #include <QComboBox>
20 #include <QSpinBox>
21 #include <QString>
22
23
27 class cPreferencesIonTypeDialog :  public QDialog
28 {
29     Q_OBJECT
30
31 public:
32
33
38     explicit cPreferencesIonTypeDialog(QWidget *parent = 0);
39
40
45     void getValues(cIonType& ion);
46
47
52     void setValues(cIonType& ion);
53
54
58     ~cPreferencesIonTypeDialog();
59
60
61 private:
62
63     QVBoxLayout *mainlayout;
64
65     QDialogButtonBox *buttonbox;
66
67     QWidget* gridlayoutwidget;
68     QGridLayout* gridlayout;
69
70     QLabel* labelname;
71     QLineEdit* lineeditname;
72
73     QLabel* labelformula;
74     QLineEdit* lineeditformula;
75
76     QLabel* labelmode;
77     QComboBox* comboboxmode;
78
79     QLabel* labelmultiplier;
80     QSpinBox* spinboxmultiplier;
81
82 };
83
84
85 #endif
86
```

17.129 gui/cPreferencesTabDataProcessing.h File Reference

The implementation of a tab used to define the data processing parameters in preferences dialog.

```
#include "core/cGlobalPreferences.h"
#include <QWidget>
#include <QHBoxLayout>
#include <QVBoxLayout>
```

```
#include <QLabel>
#include <QPushButton>
#include <QComboBox>
#include <QMessageBox>
#include <QFileInfo>
```

Classes

- class cPreferencesTabDataProcessing

  The class representing a tab used to define the data processing parameters.

17.129.1 Detailed Description

The implementation of a tab used to define the data processing parameters in preferences dialog.

17.130 cPreferencesTabDataProcessing.h

Go to the documentation of this file.

```
1
7 #ifndef _CPREFERENCESTABDATAPROCESSING_H
8 #define _CPREFERENCESTABDATAPROCESSING_H
9
10 #include "core/cGlobalPreferences.h"
11
12 #include <QWidget>
13 #include <QHBoxLayout>
14 #include <QVBoxLayout>
15 #include <QLabel>
16 #include <QPushButton>
17 #include <QComboBox>
18 #include <QMessageBox>
19 #include <QFileInfo>
20
21
25 class cPreferencesTabDataProcessing :  public QWidget
26 {
27     Q_OBJECT
28
29 public:
30
31
37     explicit cPreferencesTabDataProcessing(cGlobalPreferences& globalpreferences, QWidget *parent = 0);
38
39
44     void setPreferences(cGlobalPreferences& globalpreferences);
45
46
51     void resetPreferences(cGlobalPreferences& globalpreferences);
52
53
57     ~cPreferencesTabDataProcessing();
58
59
60 private:
61
62     QVBoxLayout *mainlayout;
63
64     QWidget* gridlayoutwidget;
65     QGridLayout* gridlayout;
66
67     QLabel* labellinebafprocessingmethod;
68     QComboBox* comboboxlinebafprocessingmethod;
69
70     QLabel* labelprofilebafprocessingmethod;
71     QComboBox* comboboxprofilebafprocessingmethod;
72
73     QLabel* labelrawdataprocessingmethod;
74     QComboBox* comboboxrawdataprocessingmethod;
75
76     QWidget* hwidget;
77     QHBoxLayout* hlayout;
78     QPushButton* defaultbutton;
79
80
81 private slots:
82
```

```
83
84      void defaultButtonReleased();
85
86 };
87
88
89 #endif
90
```

17.131 gui/cPreferencesTabDirectories.h File Reference

The implementation of a tab used to define default directories in preferences dialog.

```
#include "core/cGlobalPreferences.h"
#include <QWidget>
#include <QGridLayout>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>
#include <QFileDialog>
```

Classes

- class cPreferencesTabDirectories

  The class representing a tab of default directories.

17.131.1 Detailed Description

The implementation of a tab used to define default directories in preferences dialog.

17.132 cPreferencesTabDirectories.h

Go to the documentation of this file.

```
1
7 #ifndef _CPREFERENCESTABDIRECTORIES_H
8 #define _CPREFERENCESTABDIRECTORIES_H
9
10 #include "core/cGlobalPreferences.h"
11
12 #include <QWidget>
13 #include <QGridLayout>
14 #include <QHBoxLayout>
15 #include <QVBoxLayout>
16 #include <QLabel>
17 #include <QLineEdit>
18 #include <QPushButton>
19 #include <QFileDialog>
20
21
25 class cPreferencesTabDirectories :  public QWidget
26 {
27     Q_OBJECT
28
29 public:
30
31
37     explicit cPreferencesTabDirectories(cGlobalPreferences& globalpreferences, QWidget *parent = 0);
38
39
44     void setPreferences(cGlobalPreferences& globalpreferences);
45
46
51     void resetPreferences(cGlobalPreferences& globalpreferences);
52
53
57     ~cPreferencesTabDirectories();
58
59
60 private:
```

```
61
62      QVBoxLayout *mainlayout;
63
64      QWidget* gridlayoutwidget;
65      QGridLayout* gridlayout;
66
67      QLabel* labelsettingsdefaultdir;
68      QLineEdit* lineeditsettingsdefaultdir;
69      QPushButton* buttonsettingsdefaultdir;
70
71      QLabel* labelpeaklistsdefaultdir;
72      QLineEdit* lineeditpeaklistsdefaultdir;
73      QPushButton* buttonpeaklistsdefaultdir;
74
75      QLabel* labelblocksdefaultdir;
76      QLineEdit* lineeditblocksdefaultdir;
77      QPushButton* buttonblocksdefaultdir;
78
79      QLabel* labelsequencesdefaultdir;
80      QLineEdit* lineeditsequencesdefaultdir;
81      QPushButton* buttonsequencesdefaultdir;
82
83      QLabel* labelmodificationsdefaultdir;
84      QLineEdit* lineeditmodificationsdefaultdir;
85      QPushButton* buttonmodificationsdefaultdir;
86
87      QLabel* labelopenopticalimagedefaultdir;
88      QLineEdit* lineeditopenopticalimagedefaultdir;
89      QPushButton* buttonopenopticalimagedefaultdir;
90
91      QLabel* labelopenhistologyimagedefaultdir;
92      QLineEdit* lineeditopenhistologyimagedefaultdir;
93      QPushButton* buttonopenhistologyimagedefaultdir;
94
95      QLabel* labelopenmicroscopyimagedefaultdir;
96      QLineEdit* lineeditopenmicroscopyimagedefaultdir;
97      QPushButton* buttonopenmicroscopyimagedefaultdir;
98
99      QLabel* labelresultsdefaultdir;
100      QLineEdit* lineeditresultsdefaultdir;
101      QPushButton* buttonresultsdefaultdir;
102
103      QLabel* labelexportcsvdefaultdir;
104      QLineEdit* lineeditexportcsvdefaultdir;
105      QPushButton* buttonexportcsvdefaultdir;
106
107      QLabel* labelexporthtmldefaultdir;
108      QLineEdit* lineeditexporthtmldefaultdir;
109      QPushButton* buttonexporthtmldefaultdir;
110
111      QLabel* labelexportimagedefaultdir;
112      QLineEdit* lineeditexportimagedefaultdir;
113      QPushButton* buttonexportimagedefaultdir;
114
115      QWidget* hwidget;
116      QHBoxLayout* hlayout;
117      QPushButton* defaultbutton;
118
119
120  private slots:
121
122
123      void settingsButtonReleased();
124
125
126      void peaklistsButtonReleased();
127
128
129      void blocksButtonReleased();
130
131
132      void sequencesButtonReleased();
133
134
135      void modificationsButtonReleased();
136
137
138      void openOpticalImageButtonReleased();
139
140
141      void openHistologyImageButtonReleased();
142
143
144      void openMicroscopyImageButtonReleased();
145
146
147      void resultsButtonReleased();
```

```
148
149
150     void exportCSVButtonReleased();
151
152
153     void exportHTMLButtonReleased();
154
155
156     void exportImageButtonReleased();
157
158
159     void defaultButtonReleased();
160
161 };
162
163
164 #endif
165
```

17.133 gui/cPreferencesTabIonTypes.h File Reference

The implementation of a tab used to define ion types in preferences dialog.

```
#include "core/cGlobalPreferences.h"
#include "gui/cPreferencesIonTypeDialog.h"
#include <QWidget>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QListWidget>
#include <QPushButton>
#include <QMessageBox>
```

Classes

- class cPreferencesTabIonTypes

  The class representing a tab of ion types.

17.133.1 Detailed Description

The implementation of a tab used to define ion types in preferences dialog.

17.134 cPreferencesTabIonTypes.h

Go to the documentation of this file.

```
1
7 #ifndef _CPREFERENCESTABIONTYPES_H
8 #define _CPREFERENCESTABIONTYPES_H
9
10 #include "core/cGlobalPreferences.h"
11 #include "gui/cPreferencesIonTypeDialog.h"
12
13 #include <QWidget>
14 #include <QHBoxLayout>
15 #include <QVBoxLayout>
16 #include <QListWidget>
17 #include <QPushButton>
18 #include <QMessageBox>
19
20
24 class cPreferencesTabIonTypes :  public QWidget
25 {
26     Q_OBJECT
27
28 public:
29
30
36     explicit cPreferencesTabIonTypes(cGlobalPreferences& globalpreferences, QWidget *parent = 0);
37
38
43     void setPreferences(cGlobalPreferences& globalpreferences);
44
```

```
45
50      void resetPreferences(cGlobalPreferences& globalpreferences);
51
52
56      ~cPreferencesTabIonTypes();
57
58
59  private:
60
61      vector<cIonType> localcustomions;
62
63      cPreferencesIonTypeDialog* iontypedialog;
64
65      QListWidget* list;
66
67      QHBoxLayout *hlayout;
68      QWidget* hwidget;
69
70      QVBoxLayout *vlayout;
71
72      QPushButton* addbutton;
73      QPushButton* editbutton;
74      QPushButton* removebutton;
75      QPushButton* defaultbutton;
76
77
78      bool isValid(cIonType& ion, int row, string& errormessage);
79
80
81  private slots:
82
83      void listItemDoubleClicked(QListWidgetItem* item);
84
85
86      void addButtonReleased();
87
88
89      void editButtonReleased();
90
91
92      void removeButtonReleased();
93
94
95      void defaultButtonReleased();
96
97  };
98
99
100 #endif
101
```

17.135 gui/cPreferencesTabURLs.h File Reference

The implementation of a tab used to define URLs in preferences dialog.

```
#include "core/cGlobalPreferences.h"
#include <QWidget>
#include <QGridLayout>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>
#include <QCheckBox>
```

Classes

- class cPreferencesTabURLs

  The class representing a tab of URLs.

17.135.1 Detailed Description

The implementation of a tab used to define URLs in preferences dialog.

17.136 cPreferencesTabURLs.h

Go to the documentation of this file.

```
1
7 #ifndef _CPREFERENCESTABURLS_H
8 #define _CPREFERENCESTABURLS_H
9
10 #include "core/cGlobalPreferences.h"
11
12 #include <QWidget>
13 #include <QGridLayout>
14 #include <QHBoxLayout>
15 #include <QVBoxLayout>
16 #include <QLabel>
17 #include <QLineEdit>
18 #include <QPushButton>
19 #include <QCheckBox>
20
21
25 class cPreferencesTabURLs :  public QWidget
26 {
27     Q_OBJECT
28
29 public:
30
31
37     explicit cPreferencesTabURLs(cGlobalPreferences& globalpreferences, QWidget *parent = 0);
38
39
44     void setPreferences(cGlobalPreferences& globalpreferences);
45
46
51     void resetPreferences(cGlobalPreferences& globalpreferences);
52
53
57     ~cPreferencesTabURLs();
58
59
60 private:
61
62     QVBoxLayout *mainlayout;
63
64     QWidget* gridlayoutwidget;
65     QGridLayout* gridlayout;
66
67     QLabel* labelurl1;
68     QLineEdit* lineeditname1;
69     QLineEdit* lineediturl1;
70
71     QLabel* labelurl2;
72     QLineEdit* lineeditname2;
73     QLineEdit* lineediturl2;
74
75     QLabel* labelurl3;
76     QLineEdit* lineeditname3;
77     QLineEdit* lineediturl3;
78
79     QLabel* labelurl4;
80     QLineEdit* lineeditname4;
81     QLineEdit* lineediturl4;
82
83     QLabel* labelurl5;
84     QLineEdit* lineeditname5;
85     QLineEdit* lineediturl5;
86
87     QLabel* labelurl6;
88     QLineEdit* lineeditname6;
89     QLineEdit* lineediturl6;
90
91     QLabel* labelurl7;
92     QLineEdit* lineeditname7;
93     QLineEdit* lineediturl7;
94
95     QLabel* labelurl8;
96     QLineEdit* lineeditname8;
97     QLineEdit* lineediturl8;
98
99     QLabel* labelurl9;
100     QLineEdit* lineeditname9;
101     QLineEdit* lineediturl9;
102
103     QLabel* labelurl10;
104     QLineEdit* lineeditname10;
105     QLineEdit* lineediturl10;
106
107     QLabel* labelviewbookmarks;
```

```
108     QCheckBox* checkboxviewbookmarks;
109
110     QWidget* hwidget;
111     QHBoxLayout* hlayout;
112     QPushButton* defaultbutton;
113
114
115 private slots:
116
117     void defaultButtonReleased();
118
119 };
120
121
122 #endif
123
```

17.137 gui/cPubChemSearchWidget.h File Reference

The implementation of the dialog PubChem search widget.

```
#include <QWidget>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include <QNetworkAccessManager>
#include <QNetworkRequest>
#include <QNetworkReply>
#include <QtConcurrent>
#include <QApplication>
#include <QThreadPool>
#include "core/utilities.h"
```

Classes

- class cPubChemSearchWidget

  The widget representing the PubChem search widget.
- struct cPubchemCIDReader

  The functor used to get the CIDs of compounds with the same formula.
- struct cPubchemNameReader

  The functor used to get the names of compounds from CID.
- struct cRowItemInfo

  An auxiliary row info structure.

17.137.1 Detailed Description

The implementation of the dialog PubChem search widget.

17.138 cPubChemSearchWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CPUBCHEMSEARCHWIDGET_H
8 #define _CPUBCHEMSEARCHWIDGET_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QUrl>
13 #include <QFileInfo>
14 #include <QNetworkAccessManager>
15 #include <QNetworkRequest>
16 #include <QNetworkReply>
17 #include <QtConcurrent>
18 #include <QApplication>
19 #include <QThreadPool>
20
```

```
21 #include "core/utilities.h"
22
23
24 // forward declaration
25 class QTextBrowser;
26 class QVBoxLayout;
27
28
32 class cPubChemSearchWidget :  public QWidget
33 {
34     Q_OBJECT
35
36 public:
37
38
42     cPubChemSearchWidget();
43
44
48     ~cPubChemSearchWidget();
49
50
55     void setHTML(string html);
56
57
62     string getHTML();
63
64
69     void closeEvent(QCloseEvent *event);
70
71
76     void store(ofstream& os);
77
78
83     void load(ifstream& is);
84
85
86 private:
87
88     QVBoxLayout* layout;
89     QTextBrowser* textbrowser;
90     string htmlstring;
91
92
93 protected:
94
95
100     void keyPressEvent(QKeyEvent *event);
101
102
103 };
104
105
109 struct cPubchemCIDReader {
110
114     typedef string result_type;
115
116
122     string getCIDList(const string& compound) {
123         string url = "https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/fastformula/";
124         url += compound;
125         url += "/cids/txt";
126
127         QNetworkAccessManager networkmanager;
128         QNetworkRequest request(QUrl(url.c_str()));
129         QNetworkReply *response = networkmanager.get(request);
130
131         QEventLoop loop;
132         QObject::connect(response, SIGNAL(finished()), &loop, SLOT(quit()));
133         loop.exec();
134
135         string cids = response->readAll().toStdString();
136
137         stringstream errorstream;
138         if (cids.empty()) {
139             errorstream « "@ERROR" « endl;
140             errorstream « "URL: " « url « endl;
141             errorstream « "ERROR: " « response->errorString().toStdString() « endl;
142             errorstream « endl;
143         }
144
145         response->deleteLater();
146
147         if (cids.empty()) {
148             return errorstream.str();
149         }
150
151         return cids;
```

```
152     }
153
154
160     void getCommaSeparatedCIDs(string& cids, string& reducedcids) {
161         reducedcids.clear();
162         if (cids.find("Status:") == string::npos) {
163             istringstream inputstream(cids);
164             string line;
165             while (getline(inputstream, line)) {
166                 if (!reducedcids.empty()) {
167                     reducedcids += ",";
168                 }
169                 reducedcids += line;
170             }
171         }
172     }
173
174
180     void attachVectorOfCIDs(string& cids, vector<string>& cidsvector) {
181         if (cids.find("Status:") == string::npos) {
182             istringstream inputstream(cids);
183             string line;
184             while (getline(inputstream, line)) {
185                 cidsvector.push_back(line);
186             }
187         }
188     }
189
190
196     int getNumberOfCIDs(string& cids) {
197         int count = 0;
198         if (cids.find("Status:") == string::npos) {
199             istringstream inputstream(cids);
200             string line;
201             while (getline(inputstream, line)) {
202                 count++;
203             }
204         }
205         else {
206             if (cids.find("Status:  404") == string::npos) {
207                 // report error codes 500 (server error), 503 (server busy), etc.
208                 cout « cids « endl « endl;
209             }
210         }
211
212         return count;
213     }
214
215
221     string operator()(const string& compound) {
222         return getCIDList(compound);
223     }
224 };
225
226
230 struct cPubchemNameReader {
231
235     typedef string result_type;
236
237
243     string getNameList(const string& cid) {
244         bool usepost = true;
245
246         string url = "https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/cid/";
247         if (!usepost) {
248             url += cid + "/";
249         }
250         url += "synonyms/txt";
251
252         if (!usepost) {
253             // URL too long
254             if (url.size() > 4000) {
255                 return "Status:  414";
256             }
257         }
258
259         QNetworkAccessManager networkmanager;
260         QNetworkRequest request(QUrl(url.c_str()));
261         QNetworkReply *response;
262
263         if (usepost) {
264             QString body = "cid=";
265             body += cid.c_str();
266             request.setHeader(QNetworkRequest::ContentTypeHeader, "application/x-www-form-urlencoded");
267             response = networkmanager.post(request, body.toUtf8());
268         }
269         else {
```

```
270             response = networkmanager.get(request);
271         }
272 
273         QEventLoop loop;
274         QObject::connect(response, SIGNAL(finished()), &loop, SLOT(quit()));
275         loop.exec();
276 
277         string names = response->readAll().toStdString();
278 
279         stringstream errorstream;
280         if (names.empty()) {
281             errorstream « "@ERROR" « endl;
282             errorstream « "URL: " « url « endl;
283             errorstream « "ERROR: " « response->errorString().toStdString() « endl;
284             errorstream « endl;
285         }
286 
287         response->deleteLater();
288 
289         if (names.empty()) {
290             return errorstream.str();
291         }
292 
293         return names;
294     }
295 
296 
302     void getReducedNames(string& names, set<string>& reducednames) {
303         if (names.find("Status:") == string::npos) {
304             istringstream inputstream(names);
305             string line;
306             while (getline(inputstream, line)) {
307                 if (line.find("SCHEMBL") == string::npos) {
308                     reducednames.insert(line);
309                 }
310             }
311             if (reducednames.empty()) {
312                 reducednames.insert("This item has only SCHEMBL name(s).");
313             }
314         }
315         else {
316             if (names.find("Status:  404") == string::npos) {
317                 // report error codes 500 (server error), 503 (server busy), etc.
318                 cout « names « endl « endl;
319             }
320         }
321     }
322 
323 
329     string operator()(const string& cid) {
330         if (cid.empty()) {
331             return "";
332         }
333         return getNameList(cid);
334     }
335 };
336 
337 
341 struct cRowItemInfo {
342 
346     int id;
347 
348 
352     double theoreticalmz;
353 
354 
358     double experimentalint;
359 
360 
364     string iontype;
365 
366 };
367 
368 #endif
369 
```

17.139 gui/cSequenceDatabaseProxyModel.h File Reference

A proxy model for editor of database of sequences.

```
#include <QString>
#include <QVariant>
#include <QComboBox>
```

```
#include <QSortFilterProxyModel>
#include <QLineEdit>
#include "core/utilities.h"
```

Classes

- class cSequenceDatabaseProxyModel

  A proxy model for editor of database of sequences.

17.139.1 Detailed Description

A proxy model for editor of database of sequences.

17.140 cSequenceDatabaseProxyModel.h

Go to the documentation of this file.

```
1
7 #ifndef _CSEQUENCEDATABASEPROXYMODEL_H
8 #define _CSEQUENCEDATABASEPROXYMODEL_H
9
10 #include <QString>
11 #include <QVariant>
12 #include <QComboBox>
13 #include <QSortFilterProxyModel>
14 #include <QLineEdit>
15 #include "core/utilities.h"
16
17
21 class cSequenceDatabaseProxyModel :  public QSortFilterProxyModel {
22
23     Q_OBJECT
24
25 private:
26
27     QComboBox* filteroperator;
28     QComboBox* filtercombobox1;
29     QComboBox* filtercomparatorcombobox1;
30     QLineEdit* filterline1;
31     QComboBox* filtercombobox2;
32     QComboBox* filtercomparatorcombobox2;
33     QLineEdit* filterline2;
34     bool wholeword;
35
36 public:
37
38
43     cSequenceDatabaseProxyModel(QObject *parent = 0);
44
45
56     void initialize(QComboBox* rowsfilteroperator, QComboBox* rowsfiltercombobox1, QComboBox*
      rowsfiltercomparatorcombobox1, QLineEdit* rowsfilterline1, QComboBox* rowsfiltercombobox2, QComboBox*
      rowsfiltercomparatorcombobox2, QLineEdit* rowsfilterline2);
57
58
63     void setWholeWord(bool wholeword);
64
65
66 protected:
67
68
75     bool filterAcceptsRow(int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE;
76
77
84     bool lessThan(const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE;
85
86
87 };
88
89
90 #endif
91
```

17.141 gui/cSequenceDatabaseWidget.h File Reference

An editor of sequences database.

```
#include <QWidget>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include <QMainWindow>
#include <QToolBar>
#include <QAction>
#include <QTableView>
#include <QStandardItemModel>
#include <QStandardItem>
#include <QComboBox>
#include <fstream>
#include "core/utilities.h"
#include "core/cGlobalPreferences.h"
#include "core/cSequenceDatabase.h"
#include "gui/cSequenceDatabaseProxyModel.h"
#include "gui/cViewButtonDelegate.h"
#include "gui/cCheckBoxDelegate.h"
#include "gui/cComboBoxDelegate.h"
```

Classes

- class cSequenceDatabaseWidget

  An editor of sequences database.

17.141.1 Detailed Description

An editor of sequences database.

17.142 cSequenceDatabaseWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CSEQUENCEDATABASEWIDGET_H
8 #define _CSEQUENCEDATABASEWIDGET_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QUrl>
13 #include <QFileInfo>
14 #include <QMainWindow>
15 #include <QToolBar>
16 #include <QAction>
17 #include <QTableView>
18 #include <QStandardItemModel>
19 #include <QStandardItem>
20 #include <QComboBox>
21 #include <fstream>
22 #include "core/utilities.h"
23 #include "core/cGlobalPreferences.h"
24 #include "core/cSequenceDatabase.h"
25 #include "gui/cSequenceDatabaseProxyModel.h"
26 #include "gui/cViewButtonDelegate.h"
27 #include "gui/cCheckBoxDelegate.h"
28 #include "gui/cComboBoxDelegate.h"
29
30
31 // forward declaration
32 class QHBoxLayout;
33 class QVBoxLayout;
34 class QPushButton;
35 class QLineEdit;
36 class QMenuBar;
37 class QMenu;
```

```
38
39
43 class cSequenceDatabaseWidget :  public QMainWindow
44 {
45     Q_OBJECT
46
47 public:
48
49
55     cSequenceDatabaseWidget(cGlobalPreferences* globalpreferences, QWidget* parent = (QWidget *)0);
56
57
61     ~cSequenceDatabaseWidget();
62
63
68     void closeEvent(QCloseEvent *event);
69
70
76     void importSequence(int peptidetypeindex, QString sequence);
77
78
83     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
84
85
86 private:
87
88     QString editorname;
89
90     QWidget* parent;
91
92     cGlobalPreferences* globalpreferences;
93
94     QMenuBar* menuBar;
95     QMenu* menuFile;
96     QMenu* menuEdit;
97     QMenu* menuHelp;
98
99     QToolBar* toolbarFile;
100     QAction* actionNewDatabase;
101     QAction* actionOpenDatabase;
102     QAction* actionSaveDatabase;
103     QAction* actionSaveDatabaseAs;
104     QAction* actionImportDatabase;
105     QAction* actionCloseWindow;
106
107     QToolBar* toolbarEdit;
108     QAction* actionAddRow;
109     QAction* actionRemoveSelectedRows;
110     QAction* actionSelectAll;
111     QAction* actionUnselectAll;
112     QAction* actionRemoveDuplicateRows;
113
114     QToolBar* toolbarHelp;
115     QAction* actionHTMLDocumentation;
116
117     QToolBar* toolbarFilter;
118     QWidget* rowsfilterwidget;
119     QHBoxLayout* rowsfilterhbox;
120     QComboBox* rowsfilteroperator;
121     QComboBox* rowsfiltercombobox1;
122     QComboBox* rowsfiltercomparatorcombobox1;
123     QLineEdit* rowsfilterline1;
124     QComboBox* rowsfiltercombobox2;
125     QComboBox* rowsfiltercomparatorcombobox2;
126     QLineEdit* rowsfilterline2;
127     QCheckBox* rowsfiltercasesensitive;
128     QCheckBox* rowsfilterwholeword;
129     QPushButton* rowsfilterbutton;
130     QPushButton* rowsfilterclearbutton;
131
132     QTableView* database;
133     QStandardItemModel* databasemodel;
134     cSequenceDatabaseProxyModel* proxymodel;
135     QVBoxLayout* mainlayout;
136     QWidget* mainwidget;
137
138     QString databasefile;
139     QString lastdir;
140     ifstream inputstream;
141     ofstream outputstream;
142     cSequenceDatabase sequences;
143
144     bool datamodified;
145
146     void deleteTable();
147
148     void resetHeader();
```

```
149
150     bool checkTable();
151
152     bool checkFormula(int row, const string& summary);
153
154     bool checkSequence(int row);
155
156     void setDataModified(bool datamodified);
157
158
159 protected:
160
161
166     void keyPressEvent(QKeyEvent *event);
167
168
169 private slots:
170
171     void closeWindow();
172
173     void openDatabase();
174
175     bool saveDatabase();
176
177     bool saveDatabaseAs();
178
179     void saveDatabaseCheck();
180
181     void addRow();
182
183     void removeSelectedRows();
184
185     void itemChanged(QStandardItem* item);
186
187     void headerItemClicked(int index);
188
189     void filterRows();
190
191     void resetFilter();
192
193     void createNewDatabase();
194
195     void importDatabase();
196
197     void selectAll();
198
199     void unselectAll();
200
201     void showHTMLDocumentation();
202
203     void editItem(const QModelIndex& index);
204
205     void removeDuplicateRows();
206
207 };
208
209 #endif
```

17.143 gui/cSpectrumDetailProxyModel.h File Reference

A table proxy model.

```
#include <QString>
#include <QVariant>
#include <QSortFilterProxyModel>
#include <QStandardItemModel>
```

Classes

- class cSpectrumDetailProxyModel

  A table proxy model.

17.143.1 Detailed Description

A table proxy model.

17.144 cSpectrumDetailProxyModel.h

Go to the documentation of this file.

```
1
7 #ifndef _CSPECTRUMDETAILPROXYMODEL_H
8 #define _CSPECTRUMDETAILPROXYMODEL_H
9
10 #include <QString>
11 #include <QVariant>
12 #include <QSortFilterProxyModel>
13 #include <QStandardItemModel>
14
15
19 class cSpectrumDetailProxyModel :  public QSortFilterProxyModel {
20
21     Q_OBJECT
22
23 private:
24
25     bool hidematched, hideunmatched, hidescrambled, hideionseries, hideneutralloss, hiderotations,
      hidetrotations;
26
27 public:
28
29
34     cSpectrumDetailProxyModel(QObject *parent = 0);
35
36
47     void setFlags(bool hidematched, bool hideunmatched, bool hidescrambled, bool hideionseries, bool
      hideneutralloss, bool hiderotations, bool hidetrotations);
48
49
50 protected:
51
52
59     bool filterAcceptsRow(int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE;
60
61
68     bool lessThan(const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE;
69
70
71 };
72
73
74 #endif
75
```

17.145 gui/cSpectrumDetailWidget.h File Reference

The widget for visualisation of a detail of a spectrum (opened after doubleclick on a row when results are reported).

```
#include <QWidget>
#include <QDesktopServices>
#include <QToolBar>
#include <QMenuBar>
#include <QMenu>
#include <QUrl>
#include <QFileInfo>
#include <QMainWindow>
#include <QTextDocument>
#include <QTableView>
#include <QStandardItemModel>
#include <QStandardItem>
#include <QHeaderView>
#include <QProgressDialog>
#include <QTextStream>
#include "core/cGlobalPreferences.h"
#include "core/cTheoreticalSpectrum.h"
#include "gui/cLinearWidget.h"
#include "gui/cCyclicWidget.h"
#include "gui/cBranchedWidget.h"
#include "gui/cBranchCyclicWidget.h"
#include "gui/cSpectrumSceneWidget.h"
```

```
#include "gui/cFindDialog.h"
#include "gui/cExportDialog.h"
#include "gui/cEventFilter.h"
#include "gui/cViewButtonDelegate.h"
#include "gui/cSpectrumDetailProxyModel.h"
#include "gui/cHTMLDelegate.h"
```

Classes

- struct cTablePosition

  Position in QTableWidget.
- class cSpectrumDetailWidget

  The class representing a widget for visualisation of a detail of a spectrum (opened after doubleclick on a row when results are reported).

17.145.1 Detailed Description

The widget for visualisation of a detail of a spectrum (opened after doubleclick on a row when results are reported).

17.146 cSpectrumDetailWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CSPECTRUMDETAILWIDGET_H
8 #define _CSPECTRUMDETAILWIDGET_H
9
10 #include <QWidget>
11 #include <QDesktopServices>
12 #include <QToolBar>
13 #include <QMenuBar>
14 #include <QMenu>
15 #include <QUrl>
16 #include <QFileInfo>
17 #include <QMainWindow>
18 #include <QTextDocument>
19 #include <QTableView>
20 #include <QStandardItemModel>
21 #include <QStandardItem>
22 #include <QHeaderView>
23 #include <QProgressDialog>
24 #include <QTextStream>
25 #include "core/cGlobalPreferences.h"
26 #include "core/cTheoreticalSpectrum.h"
27 #include "gui/cLinearWidget.h"
28 #include "gui/cCyclicWidget.h"
29 #include "gui/cBranchedWidget.h"
30 #include "gui/cBranchCyclicWidget.h"
31 #include "gui/cSpectrumSceneWidget.h"
32 #include "gui/cFindDialog.h"
33 #include "gui/cExportDialog.h"
34 #include "gui/cEventFilter.h"
35 #include "gui/cViewButtonDelegate.h"
36 #include "gui/cSpectrumDetailProxyModel.h"
37 #include "gui/cHTMLDelegate.h"
38
39
40 // forward declaration
41 class QTextBrowser;
42 class QHBoxLayout;
43 class QVBoxLayout;
44 class QFormLayout;
45 class QCheckBox;
46 class QPushButton;
47 class QSplitter;
48 class QComboBox;
49 class QDoubleSpinBox;
50 class QAction;
51 class QLabel;
52
53
57 struct cTablePosition {
61     int row;
62
```

```
63
67      int column;
68
69
70      cTablePosition() {
71          row = 0;
72          column = 0;
73      }
74  };
75
76
80  class cSpectrumDetailWidget :  public QMainWindow
81  {
82      Q_OBJECT
83
84  public:
85
86
90      cSpectrumDetailWidget();
91
92
97      cSpectrumDetailWidget(const cSpectrumDetailWidget& sd);
98
99
104      cSpectrumDetailWidget& operator=(const cSpectrumDetailWidget& sd);
105
106
110      ~cSpectrumDetailWidget();
111
112
117      void closeEvent(QCloseEvent *event);
118
119
130      void initialize(int rowid, string scantitle, int activefileid, cGlobalPreferences*
     globalpreferences, cParameters* parameters, cTheoreticalSpectrum& theoreticalspectrum, QWidget*
     parent);
131
132
137      cTheoreticalSpectrum& getTheoreticalSpectrum();
138
139
144      string getDetailsAsHTMLString();
145
146
154      string getPeaksTableAsHTMLString(bool unmatchedtheoreticalpeaks, bool unmatchedexperimentalpeaks,
     bool exportisomers);
155
156
163      string getPartialPeaksTableAsHTMLString(int id, bool exportisomers);
164
165
175      void prepareToShow(QAction* actionShowIsomers, cPeakListSeries* rawdata, vector<cImzMLItem>*
     imzmlprofilemetadata, bool profilemz64precision, bool profileintensity64precision, bool
     usedefaultsize);
176
177
184      void findAll(const QString& str, QTextDocument::FindFlags opt = 0, bool errormessage = true);
185
186
191      void exportImage(bool exportspectrum);
192
193
198      void zoomToPeak(double value);
199
200
205      set<string>& getLocalNeutralLosses();
206
207
212      void setProfileSpectrumEnabled(bool enable);
213
214
219      bool hasProfileSpectrumEnabled();
220
221
225      void disableProfileMode();
226
227
232      void setAbsoluteIntensityEnabled(bool enable);
233
234
239      bool hasAbsoluteIntensityEnabled();
240
241
246      void setHideMatchedPeaksEnabled(bool enable);
247
248
253      bool hasHideMatchedPeaksEnabled();
```

```
254
255
260     void setHideUnmatchedPeaksEnabled(bool enable);
261
262
267     bool hasHideUnmatchedPeaksEnabled();
268
269
274     void setHideScrambledPeaksEnabled(bool enable);
275
276
281     bool hasHideScrambledPeaksEnabled();
282
283
288     void setHidePeakLabelsEnabled(bool enable);
289
290
295     bool hasHidePeakLabelsEnabled();
296
297
302     void setMouseMzSelectionEnabled(bool enable);
303
304
309     bool hasMouseMzSelectionEnabled();
310
311
312 protected:
313
314
319     void keyPressEvent(QKeyEvent *event);
320
321
326     void moveEvent(QMoveEvent *event);
327
328
329 private:
330
331     QWidget* parent;
332     QAction* actionShowIsomers;
333
334     QMenuBar* menuBar;
335     QMenu* menuFile;
336     QMenu* menuFind;
337     QMenu* menuView;
338     QMenu* menuHelp;
339
340     QToolBar* toolbarFile;
341     QToolBar* toolbarFind;
342     QToolBar* toolbarView;
343     QToolBar* toolbarHelp;
344     QToolBar* toolbarMz;
345     QToolBar* toolbarIonSeries;
346     QToolBar* toolbarNeutralLoss;
347     QToolBar* toolbarRotation;
348     QToolBar* toolbarTrotation;
349
350     QAction* actionExportTable;
351     QAction* actionExportSpectrum;
352     QAction* actionCloseWindow;
353     QAction* actionFind;
354     QAction* actionPrevious;
355     QAction* actionNext;
356     QAction* actionZoomIn;
357     QAction* actionZoomOut;
358     QAction* actionZoomReset;
359     QAction* actionAbsoluteIntensity;
360     QAction* actionRawData;
361     QAction* actionHideMatched;
362     QAction* actionHideUnmatched;
363     QAction* actionHideScrambled;
364     QAction* actionHideLabels;
365     QAction* actionMouseMzSelection;
366     QAction* actionHTMLDocumentation;
367     QSplitter* hsplitter1;
368     QSplitter* hsplitter2;
369     QSplitter* vsplitter;
370
371     QWidget* widgetmz;
372     QHBoxLayout* hboxmz;
373     QLabel* labelmz;
374     QDoubleSpinBox* minmz;
375     QLabel* labelseparator;
376     QDoubleSpinBox* maxmz;
377     QPushButton* setmzinterval;
378     QPushButton* resetmzinterval;
379
380     QWidget* ionserieswidget;
```

```
381     QHBoxLayout* ionserieshbox;
382     QLabel* ionserieslabel;
383     QComboBox* ionseriescombobox;
384
385     QWidget* neutrallosswidget;
386     QHBoxLayout* neutrallosshbox;
387     QLabel* neutrallosslabel;
388     QComboBox* neutrallosscombobox;
389
390     QWidget* rotationwidget;
391     QHBoxLayout* rotationhbox;
392     QLabel* rotationlabel;
393     QComboBox* rotationcombobox;
394
395     QWidget* trotationwidget;
396     QHBoxLayout* trotationhbox;
397     QLabel* trotationlabel;
398     QComboBox* trotationcombobox;
399
400     QTextBrowser* textbrowser;
401
402     QTableView* peakstable;
403     QStandardItemModel* peakstablemodel;
404     cSpectrumDetailProxyModel* proxymodel;
405
406     vector<cTablePosition> tablematches;
407
408     cLinearWidget* linearwidget;
409     cCyclicWidget* cyclicwidget;
410     cBranchedWidget* branchedwidget;
411     cBranchCyclicWidget* branchcyclicwidget;
412
413     cSpectrumSceneWidget* spectrumscene;
414
415     cTheoreticalSpectrum* theoreticalspectrum;
416
417     int rowid;
418     string scantitle;
419     bool preparedToShow;
420
421     set<string> localneutralosses;
422
423     cGlobalPreferences* globalpreferences;
424     cParameters* parameters;
425
426     cPeakListSeries* rawdata;
427     vector<cImzMLItem>* imzmlprofilemetadata;
428     bool profilemz64precision;
429     bool profileintensity64precision;
430
431     cFindDialog* finddialog;
432     cExportDialog* exportdialog;
433     int currentfinditem;
434
435     bool reportisomers;
436
437     int activefileid;
438
439     void preparePeaksTable(QRect geometry);
440
441     string printHTMLTableCell(string text, bool red);
442
443
444 signals:
445
446
452     void emitMZInterval(double minmz, double maxmz);
453
454
459     void rawDataStateChangedSignal(bool state);
460
461
466     void absoluteIntensityStateChangedSignal(bool state);
467
468
473     void lastActiveDetail(int rowid);
474
475
476 private slots:
477
478
479     void updateMZInterval(double minmz, double maxmz);
480
481
482     void setMZInterval();
483
484
```

```
485     void exportSpectrum();
486
487
488     void exportPeptide();
489
490
491     void openFindDialog();
492
493
494     void openExportImageDialog();
495
496
497     void exportTableToCSV();
498
499
500     void movePrevious();
501
502
503     void moveNext();
504
505
506     void headerItemClicked(int index);
507
508
509     void filterPeaksTable();
510
511
512     void rawDataStateChanged(bool state);
513
514
515     void absoluteIntensityStateChanged(bool state);
516
517
518     void hideMatchedPeaks(bool hide);
519
520
521     void hideUnmatchedPeaks(bool hide);
522
523
524     void hideScrambledPeaks(bool hide);
525
526
527     void hidePeakLabels(bool hide);
528
529
530     void filterTableAfterIonSeriesChanged(int index);
531
532
533     void filterTableAfterNeutralLossChanged(int index);
534
535
536     void filterTableAfterRotationChanged(int index);
537
538
539     void filterTableAfterTRotationChanged(int index);
540
541
542     void showIsomersStateChanged();
543
544
545     void showHTMLDocumentation();
546
547
548     void closeWindow();
549
550
551     void tableDoubleClicked(const QModelIndex& index);
552
553 };
554
555 #endif
556
```

17.147 gui/cSpectrumSceneWidget.h File Reference

The implementation of a widget painting a 2D mass spectrum.

```
#include <iostream>
#include <QWidget>
#include <QGraphicsView>
#include "core/cTheoreticalSpectrum.h"
```

Classes

- class cSpectrumSceneWidget

  The widget representing a 2D mass spectrum.

17.147.1 Detailed Description

The implementation of a widget painting a 2D mass spectrum.

17.148 cSpectrumSceneWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CSPECTRUMSCENEWIDGET_H
8 #define _CSPECTRUMSCENEWIDGET_H
9
10 #include <iostream>
11 #include <QWidget>
12 #include <QGraphicsView>
13
14 #include "core/cTheoreticalSpectrum.h"
15
16
17 using namespace std;
18
19
20 // forward declaration
21 class QPaintEvent;
22 class QGraphicsScene;
23 class QMatrix;
24 class QGraphicsItemGroup;
25 class QGraphicsRectItem;
26 class QGraphicsSimpleTextItem;
27
28
32 class cSpectrumSceneWidget :  public QGraphicsView
33 {
34     Q_OBJECT
35
36 public:
37
38
43     cSpectrumSceneWidget(QWidget* parent = 0);
44
45
49     ~cSpectrumSceneWidget();
50
51
60     void initialize(cParameters* parameters, cTheoreticalSpectrum* theoreticalspectrum, cPeakListSeries*
      rawdata, int rowid, int activefileid);
61
62
68     void exportToPDF(QString filename, bool postscript);
69
70
75     void exportToSVG(QString filename);
76
77
82     void exportToPNG(QString filename);
83
84
89     void hideUnmatchedPeaks(bool state);
90
91
96     void hideMatchedPeaks(bool state);
97
98
103     void hideScrambledPeaks(bool state);
104
105
110     void hidePeakLabels(bool state);
111
112
113 protected:
114
115
120     void wheelEvent(QWheelEvent *event);
121
122
127     void mouseMoveEvent(QMouseEvent *event);
128
```

```
129
134     void mouseReleaseEvent(QMouseEvent *event);
135
136
141     void mousePressEvent(QMouseEvent *event);
142
143
148     void showEvent(QShowEvent *event);
149
150
155     void resizeEvent(QResizeEvent *event);
156
157
158 private:
159
160     QWidget* parent;
161     QGraphicsScene* scene;
162     QMatrix originalmatrix;
163
164     QGraphicsItemGroup* zoomgroup;
165     QGraphicsRectItem* zoomrect;
166     QGraphicsSimpleTextItem* zoomsimpletextitem;
167
168     QGraphicsSimpleTextItem* cursorsimpletextitem;
169
170     cParameters* parameters;
171     cTheoreticalSpectrum* theoreticalspectrum;
172     cPeaksList* rawdatapeaklist;
173
174     qreal currentscale;
175     qreal factor;
176
177     int origwidth;
178     int origheight;
179
180     bool calledbyresizeevent;
181     deque<int> oldwidth;
182     deque<int> oldheight;
183
184     string visibleionseriespart1;
185     string visibleionseriespart2;
186     string visibleneutralloss;
187     int coloredrotationid;
188     string coloredrotationstring;
189     int coloredtrotationid;
190
191     bool absoluteintensity;
192     bool rawdatastate;
193     bool hideunmatched;
194     bool hidematched;
195     bool hidescrambled;
196     bool hidelabels;
197
198     int pressedx;
199     int pressedy;
200
201     int currentx;
202     int currenty;
203
204     double minmzratio;
205     double maxmzratio;
206
207     int topmargin;
208     int bottommargin;
209     int leftmargin;
210     int rightmargin;
211
212     bool firstshow;
213     bool enablemousemzselection;
214
215     double maxmzoverhead;
216
217     int activefileid;
218
219
220     double getMZRatioFromXPosition(int x, int w);
221
222
223     int getXPositionFromMZRatio(double mzratio, int w);
224
225
226     double getIntensityFromYPosition(int y, int h);
227
228
229     void redrawScene();
230
231
```

```
232     void updateZoomGroup();
233
234
235     void calculateMinMaxMZ();
236
237
238     double getMaximumIntensity();
239
240
241 signals:
242
243
249     void updateMZInterval(double minmz, double maxmz);
250
251
252 private slots:
253
254
255     void zoomIn();
256
257
258     void zoomOut();
259
260
261     void normalSize();
262
263
264     void absoluteIntensityStateChanged(bool state);
265
266
267     void rawDataStateChanged(bool state);
268
269
270     void setMZInterval(double minmz, double maxmz);
271
272
273     void resetMZInterval();
274
275
276     void ionSeriesChanged(QString text);
277
278
279     void neutralLossChanged(QString text);
280
281
282     void rotationChanged(int index);
283
284
285     void trotationChanged(int index);
286
287
288     void rotationChanged(QString text);
289
290
291     void enableMouseMzSelectionTool(bool enable);
292
293
294 };
295
296 #endif
```

17.149 gui/cSummaryPeaksTableProxyModel.h File Reference

A table proxy model.

```
#include <QString>
#include <QVariant>
#include <QSortFilterProxyModel>
#include <QComboBox>
#include <QLineEdit>
#include "core/cParameters.h"
```

Classes

- class cSummaryPeaksTableProxyModel

  A table proxy model.

17.149.1 Detailed Description

A table proxy model.

17.150 cSummaryPeaksTableProxyModel.h

Go to the documentation of this file.

```
1
7 #ifndef _CSUMMARYPEAKSTABLEPROXYMODEL_H
8 #define _CSUMMARYPEAKSTABLEPROXYMODEL_H
9
10 #include <QString>
11 #include <QVariant>
12 #include <QSortFilterProxyModel>
13 #include <QComboBox>
14 #include <QLineEdit>
15 #include "core/cParameters.h"
16
17
21 class cSummaryPeaksTableProxyModel :  public QSortFilterProxyModel {
22
23     Q_OBJECT
24
25 private:
26
27     eModeType mode;
28     ePeakListFileFormat peaklistfileformat;
29     bool generateisotopepattern;
30     QComboBox* filteroperator;
31     QComboBox* filtercombobox1;
32     QComboBox* filtercomparatorcombobox1;
33     QLineEdit* filterline1;
34     QComboBox* filtercombobox2;
35     QComboBox* filtercomparatorcombobox2;
36     QLineEdit* filterline2;
37     bool wholeword;
38
39
40 public:
41
45     int xmin;
46
47
51     int xmax;
52
53
57     int ymin;
58
59
63     int ymax;
64
65
70     cSummaryPeaksTableProxyModel(QObject *parent = 0);
71
72
86     void initialize(eModeType mode, ePeakListFileFormat peaklistfileformat, bool generateisotopepattern,
       QComboBox* rowsfilteroperator, QComboBox* rowsfiltercombobox1, QComboBox*
       rowsfiltercomparatorcombobox1, QLineEdit* rowsfilterline1, QComboBox* rowsfiltercombobox2, QComboBox*
       rowsfiltercomparatorcombobox2, QLineEdit* rowsfilterline2);
87
88
96     void setRegion(int xmin, int xmax, int ymin, int ymax);
97
98
103     void setWholeWord(bool wholeword);
104
105
106 protected:
107
108
115     bool filterAcceptsRow(int sourceRow, const QModelIndex &sourceParent) const Q_DECL_OVERRIDE;
116
117
124     bool lessThan(const QModelIndex &left, const QModelIndex &right) const Q_DECL_OVERRIDE;
125
126
127 };
128
129
130 #endif
131
```

17.151 gui/cSummaryPeaksTableWidget.h File Reference

Summary table of matched peaks.

```
#include <map>
#include <QWidget>
#include <QDesktopServices>
#include <QUrl>
#include <QFileInfo>
#include <QMainWindow>
#include <QToolBar>
#include <QAction>
#include <QTableView>
#include <QStandardItemModel>
#include <QStandardItem>
#include <QItemDelegate>
#include <QMessageBox>
#include <QComboBox>
#include <QStringListModel>
#include <QCompleter>
#include "core/utilities.h"
#include "core/cGlobalPreferences.h"
#include "gui/cViewButtonDelegate.h"
#include "gui/cSummaryPeaksTableProxyModel.h"
#include "gui/cMainWindowProxyModel.h"
#include "gui/cHTMLDelegate.h"
#include "gui/cPubChemSearchWidget.h"
#include "gui/cEventFilter.h"
```

Classes

- struct cSummaryTableKeyMS

  Key to reduce the summary table when exporting statistics for MS and MSI data.
- struct cSummaryTableKeyMSMS

  Key to reduce the summary table when exporting statistics for MS/MS data.
- struct cIonSummaryTableKeyMS_comp

  Comparison object for cSummaryTableKeyMS (independent ions).
- struct cEnvelopeSummaryTableKeyMS_comp

  Comparison object for cSummaryTableKeyMS (whole envelope).
- struct cSummaryTableKeyMSMS_comp

  Comparison object for cSummaryTableKeyMSMS.
- class cSummaryPeaksTableWidget

  Summary table of matched peaks.

Functions

- void addEICPeak (cParameters *parameters, cPeaksList &eicchromatogram, cTheoreticalSpectrum &theoreticalspectrum, cPeaksList &experimentalspectrum)

  Add a peak to an extracted ion chromatogram.

17.151.1 Detailed Description

Summary table of matched peaks.

17.151.2 Function Documentation

17.151.2.1 addEICPeak()

```
void addEICPeak (
            cParameters * parameters,
            cPeaksList & eicchromatogram,
            cTheoreticalSpectrum & theoreticalspectrum,
            cPeaksList & experimentalspectrum )
```

Add a peak to an extracted ion chromatogram.

Parameters

parameters	application parameters
eicchromatogram	EIC chromatogram
theoreticalspectrum	theoretical spectrum
experimentalspectrum	experimental spectrum

17.152 cSummaryPeaksTableWidget.h

Go to the documentation of this file.

```
1
7 #ifndef _CSUMMARYPEAKSTABLEWIDGET_H
8 #define _CSUMMARYPEAKSTABLEWIDGET_H
9
10 #include <map>
11 #include <QWidget>
12 #include <QDesktopServices>
13 #include <QUrl>
14 #include <QFileInfo>
15 #include <QMainWindow>
16 #include <QToolBar>
17 #include <QAction>
18 #include <QTableView>
19 #include <QStandardItemModel>
20 #include <QStandardItem>
21 #include <QItemDelegate>
22 #include <QMessageBox>
23 #include <QComboBox>
24 #include <QStringListModel>
25 #include <QCompleter>
26
27 #include "core/utilities.h"
28 #include "core/cGlobalPreferences.h"
29 #include "gui/cViewButtonDelegate.h"
30 #include "gui/cSummaryPeaksTableProxyModel.h"
31 #include "gui/cMainWindowProxyModel.h"
32 #include "gui/cHTMLDelegate.h"
33 #include "gui/cPubChemSearchWidget.h"
34 #include "gui/cEventFilter.h"
35
36
37 // forward declaration
38 class cParameters;
39 class cTheoreticalSpectrum;
40 class cTheoreticalSpectrumList;
41 class QHBoxLayout;
42 class QVBoxLayout;
43 class QPushButton;
44 class QCheckBox;
45 class QLineEdit;
46 class QMenuBar;
47 class QMenu;
48
49
53 struct cSummaryTableKeyMS {
54
58     string id;
59
60
64     string iontype;
65
66
70     string theoreticalmz;
71
72
76     string summaryformula;
```

```
77
78
82     string name;
83
84
88     string reference;
89
90
94     string score;
95
96
100     string fdr;
101
102
106     void clear() {
107         id.clear();
108         iontype.clear();
109         theoreticalmz.clear();
110         summaryformula.clear();
111         name.clear();
112         reference.clear();
113         score.clear();
114         fdr.clear();
115     }
116
117 };
118
119
123 struct cSummaryTableKeyMSMS {
124
128     string theoreticalmz;
129
130
134     string thint;
135
136
140     string experimentalmz;
141
142
146     string relint;
147
148
152     string absint;
153
154
158     string ppmerror;
159
160
164     string summary;
165
166
170     void clear() {
171         theoreticalmz.clear();
172         thint.clear();
173         experimentalmz.clear();
174         relint.clear();
175         absint.clear();
176         ppmerror.clear();
177         summary.clear();
178     }
179
180 };
181
182
186 struct cIonSummaryTableKeyMS_comp {
187
194     bool operator()(const cSummaryTableKeyMS& left, const cSummaryTableKeyMS& right)const {
195         return (left.iontype + left.theoreticalmz + left.summaryformula + left.name + left.reference <
      right.iontype + right.theoreticalmz + right.summaryformula + right.name + right.reference);
196     }
197
198 };
199
200
204 struct cEnvelopeSummaryTableKeyMS_comp {
205
212     bool operator()(const cSummaryTableKeyMS& left, const cSummaryTableKeyMS& right)const {
213         return (left.id + left.iontype + left.name + left.reference < right.id + right.iontype +
      right.name + right.reference);
214     }
215
216 };
217
218
222 struct cSummaryTableKeyMSMS_comp {
223
230     bool operator()(const cSummaryTableKeyMSMS& left, const cSummaryTableKeyMSMS& right)const {
```

```
231         return (left.theoreticalmz + left.thint + left.experimentalmz + left.relint + left.absint +
      left.ppmerror + left.summary < right.theoreticalmz + right.thint + right.experimentalmz + right.relint
      + right.absint + right.ppmerror + right.summary);
232     }
233
234 };
235
236
244 void addEICPeak(cParameters* parameters, cPeaksList& eicchromatogram, cTheoreticalSpectrum&
      theoreticalspectrum, cPeaksList& experimentalspectrum);
245
246
250 class cSummaryPeaksTableWidget :  public QMainWindow
251 {
252     Q_OBJECT
253
254 public:
255
256
262     cSummaryPeaksTableWidget(cGlobalPreferences* globalpreferences, QWidget* parent = (QWidget *)0);
263
264
268     ~cSummaryPeaksTableWidget();
269
270
275     void closeEvent(QCloseEvent *event);
276
277
288     bool prepareToShow(QStandardItemModel* resultsstandardmodel, cMainWindowProxyModel*
      resultsproxymodel, cParameters* parameters, cTheoreticalSpectrumList* listoftheoreticalspectra, int
      activefileid, bool showisomers);
289
290
294     void deleteTable();
295
296
304     void updateFilterBySelectedRegion(int xmin, int xmax, int ymin, int ymax);
305
306
311     void applyGlobalPreferences(cGlobalPreferences* globalpreferences);
312
313
318     cPubChemSearchWidget* getPubChemSearchWidget();
319
320
324     void filterTablerows();
325
326
334     void filterCompound(string name, string iontype, int datatypeview, string mzstr);
335
336
337 private:
338
339     QString title;
340
341     cGlobalPreferences* globalpreferences;
342
343     cParameters* parameters;
344     QWidget* parent;
345
346     cPubChemSearchWidget* pubchemsearchwidget;
347
348     QMenuBar* menuBar;
349     QMenu* menuFile;
350     //QMenu* menuSearch;
351     QMenu* menuHelp;
352
353     QToolBar* toolbarFile;
354     QAction* actionExportCSV;
355     QAction* actionExportDatabase;
356     QAction* actionExportStatistics;
357     QAction* actionCloseWindow;
358
359     //QToolBar* toolbarSearch;
360     //QAction* actionSearchPubChem;
361
362     QToolBar* toolbarHelp;
363     QAction* actionHTMLDocumentation;
364
365     QToolBar* toolbarFilter;
366     QWidget* rowsfilterwidget;
367     QHBoxLayout* rowsfilterhbox;
368
369     QComboBox* rowsfilteroperator;
370
371     QComboBox* rowsfiltercombobox1;
372     QComboBox* rowsfiltercomparatorcombobox1;
```

```
373     QStringListModel* rowsfilterstringlistmodel1;
374     QCompleter* rowsfilterlinecompleter1;
375     QLineEdit* rowsfilterline1;
376     QPushButton* rowsfilterleft1;
377     QPushButton* rowsfilterright1;
378
379     QComboBox* rowsfiltercombobox2;
380     QComboBox* rowsfiltercomparatorcombobox2;
381     QStringListModel* rowsfilterstringlistmodel2;
382     QCompleter* rowsfilterlinecompleter2;
383     QLineEdit* rowsfilterline2;
384     QPushButton* rowsfilterleft2;
385     QPushButton* rowsfilterright2;
386
387     QCheckBox* rowsfiltercasesensitive;
388     QCheckBox* rowsfilterwholeword;
389     QPushButton* rowsfilterbutton;
390     QPushButton* rowsfilterclearbutton;
391
392     QTableView* database;
393     QStandardItemModel* databasemodel;
394     cSummaryPeaksTableProxyModel* proxymodel;
395     QVBoxLayout* mainlayout;
396     QWidget* mainwidget;
397
398     QString lastdirexporttocsv;
399     QString lastdirexportdatabase;
400     QString lastdirexportstatisticstocsv;
401
402     vector<double> rtimes;
403
404     vector<cCoordinateInfo> origcoordinateinfo;
405     cPeaksList origeicchromatogram;
406
407     int activefileid;
408
409     void addCoordinateInfo(int spectrumindex, vector<cCoordinateInfo>& coordinateinfo,
      cTheoreticalSpectrum& theoreticalspectrum, cPeaksList& experimentalspectrum);
410
411
412 protected:
413
414
419     void keyPressEvent(QKeyEvent *event);
420
421
422 private slots:
423
424     void closeWindow();
425
426
427     void filterRows();
428
429
430     void resetFilter();
431
432
433     void exportToCsv();
434
435
436     void exportToDatabase();
437
438
439     void exportStatistics();
440
441
442     void showHTMLDocumentation();
443
444
445     void rowDoubleClicked(const QModelIndex& item);
446
447
448     void rowsFilterLeft1Slot();
449
450
451     void rowsFilterRight1Slot();
452
453
454     void rowsFilterLeft2Slot();
455
456
457     void rowsFilterRight2Slot();
458
459
460     void searchPubChem();
461
462
```

```
463 signals:
464
465
469     void tableCancelled();
470
471
477     void summaryPeaksTableRowDoubleClicked(int rowid, double experimentalmz);
478
479
483     void resetRegion();
484
485
499     void sendFilterOptionsToImageWindow(vector<cCoordinateInfo> coordinateinfo, bool operatortype,
      string columnname1, string comparatorname1, string filterstring1, string columnname2, string
      comparatorname2, string filterstring2, bool casesensitive, bool wholeword);
500
501
506     void sendFilterOptionsToChromatogram(cPeaksList eic);
507
508
509 };
510
511 #endif
512
```

17.153 gui/cViewButtonDelegate.h File Reference

A delegate to create a reference view button.

```
#include <QString>
#include <QLocale>
#include <QVariant>
#include <QStyledItemDelegate>
#include <QPainter>
#include <QModelIndex>
#include <QEvent>
#include <QAbstractItemModel>
```

Classes

- class cViewButtonDelegate

  A delegate to create a reference view button.

17.153.1 Detailed Description

A delegate to create a reference view button.

17.154 cViewButtonDelegate.h

Go to the documentation of this file.

```
1
7 #ifndef _CVIEWBUTTONDELEGATE_H
8 #define _CVIEWBUTTONDELEGATE_H
9
10 #include <QString>
11 #include <QLocale>
12 #include <QVariant>
13 #include <QStyledItemDelegate>
14 #include <QPainter>
15 #include <QModelIndex>
16 #include <QEvent>
17 #include <QAbstractItemModel>
18
19
23 class cViewButtonDelegate :  public QStyledItemDelegate {
24
25     Q_OBJECT
26
27 public:
28
29
```

```
34      cViewButtonDelegate(QObject* parent = (QObject *)0);
35
36
43      QString displayText(const QVariant& value, const QLocale& locale) const;
44
45
52      void paint(QPainter* painter, const QStyleOptionViewItem& option, const QModelIndex& index) const;
53
54
55  protected:
56
57
65      bool editorEvent(QEvent* event, QAbstractItemModel* model, const QStyleOptionViewItem& option, const
        QModelIndex& index);
66
67  };
68
69
70  #endif
71
```

17.155 parallel/cFormulaPredictorThread.h File Reference

The thread which searches molecular formulas corresponding to m/z value.

```
#include <QThread>
#include "core/utilities.h"
#include "core/cSummaryFormula.h"
#include "core/cBricksDatabase.h"
#include "core/cParameters.h"
```

Classes

- struct cFormulaPredictorRowItem

  An auxiliary structure representing a row in the output table.
- class cFormulaPredictorThread

  The thread which searches molecular formulas corresponding to m/z value.

17.155.1 Detailed Description

The thread which searches molecular formulas corresponding to m/z value.

17.156 cFormulaPredictorThread.h

Go to the documentation of this file.

```
1
7  #ifndef _CFORMULAPREDICTORTHREAD_H
8  #define _CFORMULAPREDICTORTHREAD_H
9
10 #include <QThread>
11 #include "core/utilities.h"
12 #include "core/cSummaryFormula.h"
13 #include "core/cBricksDatabase.h"
14 #include "core/cParameters.h"
15
16
20 struct cFormulaPredictorRowItem {
21
25      string ionformula;
26
27
31      double theoreticalmz;
32
33
37      double mzerror;
38
39
43      string iontype;
44
45
49      int charge;
50
```

```
51 };
52
53
57 class cFormulaPredictorThread :  public QThread {
58
59     Q_OBJECT
60
61 private:
62
63     int chargevalue;
64     int maximumcombinedelements;
65     bool basicformulacheck;
66     bool advancedformulacheck;
67     bool noratiocheck;
68     double mzvalue;
69     double mzerror;
70     vector<cIonType>* selectedions;
71     vector<string>* selectedelements;
72     vector<cFormulaPredictorRowItem>* ionitems;
73     bool* success;
74     bool* terminatecomputation;
75     string* errmsg;
76
77     int generateCompoundsFormulaPredictor(vector<string>& formulas);
78
79
80 public:
81
82
86     cFormulaPredictorThread();
87
88
105     void initialize(int chargevalue, int maximumcombinedelements, bool basicformulacheck, bool
      advancedformulacheck, bool noratiocheck, double mzvalue, double mzerror, vector<cIonType>&
      selectedions, vector<string>& selectedelements, vector<cFormulaPredictorRowItem>& ionitems, bool&
      success, bool& terminatecomputation, string& errmsg);
106
107
108 protected:
109
110
114     void run();
115
116 };
117
118
119 #endif
120
```

17.157 parallel/cGraphReaderThread.h File Reference

The thread which scans the de novo graph for peptide sequence candidates.

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <QThread>
#include "core/cParameters.h"
#include "core/cCandidateSet.h"
#include "core/cDeNovoGraphNode.h"
#include "core/cBricksDatabase.h"
```

Classes

- class cGraphReaderThread

  The class representing a thread which scans the de novo graph for peptide sequence candidates.

17.157.1 Detailed Description

The thread which scans the de novo graph for peptide sequence candidates.

17.158 cGraphReaderThread.h

Go to the documentation of this file.

```
1
7 #ifndef _CGRAPHREADERTHREAD_H
8 #define _CGRAPHREADERTHREAD_H
9
10 #include <iostream>
11 #include <string>
12 #include <vector>
13 #include <algorithm>
14 #include <QThread>
15
16 #include "core/cParameters.h"
17 #include "core/cCandidateSet.h"
18 #include "core/cDeNovoGraphNode.h"
19 #include "core/cBricksDatabase.h"
20
21
22 using namespace std;
23
24
25 class cMainThread;
26
27
31 class cGraphReaderThread :  public QThread {
32
33     Q_OBJECT
34
35 private:
36
37     vector<cDeNovoGraphNode>* graph;
38     cBricksDatabase* bricksdatabasewithcombinations;
39     cCandidateSet* candidates;
40     cParameters* parameters;
41     cMainThread* os;
42     int lastsystemnode;
43     long long unsigned* count;
44     int scanmode;
45     bool* terminatecomputation;
46
47     int getCandidatesIter(eFragmentIonType startiontype, cCandidateSet* candidates, int nodeid,
      vector<string>& composition, double precursormass, int startmodifID, int endmodifID, int
      middlemodifID, int middlepos, vector<nodeEdge>& perspectivepath, double cummass, string& startsummary,
      bool* terminatecomputation);
48
49     double getCandidatePrecursorMass(string& startsummary, vector<nodeEdge>& perspectivepath);
50
51     cSummaryFormula getCandidateSummaryFormula(string& startsummary, vector<nodeEdge>& perspectivepath);
52
53 public:
54
55
59     cGraphReaderThread();
60
61
74     void initialize(vector<cDeNovoGraphNode>& graph, cBricksDatabase& bricksdatabasewithcombinations,
      cCandidateSet& candidates, cParameters* parameters, cMainThread* os, int lastsystemnode, long long
      unsigned& count, int scanmode, bool& terminatecomputation);
75
76
77 protected:
78
79
83     void run();
84
85 };
86
87
88 #endif
89
```

17.159 parallel/cSpectrumAnalyzerThreadMS1.h File Reference

The thread for an initial check of LC-MS and MSI spectra.

```
#include <QRunnable>
#include <QObject>
#include "core/cTheoreticalSpectrum.h"
```

Classes

- class cSpectrumAnalyzerThreadMS1

  The class representing a thread for an initial check of LC-MS and MSI spectra.

17.159.1 Detailed Description

The thread for an initial check of LC-MS and MSI spectra.

17.160 cSpectrumAnalyzerThreadMS1.h

Go to the documentation of this file.

```
1
7 #ifndef _CSPECTRUMANALYZERTHREADMS1_H
8 #define _CSPECTRUMANALYZERTHREADMS1_H
9
10 #include <QRunnable>
11 #include <QObject>
12
13 #include "core/cTheoreticalSpectrum.h"
14
15
16 class cMainThread;
17
18
22 class cSpectrumAnalyzerThreadMS1 :  public QObject, public QRunnable {
23
24     Q_OBJECT
25
26 private:
27
28     cMainThread* mainthread;
29     cParameters* parameters;
30     int peaklistseriesvectorid;
31     cTheoreticalSpectrum* poolspectrum;
32     vector< vector<int> >* hintsindex;
33     bool* terminatecomputation;
34
35
36 public:
37
38
48     void initialize(cMainThread* mainthread, cParameters* parameters, int peaklistseriesvectorid,
      cTheoreticalSpectrum* poolspectrum, vector< vector<int> >* hintsindex, bool* terminatecomputation);
49
50
51 protected:
52
53
57     void run();
58
59 };
60
61
62 #endif
63
```

17.161 parallel/cSpectrumComparatorThreadMS1.h File Reference

The thread for comparison of a theoretical spectrum with a peak list (MS1 level).

```
#include <QRunnable>
#include <QObject>
#include "core/cTheoreticalSpectrum.h"
```

Classes

- class cSpectrumComparatorThreadMS1

  The class representing a thread for comparison of a theoretical spectrum with a peak list (MS1 level).

17.161.1 Detailed Description

The thread for comparison of a theoretical spectrum with a peak list (MS1 level).

17.162 cSpectrumComparatorThreadMS1.h

Go to the documentation of this file.

```
1
7 #ifndef _CSPECTRUMCOMPARATORTHREADMS1_H
8 #define _CSPECTRUMCOMPARATORTHREADMS1_H
9
10 #include <QRunnable>
11 #include <QObject>
12
13 #include "core/cTheoreticalSpectrum.h"
14
15
16 class cTheoreticalSpectrumList;
17 class cMainThread;
18
19
23 class cSpectrumComparatorThreadMS1 :  public QObject, public QRunnable {
24
25     Q_OBJECT
26
27 private:
28
29     cMainThread* mainthread;
30     cParameters* parameters;
31     int peaklistseriesvectorid;
32     cTheoreticalSpectrum* poolspectrum;
33     vector< vector<int> >* hintsindex;
34     cTheoreticalSpectrumList* theoreticalspectrumlist;
35     vector<cPeaksList>* unmatchedpeaks;
36     bool* terminatecomputation;
37
38
39 public:
40
41
53     void initialize(cMainThread* mainthread, cParameters* parameters, int peaklistseriesvectorid,
      cTheoreticalSpectrum* poolspectrum, vector< vector<int> >* hintsindex, cTheoreticalSpectrumList*
      theoreticalspectrumlist, vector<cPeaksList>* unmatchedpeaks, bool* terminatecomputation);
54
55
56 protected:
57
58
62     void run();
63
64 };
65
66
67 #endif
68
```

17.163 parallel/cSpectrumComparatorThreadMS2.h File Reference

The thread for comparison of a theoretical spectrum with a peak list (MS2 level).

```
#include <QRunnable>
#include <QObject>
#include "core/cTheoreticalSpectrum.h"
```

Classes

- class cSpectrumComparatorThreadMS2

  The class representing a thread for comparison of a theoretical spectrum with a peak list (MS2 level).

Functions

- bool comparePaths (vector< nodeEdge > &a, vector< nodeEdge > &b)

  Compare two paths of a peptide sequence candidate as a string.

- bool compareBandAllIonsDesc (const cTheoreticalSpectrum &a, const cTheoreticalSpectrum &b)

  Compare scores of two theoretical spectra (number of b-ions and all ions secondly).
- bool compareYBandAllIonsDesc (const cTheoreticalSpectrum &a, const cTheoreticalSpectrum &b)

  Compare scores of two theoretical spectra (number of b-ions + y-ions and all ions as a secondary score).
- bool compareYandAllIonsDesc (const cTheoreticalSpectrum &a, const cTheoreticalSpectrum &b)

  Compare scores of two theoretical spectra (number of y-ions and all ions as a secondary score).
- bool compareSumOfRelIntDesc (const cTheoreticalSpectrum &a, const cTheoreticalSpectrum &b)

  Compare scores of two theoretical spectra (sum of relative intensities of matched peaks).
- bool compareNumberOfMatchedPeaksDesc (const cTheoreticalSpectrum &a, const cTheoreticalSpectrum &b)

  Compare scores of two theoretical spectra (number of matched peaks).
- bool compareWeightedRatioDesc (const cTheoreticalSpectrum &a, const cTheoreticalSpectrum &b)

  Compare scores of two theoretical spectra (weighted ratio of matched peaks).
- bool compareCosineSimilarityDesc (const cTheoreticalSpectrum &a, const cTheoreticalSpectrum &b)

  Compare scores of two theoretical spectra (cosine similarity).

17.163.1 Detailed Description

The thread for comparison of a theoretical spectrum with a peak list (MS2 level).

17.163.2 Function Documentation

17.163.2.1 compareBandAllIonsDesc()

```
bool compareBandAllIonsDesc (
            const cTheoreticalSpectrum & a,
            const cTheoreticalSpectrum & b )
```

Compare scores of two theoretical spectra (number of b-ions and all ions secondly).

Parameters

a	first theoretical spectrum
b	second theoretical spectrum

Return values

bool	true if the score of *a* is greater than the score of *b*

17.163.2.2 compareCosineSimilarityDesc()

```
bool compareCosineSimilarityDesc (
            const cTheoreticalSpectrum & a,
            const cTheoreticalSpectrum & b )
```

Compare scores of two theoretical spectra (cosine similarity).

Parameters

a	first theoretical spectrum
b	second theoretical spectrum

Return values

bool	true if the score of *a* is greater than the score of *b*

17.163.2.3 compareNumberOfMatchedPeaksDesc()

```
bool compareNumberOfMatchedPeaksDesc (
            const cTheoreticalSpectrum & a,
            const cTheoreticalSpectrum & b )
```

Compare scores of two theoretical spectra (number of matched peaks).

Parameters

a	first theoretical spectrum
b	second theoretical spectrum

Return values

bool	true if the score of *a* is greater than the score of *b*

17.163.2.4 comparePaths()

```
bool comparePaths (
            vector< nodeEdge > & a,
            vector< nodeEdge > & b )
```

Compare two paths of a peptide sequence candidate as a string.

Parameters

a	first theoretical spectrum
b	second theoretical spectrum

Return values

bool	true if the path of *a* is smaller than the path of *b*

17.163.2.5 compareSumOfRelIntDesc()

```
bool compareSumOfRelIntDesc (
            const cTheoreticalSpectrum & a,
            const cTheoreticalSpectrum & b )
```

Compare scores of two theoretical spectra (sum of relative intensities of matched peaks).

Parameters

a	first theoretical spectrum
b	second theoretical spectrum

Return values

bool	true if the score of *a* is greater than the score of *b*

17.163.2.6 compareWeightedRatioDesc()

```
bool compareWeightedRatioDesc (
            const cTheoreticalSpectrum & a,
            const cTheoreticalSpectrum & b )
```

Compare scores of two theoretical spectra (weighted ratio of matched peaks).

Parameters

a	first theoretical spectrum
b	second theoretical spectrum

Return values

bool	true if the score of *a* is greater than the score of *b*

17.163.2.7 compareYandAllIonsDesc()

```
bool compareYandAllIonsDesc (
            const cTheoreticalSpectrum & a,
            const cTheoreticalSpectrum & b )
```

Compare scores of two theoretical spectra (number of y-ions and all ions as a secondary score).

Parameters

a	first theoretical spectrum
b	second theoretical spectrum

Return values

bool	true if the score of *a* is greater than the score of *b*

17.163.2.8 compareYBandAllIonsDesc()

```
bool compareYBandAllIonsDesc (
            const cTheoreticalSpectrum & a,
            const cTheoreticalSpectrum & b )
```

Compare scores of two theoretical spectra (number of b-ions + y-ions and all ions as a secondary score).

Parameters

a	first theoretical spectrum
b	second theoretical spectrum

Return values

bool	true if the score of *a* is greater than the score of *b*

17.164 cSpectrumComparatorThreadMS2.h

Go to the documentation of this file.

```
1
7 #ifndef _CSPECTRUMCOMPARATORTHREADMS2_H
8 #define _CSPECTRUMCOMPARATORTHREADMS2_H
9
10 #include <QRunnable>
11 #include <QObject>
12
13 #include "core/cTheoreticalSpectrum.h"
14
15
16 class cTheoreticalSpectrumList;
17
24 bool comparePaths(vector<nodeEdge>& a, vector<nodeEdge>& b);
25
26
33 bool compareBandAllIonsDesc(const cTheoreticalSpectrum& a, const cTheoreticalSpectrum& b);
34
35
42 bool compareYBandAllIonsDesc(const cTheoreticalSpectrum& a, const cTheoreticalSpectrum& b);
43
44
51 bool compareYandAllIonsDesc(const cTheoreticalSpectrum& a, const cTheoreticalSpectrum& b);
52
53
60 bool compareSumOfRelIntDesc(const cTheoreticalSpectrum& a, const cTheoreticalSpectrum& b);
61
62
69 bool compareNumberOfMatchedPeaksDesc(const cTheoreticalSpectrum& a, const cTheoreticalSpectrum& b);
70
71
78 bool compareWeightedRatioDesc(const cTheoreticalSpectrum& a, const cTheoreticalSpectrum& b);
79
80
87 bool compareCosineSimilarityDesc(const cTheoreticalSpectrum& a, const cTheoreticalSpectrum& b);
88
89
93 class cSpectrumComparatorThreadMS2 :  public QObject, public QRunnable {
94
95     Q_OBJECT
96
97 private:
98
99     cCandidate candidate;
100     cPeaksList peaklist;
101     cTheoreticalSpectrumList* theoreticalspectrumlist;
102     cParameters* parameters;
103     regex* rxsequencetag;
104     regex* rxsearchedsequence;
105     double lastavailableworstscore;
106     bool* terminatecomputation;
107     cBricksDatabase* bricksdatabasewithcombinations;
108
109 public:
110
111
124     void initialize(cCandidate& candidate, cPeaksList& peaklist, cBricksDatabase*
      bricksdatabasewithcombinations, cTheoreticalSpectrumList* theoreticalspectrumlist, cParameters*
      parameters, regex* rxsequencetag, regex* rxsearchedsequence, double currentworstscore, bool*
      terminatecomputation);
125
126
127 protected:
128
129
133     void run();
134
135 };
136
137
138 #endif
139
```

17.165 parallel/cSpectrumGeneratorThreadMS1.h File Reference

The thread for generating a theoretical spectrum (MS1 level).

```
#include <QRunnable>
#include <QObject>
#include "core/cTheoreticalSpectrum.h"
```

Classes

- class cSpectrumGeneratorThreadMS1

  The class representing a thread for generating a theoretical spectrum (MS1 level).

17.165.1 Detailed Description

The thread for generating a theoretical spectrum (MS1 level).

17.166 cSpectrumGeneratorThreadMS1.h

Go to the documentation of this file.

```
1
7 #ifndef _CSPECTRUMGENERATORTHREADMS1_H
8 #define _CSPECTRUMGENERATORTHREADMS1_H
9
10 #include <QRunnable>
11 #include <QObject>
12
13 #include "core/cTheoreticalSpectrum.h"
14
15
16 class cMainThread;
17
18
22 class cSpectrumGeneratorThreadMS1 :  public QObject, public QRunnable {
23
24     Q_OBJECT
25
26 private:
27
28     cMainThread* mainthread;
29     cParameters* parameters;
30     int sequencestart;
31     int sequencestop;
32     cTheoreticalSpectrum* poolspectrum;
33     bool* terminatecomputation;
34
35
36 public:
37
38
48     void initialize(cMainThread* mainthread, cParameters* parameters, int sequencestart, int
       sequencestop, cTheoreticalSpectrum* poolspectrum, bool* terminatecomputation);
49
50
51 protected:
52
53
57     void run();
58
59 };
60
61
62 #endif
63
```

Index

absoluteIntensityStateChangedSignal
    cSpectrumDetailWidget, 412
activeLayerChanged
    cImageWindow, 288
add
    cPeaksList, 364
    cTheoreticalSpectrumList, 460
addEICPeak
    cSummaryPeaksTableWidget.h, 710
addFormula
    cSummaryFormula, 426
addIonAnnotation
    cDeNovoGraphNode, 263
addItem
    cNeutralLossesListWidget, 334
addLayer
    cImageWindow, 288
addPeakList
    cPeakListSeries, 357
addStringFormulaToMap
    cSummaryFormula.h, 551
addToParallelOutputState
    cMainThread, 317
applyGlobalPreferences
    cBricksDatabaseWidget, 210
    cCalibrationChart, 212
    cChromatogramWindow, 243
    cImageWindow, 288
    cModificationsWidget, 325
    cMultipleDatasetsTableWidget, 330
    cParametersWidget, 345
    cPatternSimulatorWidget, 352
    cSequenceDatabaseWidget, 403
    cSummaryPeaksTableWidget, 434
attach
    cPeaksList, 364
    cTheoreticalSpectrum, 443
attachDecoys
    cSequenceDatabase, 398
attachVectorOfCIDs
    cPubchemCIDReader, 386

cAboutWidget, 185
    cAboutWidget, 185
    closeEvent, 186
    keyPressEvent, 186
calculateAvgSpectrum
    cChromatogramWindow, 243
calculateBaseTau
    cChromatogramWindowWidget.h, 601
calculateCaruana
    cChromatogramWindowWidget.h, 602
calculateExponentialParameters
    cChromatogramWindowWidget.h, 602
calculateGaussianParameters
    cChromatogramWindowWidget.h, 603
calculateGuo
    cChromatogramWindowWidget.h, 603
calculateLineParameterA
    cMultipleDatasetsTableWidget.h, 666
calculateLineParameters
    cMultipleDatasetsTableWidget.h, 667
calculateLineParametersWithIterations
    cMultipleDatasetsTableWidget.h, 667
calculateLOD
    cCalibrationChartScene.h, 593
calculateLOQ
    cCalibrationChartScene.h, 593
calculateNeutralLosses
    cParameters, 341
calculatePrecursorMassFromBricks
    cCandidate, 225
calculateR2Value
    cCalibrationChartScene.h, 594
calculateSummaryFormulaFromBricks
    cCandidate, 225
calculateSummaryFromComposition
    cBricksDatabase, 200
calculateTauUsingWeightedAverage
    cChromatogramWindowWidget.h, 604
cBranchCyclicWidget, 186
    exportToPDF, 187
    exportToPNG, 187
    exportToSVG, 187
    hidePeakLabels, 188
    initialize, 188
    paintEvent, 188
    setReportIsomers, 188
cBranchCyclicWidget.h
    generateBranchLabelsDown, 581
    generateBranchLabelsUp, 582
cBranchedWidget, 189
    exportToPDF, 189
    exportToPNG, 189
    exportToSVG, 190
    hidePeakLabels, 190
    initialize, 190
    paintEvent, 190
    setReportIsomers, 191

cBrick, 191
empty, 192
explodeToIntComposition, 193
explodeToStringComposition, 193
getAcronyms, 193
getAcronymsAsString, 193
getAcronymsWithReferencesAsHTMLString, 193
getComposition, 194
getFirstAcronymAsString, 194
getLosses, 194
getLossIDs, 194
getMass, 194
getName, 195
getReferences, 195
getReferencesAsString, 195
getResidueLossType, 195
getReverseComposition, 195
getSummary, 195
getSummaryMap, 196
isArtificial, 196
load, 196
setAcronyms, 196
setArtificial, 196
setComposition, 197
setLosses, 197
setLossIDs, 197
setMass, 197
setName, 198
setReferences, 198
setResidueLossType, 198
setSummary, 198
store, 198
cBrick.h
getNumberOfBricks, 477
cBricksDatabase, 199
calculateSummaryFromComposition, 200
checkKetideBlocks, 200
getAcronymName, 201
getAcronymNameOfTPeptide, 201
getMapOfNeutralLosses, 201
getMassOfComposition, 202
getMassOfNeutralLosses, 202
getRealName, 202
getSummaryFormulaOfNeutralLosses, 203
getTagName, 203
getTagNameOfTPeptide, 203
load, 203
loadFromPlainTextStream, 204
nextCombination, 204
nextCombinationFast, 205
nextCombinationFastLimited, 205
operator[], 206
push_back, 206
replaceAcronymsByIDs, 206
size, 207
store, 207
storeToPlainTextStream, 207
cBricksDatabase.h
compareBrickMasses, 480
generateBricksPermutations, 481
getNameOfCompositionFromIntVector, 481
getReversedNameOfCompositionFromStringVector, 481
cBricksDatabaseProxyModel, 207
cBricksDatabaseProxyModel, 208
filterAcceptsRow, 208
initialize, 208
lessThan, 209
setWholeWord, 209
cBricksDatabaseWidget, 210
applyGlobalPreferences, 210
cBricksDatabaseWidget, 210
closeEvent, 211
keyPressEvent, 211
cBricksDatabaseWidget.h
numberOfOccurrences, 587
cCalibrationChart, 211
applyGlobalPreferences, 212
cCalibrationChart, 212
closeEvent, 213
createTable, 213
emitXValueInterval, 213
keyPressEvent, 214
setData, 214
setLineParameters, 214
cCalibrationChartProxyModel, 215
cCalibrationChartProxyModel, 215
lessThan, 216
cCalibrationChartScene, 216
cCalibrationChartScene, 217
exportToPDF, 217
exportToPNG, 218
exportToSVG, 218
getMaximumCollectionTime, 218
getMaximumConcentration, 218
hideLabels, 219
mouseMoveEvent, 219
mousePressEvent, 219
mouseReleaseEvent, 219
resizeEvent, 219
setData, 220
setGraphType, 220
setLineParameters, 220
showEvent, 220
updateXValueInterval, 221
wheelEvent, 221
cCalibrationChartScene.h
calculateLOD, 593
calculateLOQ, 593
calculateR2Value, 594
cCandidate, 221
calculatePrecursorMassFromBricks, 225
calculateSummaryFormulaFromBricks, 225
cCandidate, 224
checkKetideBlocks, 226
checkKetideSequence, 226

compare, 226
getAcronymPeptideNameWithHTMLReferences, 227
getAcronyms, 227
getAcronymsTComposition, 227
getBackboneAcronyms, 227
getBranchAcronyms, 228
getBranchCyclicRotations, 228
getBranchEnd, 228
getBranchSize, 228
getBranchStart, 228
getComposition, 229
getEndModifID, 229
getLeftResidueType, 229
getMiddleModifID, 229
getName, 230
getPath, 230
getPathAsString, 230
getPermutations, 230
getPermutationsOfBranches, 230
getRealNameTComposition, 231
getRealPeptideName, 231
getRevertedTComposition, 231
getRightResidueType, 231
getRotations, 232
getStartIonType, 232
getStartModifID, 232
getSummaryFormula, 232
getTComposition, 232
hasEqualTPermutations, 233
hasFirstBrickArtificial, 233
hasLastBrickArtificial, 233
hasLastBrickInvalid, 233
hasOnlyArtificialBricks, 234
isEqualTo, 234
load, 234
prepareBranchedCandidates, 235
setAcronymPeptideNameWithHTMLReferences, 235
setAcronyms, 235
setBackboneAcronyms, 235
setBranchAcronyms, 236
setCandidate, 236
setEndModifID, 237
setMiddleModifID, 237
setName, 237
setPath, 237
setRealPeptideName, 238
setStartIonType, 238
setStartModifID, 238
setSummaryFormula, 238
store, 238
cCandidate.h
operator==, 484
cCandidateSet, 239
getSet, 239
size, 239
cCheckBoxDelegate, 240
cCheckBoxDelegate, 240
editorEvent, 241
paint, 241
cChromatogramWindow, 241
applyGlobalPreferences, 243
calculateAvgSpectrum, 243
cChromatogramWindow, 242
closeEvent, 243
doubleClickedScanIDSignal, 243
emitRetentionTimeInterval, 243
emitScanIDInterval, 244
keyPressEvent, 244
recalculateTICChromatogram, 244
setAbsoluteIntensityEnabled, 244
setPeakShape, 245
cChromatogramWindowWidget, 245
cChromatogramWindowWidget, 246
chromatogramWidgetDoubleClicked, 246
exportToPDF, 246
exportToPNG, 247
exportToSVG, 247
mouseDoubleClickEvent, 247
mouseMoveEvent, 247
mousePressEvent, 248
mouseReleaseEvent, 248
recalculateTICChromatogram, 248
resizeEvent, 248
showEvent, 248
updateRetentionTimeInterval, 249
updateScanIDInterval, 249
wheelEvent, 249
cChromatogramWindowWidget.h
calculateBaseTau, 601
calculateCaruana, 602
calculateExponentialParameters, 602
calculateGaussianParameters, 603
calculateGuo, 603
calculateTauUsingWeightedAverage, 604
computeExponentialFunction, 604
computeExponentiallyModifiedGaussFunction, 604
erfcx, 605
erfcxinv, 605
getExponentiallyModifiedGaussianArea, 605
getExponentiallyModifiedGaussianAreas, 606
getGaussianArea, 606
getGaussianAreas, 607
cComboBoxDelegate, 249
cComboBoxDelegate, 250
createEditor, 250
paint, 250
setEditorData, 251
setModelData, 251
cCoordinateInfo, 251
cCyclicWidget, 252
exportToPDF, 253
exportToPNG, 253
exportToSVG, 253
hidePeakLabels, 253

initialize, 253
paintEvent, 254
setReportIsomers, 254
cCyclicWidget.h
generateCyclicLabelsToLeft, 612
generateCyclicLabelsToRight, 613
paintCircle, 614
cDefaultNeutralLosses, 254
getNeutralLosses, 255
cDefineCalibrationWidget, 255
closeEvent, 256
getData, 256
keyPressEvent, 257
load, 257
prepareWidget, 257
store, 257
cDeNovoGraph, 258
connectEdgesFormingPathsNotFinishingInPrecursor, 258
connectEdgesFormingPathsNotStartingFromFirstNode, 259
createGraph, 259
getBrickDatabaseWithCombinations, 259
getNumberOfEdges, 260
initialize, 260
operator[], 260
printGraph, 260
printPaths, 261
removeEdgesFormingPathsNotFinishingInPrecursor, 261
removeEdgesFormingPathsNotStartingFromFirstNode, 261
removePathsWhichCanBeSubstitutedByLongerPath, 261
startGraphReader, 262
cDeNovoGraph.h
compareNodes, 489
cDeNovoGraphNode, 262
addIonAnnotation, 263
checkIonAnnotation, 263
getIntensity, 264
getMZRatio, 264
insertEdge, 264
insertTempEdge, 264
operator[], 264
removeEdge, 265
setIntensity, 265
setMZRatio, 265
size, 265
cDeNovoGraphNode.h
compareEdgesByPPMError, 492
compareEdgesByTargetNodeIntensityDesc, 492
operator==, 492, 493
cDrawPeptideWidget, 266
cDrawPeptideWidget, 266
closeEvent, 267
keyPressEvent, 267
sendInsert, 267
sendSequence, 267
sendTag, 267
setSequence, 268
cEdge, 268
cEnvelopeSummaryTableKeyMS_comp, 269
operator(), 269
cEventFilter, 270
eventFilter, 270
cExportDialog, 270
cExportDialog, 271
cFindDialog, 271
cFindDialog, 272
cFormulaPredictor, 272
cFormulaPredictor, 273
closeEvent, 273
keyPressEvent, 273
cFormulaPredictorRowItem, 273
cFormulaPredictorThread, 274
initialize, 274
cFragmentIons, 275
cFragmentIons, 276
load, 276
operator[], 276
recalculateFragments, 277
store, 277
cFragmentIons.h
charge, 510
initializeFragmentIonsForDeNovoGraphOfBranchCyclicPeptides, 511
initializeFragmentIonsForDeNovoGraphOfCyclicPeptides, 511
initializeFragmentIonsForDeNovoGraphOfCyclicPolyketide, 511
initializeFragmentIonsForDeNovoGraphOfLinearPeptides, 511
initializeFragmentIonsForDeNovoGraphOfLinearPolyketide, 512
initializeFragmentIonsForDeNovoGraphOfTPeptides, 512
loadModificationsFromPlainTextStream, 512
storeModificationsToPlainTextStream, 512
uncharge, 513
cFragmentIonsListWidget, 277
cFragmentIonsListWidget, 278
getList, 278
cFragmentIonType, 278
load, 279
store, 279
cGlobalPreferences, 279
cGraphReaderThread, 281
initialize, 281
cGraphWidget, 282
closeEvent, 283
getHTML, 283
keyPressEvent, 283
load, 283
setHTML, 283
store, 284

updateView, 284
charge
cFragmentIons.h, 510
checkAdvancedFilteringRules
cParameters.h, 533
checkAndPrepare
cParameters, 341
checkFile
utilities.h, 564
checkIonAnnotation
cDeNovoGraphNode, 263
checkKetideBlocks
cBricksDatabase, 200
cCandidate, 226
checkKetideSequence
cCandidate, 226
checkModifications
cParameters, 341
checkRegex
utilities.h, 564
checkSeniorRules
cParameters.h, 534
chromatogramWidgetDoubleClicked
cChromatogramWindowWidget, 246
cHTMLDelegate, 284
cHTMLDelegate, 285
paint, 285
cHTMLExportDialog, 285
cHTMLExportDialog, 286
cImageWindow, 286
activeLayerChanged, 288
addLayer, 288
applyGlobalPreferences, 288
cImageWindow, 287
closeEvent, 288
doubleClickedSpectrumIDSignal, 288
keyPressEvent, 289
keyReleaseEvent, 289
layerChanged, 289
setDefaultValues, 289
updateSummaryPeaksTableFilter, 290
cImageWindowImportDialog, 290
cImageWindowImportDialog, 291
getFormValues, 291
setFormValues, 292
cImageWindowTabElectron, 292
cImageWindowTabElectron, 292
getFormValues, 293
setFormValues, 293
cImageWindowTabOptical, 293
cImageWindowTabOptical, 294
getFormValues, 294
setFormValues, 294
cImageWindowTabOptical.h
parseLifMetadata, 640
parseLifMetadataSubTree, 640
readLifMetadata, 640
cImageWindowWidget, 295
getImage, 296
goToMicroscopyPosition, 297
imageWidgetDoubleClicked, 297
mouseDoubleClickEvent, 297
mouseMoveEvent, 297
mousePressEvent, 297
mouseReleaseEvent, 298
resizeEvent, 298
setCorrelationValues, 298
setDefaultMaxXY, 298
setFilterOptions, 299
setHistologyImage, 299
setHistologyPosition, 300
setKeepAspectRatio, 300
setMicroscopyImage, 300
setMicroscopyPosition, 300
setNavigationLayer, 301
setOpticalImage, 301
setSelectedRegion, 301
updateCorrelation, 302
updateFilter, 302
updateHistologyPosition, 302
updateMicroscopyPosition, 303
updateRuler, 303
wheelEvent, 303
cImzML, 304
getItems, 304
hasProfileSpectra, 304
parse, 305
updateRawDataToPeakList, 305
use64BitIntensityPrecision, 305
use64BitMzPrecision, 306
write, 306
cImzMLItem, 306
cIonLabel, 307
cIonSummaryTableKeyMS_comp, 307
operator(), 307
cIonType, 308
load, 309
store, 309
cIsotopeMap, 309
count, 309
operator[], 310
cIsotopePatternCache, 310
cIsotopePatternCache, 311
get, 311
put, 311
clear
cTheoreticalSpectrum, 443
cTheoreticalSpectrumList, 460
cLifMetadata, 312
cLinearWidget, 313
exportToPDF, 314
exportToPNG, 314
exportToSVG, 314
hidePeakLabels, 314
initialize, 314
paintEvent, 315

setReportIsomers, 315
cLinearWidget.h
drawLabels, 649
insertLabel, 649
operator==, 650
closeEvent
cAboutWidget, 186
cBricksDatabaseWidget, 211
cCalibrationChart, 213
cChromatogramWindow, 243
cDefineCalibrationWidget, 256
cDrawPeptideWidget, 267
cFormulaPredictor, 273
cGraphWidget, 283
cImageWindow, 288
cModificationsWidget, 325
cMultipleDatasetsTableWidget, 331
cParametersWidget, 345
cPatternSimulatorWidget, 352
cPubChemSearchWidget, 389
cSequenceDatabaseWidget, 403
cSpectrumDetailWidget, 412
cSummaryPeaksTableWidget, 434
cMainThread, 315
addToParallelOutputState, 317
cMainThread, 316
enableButtonsHandlingResults, 317
enableRunButtonAndSettings, 317
enableStopButton, 317
isGraphReaderWorking, 318
message, 318
sendParameters, 318
setGraph, 318
setGraphReaderIsWorking, 318
cMainWindow, 319
keyPressEvent, 319
cMainWindowProxyModel, 320
cMainWindowProxyModel, 320
filterAcceptsRow, 321
initialize, 321
lessThan, 321
setWholeWord, 322
cModificationsProxyModel, 322
cModificationsProxyModel, 323
filterAcceptsRow, 323
initialize, 323
lessThan, 324
setWholeWord, 324
cModificationsWidget, 324
applyGlobalPreferences, 325
closeEvent, 325
cModificationsWidget, 325
keyPressEvent, 326
cMultipleButtonDelegate, 326
cMultipleButtonDelegate, 326
displayText, 327
editorEvent, 327
paint, 327
cMultipleDatasetsTableProxyModel, 328
cMultipleDatasetsTableProxyModel, 328
lessThan, 328
cMultipleDatasetsTableWidget, 329
applyGlobalPreferences, 330
closeEvent, 331
cMultipleDatasetsTableWidget, 330
getCalibrationChartWidget, 331
getDefineCalibrationWidget, 331
getPubChemSearchWidget, 331
keyPressEvent, 331
multipleDatasetsTableRowClicked, 331
prepareToShow, 332
cMultipleDatasetsTableWidget.h
calculateLineParameterA, 666
calculateLineParameters, 667
calculateLineParametersWithIterations, 667
cMzML, 332
hasProfileSpectra, 333
parse, 333
cMzML.h
compareElementTagName, 531
getAttribute, 531
setAttribute, 531
cNeutralLossesListWidget, 334
addItem, 334
cNeutralLossesListWidget, 334
getList, 335
compare
cCandidate, 226
compareAverageMSSpectrum
cTheoreticalSpectrum, 443
compareBandAllIonsDesc
cSpectrumComparatorThreadMS2.h, 721
compareBranchCyclic
cTheoreticalSpectrum, 444
compareBranched
cTheoreticalSpectrum, 444
compareBrickMasses
cBricksDatabase.h, 480
compareCosineSimilarityDesc
cSpectrumComparatorThreadMS2.h, 721
compareCyclic
cTheoreticalSpectrum, 445
compareCyclicPolyketide
cTheoreticalSpectrum, 445
compareDoubles
utilities.h, 565
compareEdgesByPPMError
cDeNovoGraphNode.h, 492
compareEdgesByTargetNodeIntensityDesc
cDeNovoGraphNode.h, 492
compareElementTagName
cMzML.h, 531
compareLinear
cTheoreticalSpectrum, 446
compareLinearPolyketide
cTheoreticalSpectrum, 446

compareMapsIntDouble
utilities.h, 565
compareMSSpectrum
cTheoreticalSpectrum, 447
compareNodes
cDeNovoGraph.h, 489
compareNumberOfMatchedPeaksDesc
cSpectrumComparatorThreadMS2.h, 722
compareOther
cTheoreticalSpectrum, 447
comparePaths
cSpectrumComparatorThreadMS2.h, 722
compareRelativePeakIntensitiesDesc
cPeaksList.h, 543
compareStringBySize, 335
operator(), 335
compareStringVectors
utilities.h, 566
compareSumOfRelIntDesc
cSpectrumComparatorThreadMS2.h, 722
compareWeightedRatioDesc
cSpectrumComparatorThreadMS2.h, 723
compareYandAllIonsDesc
cSpectrumComparatorThreadMS2.h, 723
compareYBandAllIonsDesc
cSpectrumComparatorThreadMS2.h, 723
computeExponentialFunction
cChromatogramWindowWidget.h, 604
computeExponentiallyModifiedGaussFunction
cChromatogramWindowWidget.h, 604
computeGaussFunction
cPatternSimulatorSceneWidget.h, 677
computeGaussHeight
cPatternSimulatorSceneWidget.h, 677
connectEdgesFormingPathsNotFinishingInPrecursor
cDeNovoGraph, 258
connectEdgesFormingPathsNotStartingFromFirstNode
cDeNovoGraph, 259
convertStringIntUnorderedMapToStringVector
utilities.h, 566
core/cBrick.h, 477, 478
core/cBricksDatabase.h, 480, 481
core/cCandidate.h, 483, 484
core/cCandidateSet.h, 488
core/cDeNovoGraph.h, 489, 490
core/cDeNovoGraphNode.h, 491, 493
core/cFragmentIons.h, 495, 513
core/cGlobalPreferences.h, 525, 526
core/cImzML.h, 527, 528
core/cIsotopePatternCache.h, 529
core/cMzML.h, 530, 532
core/cParameters.h, 532, 534
core/cPeak.h, 540
core/cPeakListSeries.h, 541, 542
core/cPeaksList.h, 543, 545
core/cSequence.h, 548
core/cSequenceDatabase.h, 549, 550
core/cSummaryFormula.h, 551, 552
core/cTheoreticalSpectrum.h, 553, 554
core/cTheoreticalSpectrumList.h, 560
core/utilities.h, 561, 576
count
cIsotopeMap, 309
cPeriodicTableMap, 376
cParameters, 336
calculateNeutralLosses, 341
checkAndPrepare, 341
checkModifications, 341
generateCompounds, 342
load, 342
prepareLossesAndCompounds, 342
printToString, 343
reportunmatchedtheoreticalpeaks, 343
setOutputStream, 343
store, 343
cParameters.h
checkAdvancedFilteringRules, 533
checkSeniorRules, 534
searchHintForDeNovo, 534
cParametersWidget, 344
applyGlobalPreferences, 345
closeEvent, 345
cParametersWidget, 345
getParameters, 345
keyPressEvent, 346
sendSequenceLine, 346
setAndRestoreParameters, 346
setSequence, 346
setTag, 346
cPatternSimulatorSceneWidget, 347
cPatternSimulatorSceneWidget, 348
exportToPDF, 348
exportToPNG, 348
exportToSVG, 348
hidePeakLabels, 349
initialize, 349
mouseMoveEvent, 349
mousePressEvent, 349
mouseReleaseEvent, 350
resizeEvent, 350
showEvent, 350
updateMZInterval, 350
wheelEvent, 350
cPatternSimulatorSceneWidget.h
computeGaussFunction, 677
computeGaussHeight, 677
cPatternSimulatorWidget, 351
applyGlobalPreferences, 352
closeEvent, 352
cPatternSimulatorWidget, 351
emitMZInterval, 352
keyPressEvent, 352
rawDataStateChangedSignal, 353
cPeak, 353
cPeak, 355
empty, 355

equals, 355
load, 355
operator=, 355
store, 356
cPeakListSeries, 356
addPeakList, 357
cPeakListSeries, 357
load, 357
loadFromBAFStream, 358
loadFromIMZMLStream, 358
loadFromMGFStream, 359
loadFromMZMLStream, 359
loadFromPlainTextStream, 359
loadFromProfileApexStream, 360
loadSpotList, 360
operator=, 360
operator[], 360
resize, 361
size, 361
store, 361
cPeaksList, 361
add, 364
attach, 364
cPeaksList, 364
cropAbsoluteIntenzity, 365
cropMaximumMZRatio, 365
cropMinimumMZRatio, 365
cropRelativeIntenzity, 365
equals, 366
find, 366
getCoordinateX, 366
getCoordinateY, 366
getMaximumAbsoluteIntensity, 367
getMaximumAbsoluteIntensityFromMZInterval, 367
getMaximumMZRatio, 367
getMaximumRelativeIntensity, 367
getMaximumRelativeIntensityFromMZInterval, 368
getMinimumMZRatio, 368
getRetentionTime, 368
getRetentionTimeUnit, 368
getTitle, 369
load, 369
loadFromBAFStream, 369
loadFromIBDStream, 369
loadFromMGFStream, 370
loadFromPlainTextStream, 370
maxHighestPeaksInWindow, 370
normalizeIntenzity, 370
normalizeIntenzityByValue, 371
operator=, 371
operator[], 371
print, 371
reduceIsotopeFormulaDescriptions, 372
reducePeakDescriptions, 372
remove, 372
removeChargeVariants, 372
removeIsotopes, 372
removeNeutralLoss, 373
resize, 373
setCoordinates, 373
setIsotopeFlags, 373
setRetentionTime, 374
setRetentionTimeUnit, 374
setTitle, 374
size, 374
sortbyMass, 374
store, 375
storeToIBDStream, 375
cPeaksList.h
compareRelativePeakIntensitiesDesc, 543
isInPpmMassErrorTolerance, 544
ppmError, 544
searchHint, 544
cPeriodicTableMap, 375
count, 376
getAdductElements, 376
getElements, 376
operator[], 376
cPreferencesDialog, 377
cPreferencesDialog, 377
keyPressEvent, 378
resetPreferences, 378
setPreferences, 378
cPreferencesIonTypeDialog, 378
cPreferencesIonTypeDialog, 379
getValues, 379
setValues, 379
cPreferencesTabDataProcessing, 380
cPreferencesTabDataProcessing, 380
resetPreferences, 381
setPreferences, 381
cPreferencesTabDirectories, 381
cPreferencesTabDirectories, 382
resetPreferences, 382
setPreferences, 382
cPreferencesTabIonTypes, 382
cPreferencesTabIonTypes, 383
resetPreferences, 383
setPreferences, 383
cPreferencesTabURLs, 384
cPreferencesTabURLs, 384
resetPreferences, 385
setPreferences, 385
cPubchemCIDReader, 385
attachVectorOfCIDs, 386
getCIDList, 386
getCommaSeparatedCIDs, 386
getNumberOfCIDs, 386
operator(), 387
cPubchemNameReader, 387
getNameList, 387
getReducedNames, 388
operator(), 388
cPubChemSearchWidget, 388
closeEvent, 389

getHTML, 389
keyPressEvent, 389
load, 390
setHTML, 390
store, 390
createEditor
cComboBoxDelegate, 250
createGraph
cDeNovoGraph, 259
createTable
cCalibrationChart, 213
cropAbsoluteIntenzity
cPeaksList, 365
cropDecimalsByteArray
utilities.h, 566
cropMaximumMZRatio
cPeaksList, 365
cropMinimumMZRatio
cPeaksList, 365
cropPrecisionToSixDecimals
utilities.h, 566
cropPrecisionToSixDecimalsByteArray
utilities.h, 567
cropRelativeIntenzity
cPeaksList, 365
cRowItemInfo, 390
cSequence, 391
getBranchModification, 392
getCTterminalModification, 392
getMaximumRetentionTime, 392
getMinimumRetentionTime, 393
getName, 393
getNameWithReferenceAsHTMLString, 393
getNTterminalModification, 393
getPeptideType, 393
getReference, 394
getSequence, 394
getSummaryFormula, 394
isDecoy, 394
load, 394
setBranchModification, 395
setCTterminalModification, 395
setDecoy, 395
setName, 395
setNTterminalModification, 395
setPeptideType, 396
setReference, 396
setRetentionTime, 396
setSequence, 396
setSummaryFormula, 397
store, 397
cSequenceDatabase, 397
attachDecoys, 398
erase, 398
load, 398
loadFromPlainTextStream, 398
operator[], 399
push_back, 399
size, 399
store, 399
storeToPlainTextStream, 400
cSequenceDatabaseProxyModel, 400
cSequenceDatabaseProxyModel, 401
filterAcceptsRow, 401
initialize, 401
lessThan, 402
setWholeWord, 402
cSequenceDatabaseWidget, 402
applyGlobalPreferences, 403
closeEvent, 403
cSequenceDatabaseWidget, 403
importSequence, 404
keyPressEvent, 404
cSpectrumAnalyzerThreadMS1, 404
initialize, 405
cSpectrumComparatorThreadMS1, 405
initialize, 406
cSpectrumComparatorThreadMS2, 406
initialize, 407
cSpectrumComparatorThreadMS2.h
compareBandAllIonsDesc, 721
compareCosineSimilarityDesc, 721
compareNumberOfMatchedPeaksDesc, 722
comparePaths, 722
compareSumOfRelIntDesc, 722
compareWeightedRatioDesc, 723
compareYandAllIonsDesc, 723
compareYBandAllIonsDesc, 723
cSpectrumDetailProxyModel, 407
cSpectrumDetailProxyModel, 408
filterAcceptsRow, 408
lessThan, 408
setFlags, 409
cSpectrumDetailWidget, 409
absoluteIntensityStateChangedSignal, 412
closeEvent, 412
cSpectrumDetailWidget, 411
emitMZInterval, 412
exportImage, 412
findAll, 413
getDetailsAsHTMLString, 413
getLocalNeutralLosses, 413
getPartialPeaksTableAsHTMLString, 413
getPeaksTableAsHTMLString, 414
getTheoreticalSpectrum, 414
hasAbsoluteIntensityEnabled, 414
hasHideMatchedPeaksEnabled, 414
hasHidePeakLabelsEnabled, 414
hasHideScrambledPeaksEnabled, 415
hasHideUnmatchedPeaksEnabled, 415
hasMouseMzSelectionEnabled, 415
hasProfileSpectrumEnabled, 415
initialize, 415
keyPressEvent, 416
lastActiveDetail, 416
moveEvent, 416

operator=, 416
prepareToShow, 417
rawDataStateChangedSignal, 417
setAbsoluteIntensityEnabled, 417
setHideMatchedPeaksEnabled, 417
setHidePeakLabelsEnabled, 418
setHideScrambledPeaksEnabled, 418
setHideUnmatchedPeaksEnabled, 418
setMouseMzSelectionEnabled, 418
setProfileSpectrumEnabled, 419
zoomToPeak, 419
cSpectrumGeneratorThreadMS1, 419
initialize, 420
cSpectrumSceneWidget, 420
cSpectrumSceneWidget, 421
exportToPDF, 421
exportToPNG, 422
exportToSVG, 422
hideMatchedPeaks, 422
hidePeakLabels, 422
hideScrambledPeaks, 423
hideUnmatchedPeaks, 423
initialize, 423
mouseMoveEvent, 423
mousePressEvent, 423
mouseReleaseEvent, 424
resizeEvent, 424
showEvent, 424
updateMZInterval, 424
wheelEvent, 425
cSummaryFormula, 425
addFormula, 426
cSummaryFormula, 426
equals, 426
getFancySummary, 427
getIsotopePattern, 427
getMass, 427
getSummary, 428
hasAllElementsPositive, 428
isEmpty, 428
isPartial, 428
isValid, 428
load, 429
setFormula, 429
setFromMap, 429
setPartial, 429
store, 429
cSummaryFormula.h
addStringFormulaToMap, 551
mergeMaps, 551
rechargeMap, 552
reduceMapToStringFormula, 552
cSummaryPeaksTableProxyModel, 430
cSummaryPeaksTableProxyModel, 431
filterAcceptsRow, 431
initialize, 431
lessThan, 432
setRegion, 432
setWholeWord, 433
cSummaryPeaksTableWidget, 433
applyGlobalPreferences, 434
closeEvent, 434
cSummaryPeaksTableWidget, 434
filterCompound, 435
getPubChemSearchWidget, 435
keyPressEvent, 435
prepareToShow, 435
sendFilterOptionsToChromatogram, 436
sendFilterOptionsToImageWindow, 436
summaryPeaksTableRowDoubleClicked, 437
updateFilterBySelectedRegion, 437
cSummaryPeaksTableWidget.h
addEICPeak, 710
cSummaryTableKeyMS, 437
cSummaryTableKeyMSMS, 438
cSummaryTableKeyMSMS_comp, 439
operator(), 439
cTablePosition, 439
cTheoreticalSpectrum, 440
attach, 443
clear, 443
compareAverageMSSpectrum, 443
compareBranchCyclic, 444
compareBranched, 444
compareCyclic, 445
compareCyclicPolyketide, 445
compareLinear, 446
compareLinearPolyketide, 446
compareMSSpectrum, 447
compareOther, 447
cTheoreticalSpectrum, 443
equals, 448
finalizeMSSpectrum, 448
generateCTerminalFragmentIons, 448
generateFineMSSpectrum, 449
generateMSSpectrum, 449
generateNTerminalFragmentIons, 450
getCandidate, 450
getCosineSimilarity, 451
getCoverageBySeries, 451
getDecoyScores, 451
getExperimentalMatches, 451
getExperimentalSpectrum, 451
getHintsIndex, 452
getLabelsOfRotations, 452
getNumberOfCompletedSeries, 452
getNumberOfMatchedPeaks, 452, 453
getNumberOfMatchedPeaksB, 453
getNumberOfMatchedPeaksY, 453
getNumberOfMatchedPeaksYB, 453
getNumberOfPeaks, 454
getNumberOfScrambledPeaks, 454
getPathId, 454
getRatioOfMatchedPeaks, 454
getReverseValidPosition, 454
getSumOfRelativeIntensities, 454

getTargetPatternFDR, 455
getTargetPatternScore, 455
getTargetScores, 455
getTheoreticalPeaks, 455
getUnmatchedExperimentalPeaksCount, 456
getValidPosition, 456
getWeightedRatioOfMatchedPeaks, 456
isValid, 456
load, 456
operator[], 457
resizePeakList, 457
setCandidate, 457
setExperimentalSpectrum, 457
setFDRs, 458
setNumberOfCompletedSeries, 458
setParameters, 458
setPathId, 458
setValidSequence, 459
sortByMass, 459
store, 459
cTheoreticalSpectrum.h
getNumberOfScoreHits, 554
cTheoreticalSpectrumList, 459
add, 460
clear, 460
get, 461
initialize, 461
parallelCompareAndStore, 461
resize, 461, 462
size, 462
sortAndFitSize, 462
updatekNNList, 463
cTileInfo, 463
cViewButtonDelegate, 464
cViewButtonDelegate, 464
displayText, 464
editorEvent, 465
paint, 465

displayText
cMultipleButtonDelegate, 327
cViewButtonDelegate, 464
doubleClickedScanIDSignal
cChromatogramWindow, 243
doubleClickedSpectrumIDSignal
cImageWindow, 288
drawLabels
cLinearWidget.h, 649

editorEvent
cCheckBoxDelegate, 241
cMultipleButtonDelegate, 327
cViewButtonDelegate, 465
emitMZInterval
cPatternSimulatorWidget, 352
cSpectrumDetailWidget, 412
emitRetentionTimeInterval
cChromatogramWindow, 243
emitScanIDInterval
cChromatogramWindow, 244
emitXValueInterval
cCalibrationChart, 213
empty
cBrick, 192
cPeak, 355
enableButtonsHandlingResults
cMainThread, 317
enableRunButtonAndSettings
cMainThread, 317
enableStopButton
cMainThread, 317
endsWithBracket
TRotationInfo, 475
equals
cPeak, 355
cPeaksList, 366
cSummaryFormula, 426
cTheoreticalSpectrum, 448
erase
cSequenceDatabase, 398
erfcx
cChromatogramWindowWidget.h, 605
erfcxinv
cChromatogramWindowWidget.h, 605
eventFilter
cEventFilter, 270
explodeToIntComposition
cBrick, 193
explodeToStringComposition
cBrick, 193
exportImage
cSpectrumDetailWidget, 412
exportToPDF
cBranchCyclicWidget, 187
cBranchedWidget, 189
cCalibrationChartScene, 217
cChromatogramWindowWidget, 246
cCyclicWidget, 253
cLinearWidget, 314
cPatternSimulatorSceneWidget, 348
cSpectrumSceneWidget, 421
exportToPNG
cBranchCyclicWidget, 187
cBranchedWidget, 189
cCalibrationChartScene, 218
cChromatogramWindowWidget, 247
cCyclicWidget, 253
cLinearWidget, 314
cPatternSimulatorSceneWidget, 348
cSpectrumSceneWidget, 422
exportToSVG
cBranchCyclicWidget, 187
cBranchedWidget, 190
cCalibrationChartScene, 218
cChromatogramWindowWidget, 247
cCyclicWidget, 253
cLinearWidget, 314

cPatternSimulatorSceneWidget, 348
cSpectrumSceneWidget, 422

fact
utilities.h, 567
filterAcceptsRow
cBricksDatabaseProxyModel, 208
cMainWindowProxyModel, 321
cModificationsProxyModel, 323
cSequenceDatabaseProxyModel, 401
cSpectrumDetailProxyModel, 408
cSummaryPeaksTableProxyModel, 431
filterCompound
cSummaryPeaksTableWidget, 435
finalizeMSSpectrum
cTheoreticalSpectrum, 448
find
cPeaksList, 366
findAll
cSpectrumDetailWidget, 413
fixLabelCharge
utilities.h, 567

generateBranchLabelsDown
cBranchCyclicWidget.h, 581
generateBranchLabelsUp
cBranchCyclicWidget.h, 582
generateBricksPermutations
cBricksDatabase.h, 481
generateCompounds
cParameters, 342
generateCTerminalFragmentIons
cTheoreticalSpectrum, 448
generateCyclicLabelsToLeft
cCyclicWidget.h, 612
generateCyclicLabelsToRight
cCyclicWidget.h, 613
generateFineMSSpectrum
cTheoreticalSpectrum, 449
generateMSSpectrum
cTheoreticalSpectrum, 449
generateNTerminalFragmentIons
cTheoreticalSpectrum, 450
get
cIsotopePatternCache, 311
cTheoreticalSpectrumList, 461
getAcronymName
cBricksDatabase, 201
getAcronymNameOfTPeptide
cBricksDatabase, 201
getAcronymPeptideNameWithHTMLReferences
cCandidate, 227
getAcronyms
cBrick, 193
cCandidate, 227
getAcronymsAsString
cBrick, 193
getAcronymsTComposition
cCandidate, 227
getAcronymsWithReferencesAsHTMLString
cBrick, 193
getAdductElements
cPeriodicTableMap, 376
getAttribute
cMzML.h, 531
getBackboneAcronyms
cCandidate, 227
getBranchAcronyms
cCandidate, 228
getBranchCyclicRotations
cCandidate, 228
getBranchEnd
cCandidate, 228
getBranchModification
cSequence, 392
getBranchSize
cCandidate, 228
getBranchStart
cCandidate, 228
getBrickDatabaseWithCombinations
cDeNovoGraph, 259
getCalibrationChartWidget
cMultipleDatasetsTableWidget, 331
getCandidate
cTheoreticalSpectrum, 450
getCIDList
cPubchemCIDReader, 386
getCommaSeparatedCIDs
cPubchemCIDReader, 386
getComposition
cBrick, 194
cCandidate, 229
getCoordinateX
cPeaksList, 366
getCoordinateY
cPeaksList, 366
getCosineSimilarity
cTheoreticalSpectrum, 451
getCoverageBySeries
cTheoreticalSpectrum, 451
getCTterminalModification
cSequence, 392
getData
cDefineCalibrationWidget, 256
getDecoyScores
cTheoreticalSpectrum, 451
getDefineCalibrationWidget
cMultipleDatasetsTableWidget, 331
getDetailsAsHTMLString
cSpectrumDetailWidget, 413
getElements
cPeriodicTableMap, 376
getEndModifID
cCandidate, 229
getExperimentalMatches
cTheoreticalSpectrum, 451
getExperimentalSpectrum

cTheoreticalSpectrum, 451
getExponentiallyModifiedGaussianArea
cChromatogramWindowWidget.h, 605
getExponentiallyModifiedGaussianAreas
cChromatogramWindowWidget.h, 606
getFancySummary
cSummaryFormula, 427
getFirstAcronymAsString
cBrick, 194
getFormValues
cImageWindowImportDialog, 291
cImageWindowTabElectron, 293
cImageWindowTabOptical, 294
getGaussianArea
cChromatogramWindowWidget.h, 606
getGaussianAreas
cChromatogramWindowWidget.h, 607
getHintsIndex
cTheoreticalSpectrum, 452
getHTML
cGraphWidget, 283
cPubChemSearchWidget, 389
getImage
cImageWindowWidget, 296
getIntensity
cDeNovoGraphNode, 264
getIsotopePattern
cSummaryFormula, 427
getItems
cImzML, 304
getLabelsOfRotations
cTheoreticalSpectrum, 452
getLeftResidueType
cCandidate, 229
getList
cFragmentIonsListWidget, 278
cNeutralLossesListWidget, 335
getLocalNeutralLosses
cSpectrumDetailWidget, 413
getLosses
cBrick, 194
getLossIDs
cBrick, 194
getMapOfNeutralLosses
cBricksDatabase, 201
getMass
cBrick, 194
cSummaryFormula, 427
getMassOfComposition
cBricksDatabase, 202
getMassOfNeutralLosses
cBricksDatabase, 202
getMaximumAbsoluteIntensity
cPeaksList, 367
getMaximumAbsoluteIntensityFromMZInterval
cPeaksList, 367
getMaximumCollectionTime
cCalibrationChartScene, 218
getMaximumConcentration
cCalibrationChartScene, 218
getMaximumMZRatio
cPeaksList, 367
getMaximumRelativeIntensity
cPeaksList, 367
getMaximumRelativeIntensityFromMZInterval
cPeaksList, 368
getMaximumRetentionTime
cSequence, 392
getMiddleModifID
cCandidate, 229
getMinimumMZRatio
cPeaksList, 368
getMinimumRetentionTime
cSequence, 393
getMZRatio
cDeNovoGraphNode, 264
getName
cBrick, 195
cCandidate, 230
cSequence, 393
getNameList
cPubchemNameReader, 387
getNameOfCompositionFromIntVector
cBricksDatabase.h, 481
getNameWithReferenceAsHTMLString
cSequence, 393
getNeutralLosses
cDefaultNeutralLosses, 255
getNTterminalModification
cSequence, 393
getNumberOfBricks
cBrick.h, 477
getNumberOfCIDs
cPubchemCIDReader, 386
getNumberOfCompletedSeries
cTheoreticalSpectrum, 452
getNumberOfEdges
cDeNovoGraph, 260
getNumberOfMatchedPeaks
cTheoreticalSpectrum, 452, 453
getNumberOfMatchedPeaksB
cTheoreticalSpectrum, 453
getNumberOfMatchedPeaksY
cTheoreticalSpectrum, 453
getNumberOfMatchedPeaksYB
cTheoreticalSpectrum, 453
getNumberOfPeaks
cTheoreticalSpectrum, 454
getNumberOfScoreHits
cTheoreticalSpectrum.h, 554
getNumberOfScrambledPeaks
cTheoreticalSpectrum, 454
getParameters
cParametersWidget, 345
getPartialPeaksTableAsHTMLString
cSpectrumDetailWidget, 413

getPath
cCandidate, 230
getPathAsString
cCandidate, 230
getPathId
cTheoreticalSpectrum, 454
getPeaksTableAsHTMLString
cSpectrumDetailWidget, 414
getPeptideType
cSequence, 393
getPeptideTypeFromString
utilities.h, 568
getPermutations
cCandidate, 230
getPermutationsOfBranches
cCandidate, 230
getPubChemSearchWidget
cMultipleDatasetsTableWidget, 331
cSummaryPeaksTableWidget, 435
getRatioOfMatchedPeaks
cTheoreticalSpectrum, 454
getRealName
cBricksDatabase, 202
getRealNameTComposition
cCandidate, 231
getRealPeptideName
cCandidate, 231
getReducedNames
cPubchemNameReader, 388
getReference
cSequence, 394
getReferences
cBrick, 195
getReferencesAsString
cBrick, 195
getResidueLossType
cBrick, 195
getRetentionTime
cPeaksList, 368
getRetentionTimeUnit
cPeaksList, 368
getReverseComposition
cBrick, 195
getReversedNameOfCompositionFromStringVector
cBricksDatabase.h, 481
getReverseValidPosition
cTheoreticalSpectrum, 454
getRevertedTComposition
cCandidate, 231
getRightResidueType
cCandidate, 231
getRotations
cCandidate, 232
getSequence
cSequence, 394
getSet
cCandidateSet, 239
getShortFileName
utilities.h, 568
getStartIonType
cCandidate, 232
getStartModifID
cCandidate, 232
getStringFromPeptideType
utilities.h, 568
getSummary
cBrick, 195
cSummaryFormula, 428
getSummaryFormula
cCandidate, 232
cSequence, 394
getSummaryFormulaOfNeutralLosses
cBricksDatabase, 203
getSummaryMap
cBrick, 196
getSumOfRelativeIntensities
cTheoreticalSpectrum, 454
getTagName
cBricksDatabase, 203
getTagNameOfTPeptide
cBricksDatabase, 203
getTargetPatternFDR
cTheoreticalSpectrum, 455
getTargetPatternScore
cTheoreticalSpectrum, 455
getTargetScores
cTheoreticalSpectrum, 455
getTComposition
cCandidate, 232
getTheoreticalPeaks
cTheoreticalSpectrum, 455
getTheoreticalSpectrum
cSpectrumDetailWidget, 414
getTitle
cPeaksList, 369
getUnmatchedExperimentalPeaksCount
cTheoreticalSpectrum, 456
getValidPosition
cTheoreticalSpectrum, 456
getValues
cPreferencesIonTypeDialog, 379
getWeightedRatioOfMatchedPeaks
cTheoreticalSpectrum, 456
goToMicroscopyPosition
cImageWindowWidget, 297
gui/cAboutWidget.h, 580
gui/cBranchCyclicWidget.h, 581, 583
gui/cBranchedWidget.h, 584
gui/cBricksDatabaseProxyModel.h, 585
gui/cBricksDatabaseWidget.h, 586, 587
gui/cCalibrationChart.h, 589, 590
gui/cCalibrationChartProxyModel.h, 592
gui/cCalibrationChartScene.h, 592, 594
gui/cCheckBoxDelegate.h, 597
gui/cChromatogramWindow.h, 598
gui/cChromatogramWindowWidget.h, 600, 607

gui/cComboBoxDelegate.h, 610, 611
gui/cCyclicWidget.h, 611, 615
gui/cDefineCalibrationWidget.h, 616
gui/cDrawPeptideWidget.h, 618
gui/cEventFilter.h, 620
gui/cExportDialog.h, 621
gui/cFindDialog.h, 621, 622
gui/cFormulaPredictor.h, 623
gui/cFragmentIonsListWidget.h, 625, 626
gui/cGraphWidget.h, 626, 627
gui/cHTMLDelegate.h, 628
gui/cHTMLExportDialog.h, 628, 629
gui/cImageWindow.h, 630
gui/cImageWindowImportDialog.h, 636, 637
gui/cImageWindowTabElectron.h, 638
gui/cImageWindowTabOptical.h, 639, 640
gui/cImageWindowWidget.h, 644
gui/cLinearWidget.h, 649, 650
gui/cMainThread.h, 652
gui/cMainWindow.h, 654, 655
gui/cMainWindowProxyModel.h, 659, 660
gui/cModificationsProxyModel.h, 660, 661
gui/cModificationsWidget.h, 661, 662
gui/cMultipleButtonDelegate.h, 664
gui/cMultipleDatasetsTableProxyModel.h, 665
gui/cMultipleDatasetsTableWidget.h, 666, 667
gui/cNeutralLossesListWidget.h, 670, 671
gui/cParametersWidget.h, 672
gui/cPatternSimulatorSceneWidget.h, 677, 678
gui/cPatternSimulatorWidget.h, 680, 681
gui/cPreferencesDialog.h, 683
gui/cPreferencesIonTypeDialog.h, 684, 685
gui/cPreferencesTabDataProcessing.h, 685, 686
gui/cPreferencesTabDirectories.h, 687
gui/cPreferencesTabIonTypes.h, 689
gui/cPreferencesTabURLs.h, 690, 691
gui/cPubChemSearchWidget.h, 692
gui/cSequenceDatabaseProxyModel.h, 695, 696
gui/cSequenceDatabaseWidget.h, 697
gui/cSpectrumDetailProxyModel.h, 699, 700
gui/cSpectrumDetailWidget.h, 700, 701
gui/cSpectrumSceneWidget.h, 705, 706
gui/cSummaryPeaksTableProxyModel.h, 708, 709
gui/cSummaryPeaksTableWidget.h, 710, 711
gui/cViewButtonDelegate.h, 715

hasAbsoluteIntensityEnabled
    cSpectrumDetailWidget, 414
hasAllElementsPositive
    cSummaryFormula, 428
hasEqualTPermutations
    cCandidate, 233
hasFirstBrickArtificial
    cCandidate, 233
hash_cCandidate, 466
    operator(), 466
hash_cEdge, 466
    operator(), 466
hash_cIonLabel, 467
    operator(), 467
hasHideMatchedPeaksEnabled
    cSpectrumDetailWidget, 414
hasHidePeakLabelsEnabled
    cSpectrumDetailWidget, 414
hasHideScrambledPeaksEnabled
    cSpectrumDetailWidget, 415
hasHideUnmatchedPeaksEnabled
    cSpectrumDetailWidget, 415
hasLastBrickArtificial
    cCandidate, 233
hasLastBrickInvalid
    cCandidate, 233
hasMouseMzSelectionEnabled
    cSpectrumDetailWidget, 415
hasOnlyArtificialBricks
    cCandidate, 234
hasProfileSpectra
    cImzML, 304
    cMzML, 333
hasProfileSpectrumEnabled
    cSpectrumDetailWidget, 415
hideLabels
    cCalibrationChartScene, 219
hideMatchedPeaks
    cSpectrumSceneWidget, 422
hidePeakLabels
    cBranchCyclicWidget, 188
    cBranchedWidget, 190
    cCyclicWidget, 253
    cLinearWidget, 314
    cPatternSimulatorSceneWidget, 349
    cSpectrumSceneWidget, 422
hideScrambledPeaks
    cSpectrumSceneWidget, 423
hideUnmatchedPeaks
    cSpectrumSceneWidget, 423
hintStructure, 468

imageWidgetDoubleClicked
    cImageWindowWidget, 297
importSequence
    cSequenceDatabaseWidget, 404
initialize
    cBranchCyclicWidget, 188
    cBranchedWidget, 190
    cBricksDatabaseProxyModel, 208
    cCyclicWidget, 253
    cDeNovoGraph, 260
    cFormulaPredictorThread, 274
    cGraphReaderThread, 281
    cLinearWidget, 314
    cMainWindowProxyModel, 321
    cModificationsProxyModel, 323
    cPatternSimulatorSceneWidget, 349
    cSequenceDatabaseProxyModel, 401
    cSpectrumAnalyzerThreadMS1, 405
    cSpectrumComparatorThreadMS1, 406
    cSpectrumComparatorThreadMS2, 407

cSpectrumDetailWidget, 415
cSpectrumGeneratorThreadMS1, 420
cSpectrumSceneWidget, 423
cSummaryPeaksTableProxyModel, 431
cTheoreticalSpectrumList, 461
initializeFragmentIonsForDeNovoGraphOfBranchCyclicPeptides
cFragmentIons.h, 511
initializeFragmentIonsForDeNovoGraphOfCyclicPeptides
cFragmentIons.h, 511
initializeFragmentIonsForDeNovoGraphOfCyclicPolyketide
cFragmentIons.h, 511
initializeFragmentIonsForDeNovoGraphOfLinearPeptides
cFragmentIons.h, 511
initializeFragmentIonsForDeNovoGraphOfLinearPolyketide
cFragmentIons.h, 512
initializeFragmentIonsForDeNovoGraphOfTPeptides
cFragmentIons.h, 512
insertEdge
cDeNovoGraphNode, 264
insertLabel
cLinearWidget.h, 649
insertTempEdge
cDeNovoGraphNode, 264
isArtificial
cBrick, 196
isCompatibleVersion
utilities.h, 569
isDecoy
cSequence, 394
isEmpty
cSummaryFormula, 428
isEqualTo
cCandidate, 234
isGraphReaderWorking
cMainThread, 318
isInPpmMassErrorTolerance
cPeaksList.h, 544
isotope, 468
isPartial
cSummaryFormula, 428
isValid
cSummaryFormula, 428
cTheoreticalSpectrum, 456
isWhiteSpace
utilities.h, 569
isWhiteSpaceExceptSpace
utilities.h, 569

keyPressEvent
cAboutWidget, 186
cBricksDatabaseWidget, 211
cCalibrationChart, 214
cChromatogramWindow, 244
cDefineCalibrationWidget, 257
cDrawPeptideWidget, 267
cFormulaPredictor, 273
cGraphWidget, 283
cImageWindow, 289
cMainWindow, 319
cModificationsWidget, 326
cMultipleDatasetsTableWidget, 331
cParametersWidget, 346
cPatternSimulatorWidget, 352
cPreferencesDialog, 378
cPubChemSearchWidget, 389
cSequenceDatabaseWidget, 404
cSpectrumDetailWidget, 416
cSummaryPeaksTableWidget, 435
keyReleaseEvent
cImageWindow, 289

lastActiveDetail
cSpectrumDetailWidget, 416
layerChanged
cImageWindow, 289
layerInfo, 468
lessThan
cBricksDatabaseProxyModel, 209
cCalibrationChartProxyModel, 216
cMainWindowProxyModel, 321
cModificationsProxyModel, 324
cMultipleDatasetsTableProxyModel, 328
cSequenceDatabaseProxyModel, 402
cSpectrumDetailProxyModel, 408
cSummaryPeaksTableProxyModel, 432
load
cBrick, 196
cBricksDatabase, 203
cCandidate, 234
cDefineCalibrationWidget, 257
cFragmentIons, 276
cFragmentIonType, 279
cGraphWidget, 283
cIonType, 309
cParameters, 342
cPeak, 355
cPeakListSeries, 357
cPeaksList, 369
cPubChemSearchWidget, 390
cSequence, 394
cSequenceDatabase, 398
cSummaryFormula, 429
cTheoreticalSpectrum, 456
matchedSeries, 470
neutralLoss, 472
nodeEdge, 473
loadDoubleVector
utilities.h, 570
loadFromBAFStream
cPeakListSeries, 358
cPeaksList, 369
loadFromIBDStream
cPeaksList, 369
loadFromIMZMLStream
cPeakListSeries, 358
loadFromMGFStream
cPeakListSeries, 359
cPeaksList, 370

loadFromMZMLStream
cPeakListSeries, 359
loadFromPlainTextStream
cBricksDatabase, 204
cPeakListSeries, 359
cPeaksList, 370
cSequenceDatabase, 398
loadFromProfileApexStream
cPeakListSeries, 360
loadIntStringMap
utilities.h, 570
loadIntVector
utilities.h, 570
loadModificationsFromPlainTextStream
cFragmentIons.h, 512
loadSpotList
cPeakListSeries, 360
loadString
utilities.h, 570
loadStringIntMap
utilities.h, 571
loadStringIntMapVector
utilities.h, 571
loadStringVector
utilities.h, 571

matchedSeries, 470
load, 470
store, 470
maxHighestPeaksInWindow
cPeaksList, 370
median
utilities.h, 571
mergeMaps
cSummaryFormula.h, 551
message
cMainThread, 318
microscopyLayerInfo, 471
mouseDoubleClickEvent
cChromatogramWindowWidget, 247
cImageWindowWidget, 297
mouseMoveEvent
cCalibrationChartScene, 219
cChromatogramWindowWidget, 247
cImageWindowWidget, 297
cPatternSimulatorSceneWidget, 349
cSpectrumSceneWidget, 423
mousePressEvent
cCalibrationChartScene, 219
cChromatogramWindowWidget, 248
cImageWindowWidget, 297
cPatternSimulatorSceneWidget, 349
cSpectrumSceneWidget, 423
mouseReleaseEvent
cCalibrationChartScene, 219
cChromatogramWindowWidget, 248
cImageWindowWidget, 298
cPatternSimulatorSceneWidget, 350
cSpectrumSceneWidget, 424
moveEvent
cSpectrumDetailWidget, 416
multipleDatasetsTableRowClicked
cMultipleDatasetsTableWidget, 331

neutralLoss, 472
load, 472
store, 473
nextCombination
cBricksDatabase, 204
nextCombinationFast
cBricksDatabase, 205
nextCombinationFastLimited
cBricksDatabase, 205
nodeEdge, 473
load, 473
store, 473
normalizeIntenzity
cPeaksList, 370
normalizeIntenzityByValue
cPeaksList, 371
numberOfOccurrences
cBricksDatabaseWidget.h, 587

operator()
cEnvelopeSummaryTableKeyMS_comp, 269
cIonSummaryTableKeyMS_comp, 307
compareStringBySize, 335
cPubchemCIDReader, 387
cPubchemNameReader, 388
cSummaryTableKeyMSMS_comp, 439
hash_cCandidate, 466
hash_cEdge, 466
hash_cIonLabel, 467
operator=
cPeak, 355
cPeakListSeries, 360
cPeaksList, 371
cSpectrumDetailWidget, 416
operator==
cCandidate.h, 484
cDeNovoGraphNode.h, 492, 493
cLinearWidget.h, 650
operator[]
cBricksDatabase, 206
cDeNovoGraph, 260
cDeNovoGraphNode, 264
cFragmentIons, 276
cIsotopeMap, 310
cPeakListSeries, 360
cPeaksList, 371
cPeriodicTableMap, 376
cSequenceDatabase, 399
cTheoreticalSpectrum, 457

paint
cCheckBoxDelegate, 241
cComboBoxDelegate, 250
cHTMLDelegate, 285

cMultipleButtonDelegate, 327
cViewButtonDelegate, 465
paintCircle
cCyclicWidget.h, 614
paintEvent
cBranchCyclicWidget, 188
cBranchedWidget, 190
cCyclicWidget, 254
cLinearWidget, 315
parallel/cFormulaPredictorThread.h, 716
parallel/cGraphReaderThread.h, 717, 718
parallel/cSpectrumAnalyzerThreadMS1.h, 718, 719
parallel/cSpectrumComparatorThreadMS1.h, 719, 720
parallel/cSpectrumComparatorThreadMS2.h, 720, 724
parallel/cSpectrumGeneratorThreadMS1.h, 725
parallelCompareAndStore
cTheoreticalSpectrumList, 461
parse
cImzML, 305
cMzML, 333
parseBranch
utilities.h, 572
parseLifMetadata
cImageWindowTabOptical.h, 640
parseLifMetadataSubTree
cImageWindowTabOptical.h, 640
ppmError
cPeaksList.h, 544
prepareBranchedCandidates
cCandidate, 235
prepareLossesAndCompounds
cParameters, 342
prepareToShow
cMultipleDatasetsTableWidget, 332
cSpectrumDetailWidget, 417
cSummaryPeaksTableWidget, 435
prepareWidget
cDefineCalibrationWidget, 257
print
cPeaksList, 371
printGraph
cDeNovoGraph, 260
printPaths
cDeNovoGraph, 261
printToString
cParameters, 343
proxyModelCheckDouble
utilities.h, 572
proxyModelCheckInt
utilities.h, 573
proxyModelCheckString
utilities.h, 573
push_back
cBricksDatabase, 206
cSequenceDatabase, 399
put
cIsotopePatternCache, 311

rawDataStateChangedSignal
cPatternSimulatorWidget, 353
cSpectrumDetailWidget, 417
readLifMetadata
cImageWindowTabOptical.h, 640
recalculateFragments
cFragmentIons, 277
recalculateTICChromatogram
cChromatogramWindow, 244
cChromatogramWindowWidget, 248
rechargeMap
cSummaryFormula.h, 552
reduceIsotopeFormulaDescriptions
cPeaksList, 372
reduceMapToStringFormula
cSummaryFormula.h, 552
reducePeakDescriptions
cPeaksList, 372
remove
cPeaksList, 372
removeChargeVariants
cPeaksList, 372
removeEdge
cDeNovoGraphNode, 265
removeEdgesFormingPathsNotFinishingInPrecursor
cDeNovoGraph, 261
removeEdgesFormingPathsNotStartingFromFirstNode
cDeNovoGraph, 261
removeIsotopes
cPeaksList, 372
removeNeutralLoss
cPeaksList, 373
removePathsWhichCanBeSubstitutedByLongerPath
cDeNovoGraph, 261
removeWhiteSpacesExceptSpaces
utilities.h, 574
replaceAcronymsByIDs
cBricksDatabase, 206
reportunmatchedtheoreticalpeaks
cParameters, 343
resetPreferences
cPreferencesDialog, 378
cPreferencesTabDataProcessing, 381
cPreferencesTabDirectories, 382
cPreferencesTabIonTypes, 383
cPreferencesTabURLs, 385
resize
cPeakListSeries, 361
cPeaksList, 373
cTheoreticalSpectrumList, 461, 462
resizeEvent
cCalibrationChartScene, 219
cChromatogramWindowWidget, 248
cImageWindowWidget, 298
cPatternSimulatorSceneWidget, 350
cSpectrumSceneWidget, 424
resizePeakList
cTheoreticalSpectrum, 457

searchHint

cPeaksList.h, 544
searchHintForDeNovo
cParameters.h, 534
sendFilterOptionsToChromatogram
cSummaryPeaksTableWidget, 436
sendFilterOptionsToImageWindow
cSummaryPeaksTableWidget, 436
sendInsert
cDrawPeptideWidget, 267
sendParameters
cMainThread, 318
sendSequence
cDrawPeptideWidget, 267
sendSequenceLine
cParametersWidget, 346
sendTag
cDrawPeptideWidget, 267
setAbsoluteIntensityEnabled
cChromatogramWindow, 244
cSpectrumDetailWidget, 417
setAcronymPeptideNameWithHTMLReferences
cCandidate, 235
setAcronyms
cBrick, 196
cCandidate, 235
setAndRestoreParameters
cParametersWidget, 346
setArtificial
cBrick, 196
setAttribute
cMzML.h, 531
setBackboneAcronyms
cCandidate, 235
setBranchAcronyms
cCandidate, 236
setBranchModification
cSequence, 395
setCandidate
cCandidate, 236
cTheoreticalSpectrum, 457
setComposition
cBrick, 197
setCoordinates
cPeaksList, 373
setCorrelationValues
cImageWindowWidget, 298
setCTterminalModification
cSequence, 395
setData
cCalibrationChart, 214
cCalibrationChartScene, 220
setDecoy
cSequence, 395
setDefaultMaxXY
cImageWindowWidget, 298
setDefaultValues
cImageWindow, 289
setEditorData
cComboBoxDelegate, 251
setEndModifID
cCandidate, 237
setExperimentalSpectrum
cTheoreticalSpectrum, 457
setFDRs
cTheoreticalSpectrum, 458
setFilterOptions
cImageWindowWidget, 299
setFlags
cSpectrumDetailProxyModel, 409
setFormula
cSummaryFormula, 429
setFormValues
cImageWindowImportDialog, 292
cImageWindowTabElectron, 293
cImageWindowTabOptical, 294
setFromMap
cSummaryFormula, 429
setGraph
cMainThread, 318
setGraphReaderIsWorking
cMainThread, 318
setGraphType
cCalibrationChartScene, 220
setHideMatchedPeaksEnabled
cSpectrumDetailWidget, 417
setHidePeakLabelsEnabled
cSpectrumDetailWidget, 418
setHideScrambledPeaksEnabled
cSpectrumDetailWidget, 418
setHideUnmatchedPeaksEnabled
cSpectrumDetailWidget, 418
setHistologyImage
cImageWindowWidget, 299
setHistologyPosition
cImageWindowWidget, 300
setHTML
cGraphWidget, 283
cPubChemSearchWidget, 390
setIntensity
cDeNovoGraphNode, 265
setIsotopeFlags
cPeaksList, 373
setKeepAspectRatio
cImageWindowWidget, 300
setLineParameters
cCalibrationChart, 214
cCalibrationChartScene, 220
setLosses
cBrick, 197
setLossIDs
cBrick, 197
setMass
cBrick, 197
setMicroscopyImage
cImageWindowWidget, 300
setMicroscopyPosition

cImageWindowWidget, 300
setMiddleModifID
cCandidate, 237
setModelData
cComboBoxDelegate, 251
setMouseMzSelectionEnabled
cSpectrumDetailWidget, 418
setMZRatio
cDeNovoGraphNode, 265
setName
cBrick, 198
cCandidate, 237
cSequence, 395
setNavigationLayer
cImageWindowWidget, 301
setNTterminalModification
cSequence, 395
setNumberOfCompletedSeries
cTheoreticalSpectrum, 458
setOpticalImage
cImageWindowWidget, 301
setOutputStream
cParameters, 343
setParameters
cTheoreticalSpectrum, 458
setPartial
cSummaryFormula, 429
setPath
cCandidate, 237
setPathId
cTheoreticalSpectrum, 458
setPeakShape
cChromatogramWindow, 245
setPeptideType
cSequence, 396
setPreferences
cPreferencesDialog, 378
cPreferencesTabDataProcessing, 381
cPreferencesTabDirectories, 382
cPreferencesTabIonTypes, 383
cPreferencesTabURLs, 385
setProfileSpectrumEnabled
cSpectrumDetailWidget, 419
setRealPeptideName
cCandidate, 238
setReference
cSequence, 396
setReferences
cBrick, 198
setRegion
cSummaryPeaksTableProxyModel, 432
setReportIsomers
cBranchCyclicWidget, 188
cBranchedWidget, 191
cCyclicWidget, 254
cLinearWidget, 315
setResidueLossType
cBrick, 198
setRetentionTime
cPeaksList, 374
cSequence, 396
setRetentionTimeUnit
cPeaksList, 374
setSelectedRegion
cImageWindowWidget, 301
setSequence
cDrawPeptideWidget, 268
cParametersWidget, 346
cSequence, 396
setStartIonType
cCandidate, 238
setStartModifID
cCandidate, 238
setSummary
cBrick, 198
setSummaryFormula
cCandidate, 238
cSequence, 397
setTag
cParametersWidget, 346
setTitle
cPeaksList, 374
setValidSequence
cTheoreticalSpectrum, 459
setValues
cPreferencesIonTypeDialog, 379
setWholeWord
cBricksDatabaseProxyModel, 209
cMainWindowProxyModel, 322
cModificationsProxyModel, 324
cSequenceDatabaseProxyModel, 402
cSummaryPeaksTableProxyModel, 433
showEvent
cCalibrationChartScene, 220
cChromatogramWindowWidget, 248
cPatternSimulatorSceneWidget, 350
cSpectrumSceneWidget, 424
size
cBricksDatabase, 207
cCandidateSet, 239
cDeNovoGraphNode, 265
cPeakListSeries, 361
cPeaksList, 374
cSequenceDatabase, 399
cTheoreticalSpectrumList, 462
sortAndFitSize
cTheoreticalSpectrumList, 462
sortByMass
cTheoreticalSpectrum, 459
sortbyMass
cPeaksList, 374
splitSite, 474
startGraphReader
cDeNovoGraph, 262
startsWithBracket
TRotationInfo, 475

store
cBrick, 198
cBricksDatabase, 207
cCandidate, 238
cDefineCalibrationWidget, 257
cFragmentIons, 277
cFragmentIonType, 279
cGraphWidget, 284
cIonType, 309
cParameters, 343
cPeak, 356
cPeakListSeries, 361
cPeaksList, 375
cPubChemSearchWidget, 390
cSequence, 397
cSequenceDatabase, 399
cSummaryFormula, 429
cTheoreticalSpectrum, 459
matchedSeries, 470
neutralLoss, 473
nodeEdge, 473
storeDoubleVector
utilities.h, 574
storeIntStringMap
utilities.h, 574
storeIntVector
utilities.h, 574
storeModificationsToPlainTextStream
cFragmentIons.h, 512
storeString
utilities.h, 575
storeStringIntMap
utilities.h, 575
storeStringIntMapVector
utilities.h, 575
storeStringVector
utilities.h, 575
storeToIBDStream
cPeaksList, 375
storeToPlainTextStream
cBricksDatabase, 207
cSequenceDatabase, 400
stripHTML
utilities.h, 576
stripIsomers
utilities.h, 576
stripIsomersFromStringVector
utilities.h, 576
summaryPeaksTableRowDoubleClicked
cSummaryPeaksTableWidget, 437

TRotationInfo, 474
endsWithBracket, 475
startsWithBracket, 475

uncharge
cFragmentIons.h, 513
updateCorrelation
cImageWindowWidget, 302
updateFilter
cImageWindowWidget, 302
updateFilterBySelectedRegion
cSummaryPeaksTableWidget, 437
updateHistologyPosition
cImageWindowWidget, 302
updatekNNList
cTheoreticalSpectrumList, 463
updateMicroscopyPosition
cImageWindowWidget, 303
updateMZInterval
cPatternSimulatorSceneWidget, 350
cSpectrumSceneWidget, 424
updateRawDataToPeakList
cImzML, 305
updateRetentionTimeInterval
cChromatogramWindowWidget, 249
updateRuler
cImageWindowWidget, 303
updateScanIDInterval
cChromatogramWindowWidget, 249
updateSummaryPeaksTableFilter
cImageWindow, 290
updateView
cGraphWidget, 284
updateXValueInterval
cCalibrationChartScene, 221
use64BitIntensityPrecision
cImzML, 305
use64BitMzPrecision
cImzML, 306
utilities.h
checkFile, 564
checkRegex, 564
compareDoubles, 565
compareMapsIntDouble, 565
compareStringVectors, 566
convertStringIntUnorderedMapToStringVector, 566
cropDecimalsByteArray, 566
cropPrecisionToSixDecimals, 566
cropPrecisionToSixDecimalsByteArray, 567
fact, 567
fixLabelCharge, 567
getPeptideTypeFromString, 568
getShortFileName, 568
getStringFromPeptideType, 568
isCompatibleVersion, 569
isWhiteSpace, 569
isWhiteSpaceExceptSpace, 569
loadDoubleVector, 570
loadIntStringMap, 570
loadIntVector, 570
loadString, 570
loadStringIntMap, 571
loadStringIntMapVector, 571
loadStringVector, 571
median, 571
parseBranch, 572

proxyModelCheckDouble, 572
proxyModelCheckInt, 573
proxyModelCheckString, 573
removeWhiteSpacesExceptSpaces, 574
storeDoubleVector, 574
storeIntStringMap, 574
storeIntVector, 574
storeString, 575
storeStringIntMap, 575
storeStringIntMapVector, 575
storeStringVector, 575
stripHTML, 576
stripIsomers, 576
stripIsomersFromStringVector, 576

wheelEvent
cCalibrationChartScene, 221
cChromatogramWindowWidget, 249
cImageWindowWidget, 303
cPatternSimulatorSceneWidget, 350
cSpectrumSceneWidget, 425
write
cImzML, 306

zoomToPeak
cSpectrumDetailWidget, 419