

# QuickMod Tutorial

---

*QuickMod* is a software tool to perform open modification searches (OMS) in spectral libraries containing annotated peptide fragmentation mass spectra. It was developed by the Proteome Informatics Group at the Swiss Institute of Bioinformatics. This tutorial describes the use of the principal java programs within the *QuickMod* toolbox, comprising *Liberator* – a tool to create annotated MSn spectral libraries from a list of peptide spectrum matches (PSM), *DeLiberator* - a tool to create decoy spectra, and *QuickMod*, which searches the spectrum libraries in an OMS mode. The *QuickMod* results can finally be visualized by *QMViz*. For further information we recommend reading the following publications:

Ahrné E, Nikitin F, Lisacek F, Müller M (2011) *QuickMod – a tool for open modification spectrum library searches*. *Journal Proteome Research*, 10, 2913-2921

Ahrné E, Ohta Y, Scherl A, Lisacek F, Müller M (2011), *An improved method for the construction of decoy peptide MS/MS spectra suitable for the accurate estimation of false discovery rates* *Proteomics*, 11, 4085-4095

Ahrné E, Müller M, Lisacek F (2010) *Unrestricted identification of modified proteins using MS/MS*, *Proteomics*, 10, 671-686

Ahrné E, Masselot M, Binz PA, Müller M, Lisacek F (2009) *A simple workflow to increase MS2 identification rate by subsequent spectral library search*, *Proteomics*, 9, 1731-1736

## Table of contents

1. Creating a spectral library with *Liberator*
2. Visualizing spectral libraries with *MS1MS2Viewer*
3. Adding decoy spectra to the spectrum library using *Deliberator*
4. Searching the spectrum library with *QuickMod*
5. Visualize the results with *QMViz*
6. Examples

## Liberator: Creating Spectral Libraries

Starting with a pepXML file, which contains the PSMs from an MSn sequence search, and the MSn spectral files (in format .dta, .mgf, .mzXML, or .mzML), we can build a library of the confidently annotated peptide fragmentation spectra using the *Liberator* tool. The library will be written in the SpectraST .sptxt file format. Typing `java -jar Liberator.jar -h` in a terminal gives us all the options:

```
usage: Liberator <PepXML> [-a] [-b <arg>] [-d <arg>] [-e <arg>] [-f <arg>]
      [--filter <arg>] [-h] [-i <arg>] [-l] [--maxCharge <arg>] [--maxMZ <arg>]
      [--minCharge <arg>] [--minMZ <arg>] [--msLevel <arg>] [-o <arg>] [-p
      <arg>] [-q] [-r <arg>] [--tol] [-v]
-a,--average                set the average mass mode for peptide mass
                             calculation
                             by default: monoisotopic.
-b,--bSize <arg>           batch size for MS files
                             by default: 10000.
-d,--dir <arg>             root directory for MS2 spectrum files
                             by default: .;.
-e,--ext <arg>            regular expression for MS2 spectrum file names
                             by default: mgf.
-f,--frags <arg>          define the fragmentation type(s):
                             a,b,c,x,y,z for backbone type fragments
                             i for immonium and p for precursor
                             by default: b,y.
      --filter <arg>       set the PSM filter (ex: (peptideprophet >=
                             0.9) & (hyperscore > 400.))
                             by default: peptideprophet >= 0.9
-g,--graph                export PSM graph file
-h,--help                 print this message.
-i,--setting-file <arg>  give a property file with all input settings.
-l,--loss                 enable neutral loss (-H2O, -NH3) fragmentation.

      --loss-advanced <arg> enable the user to define losses himself. Set
                             the amino-acid target and the kind of loss action
                             based on the grammar below:
                             // Examples of Rules
                             # simple:
                             C -> -16
                             # AA class target:
                             [KNQR] -> -23.5
                             # AA class with a modif range as target:
                             [KNQR]/40:80 -> del
                             # alternative editions:
                             [KNQR]/40:80 -> -23.5, del
                             # few rules:
                             [KNQR]/40:80 -> -23.5, del ; C -> -16
                             # sequential edit actions:
                             Y/79.9:80.1 -> del-18 ;
                             // Grammar Definition
                             rules:          rule (';' rule)*;
                             rule:          target '->' actions;
                             target:        aaclass ('/' range) ?;
                             actions:       action (',' action)*;
                             action:        (modaction)+ | remaction (modaction)*;
                             modaction:     '-'
                             INT | '-' FLOAT;
                             remaction:     'del';
                             aaclass:       AA | '[' AA (AA)* ']';
                             range:         number ':' number;
                             number:        INT | FLOAT;
                             // Lexers
```

```

INT:          '0'..'9'+ ;
FLOAT:        INT ('.' INT);
AA:           'A'..'Z' ;
--loss-phosphate enable phosphate loss on phosphorylated [YST]
                amino acid fragments.
                by default: false.
--loss-verbose  enable loss rule verbosity mode.
                by default: false.
--max-loss-sites <arg> limit the maximum of loss sites
                    by default: 2
--maxCharge <arg>    maximal precursor charge
                    by default: 10.
--maxMZ <arg>        maximal precursor mz value
                    by default: 5000.0.
--minCharge <arg>    minimal precursor charge
                    by default: 1.
--minMZ <arg>        minimal precursor mz value
                    by default: 100.0.
--msLevel <arg>     level of ms spectra
                    by default: 2.
-o,--output <arg>   set the output filename

-p,--precision <arg> define the number of fractional digits for
                    output
                    by default: 2.
-q,--quiet          quiet mode (verbose off)
                    by default: false.
-r,--priority-rule <arg> set the annot precedence rule for multi
                    annotated peak
                    by default: y>b.
--tol               define the tolerance for mz fragment peak
                    comparison
                    by default: 0.1.
-v,--version        print the version info.

```

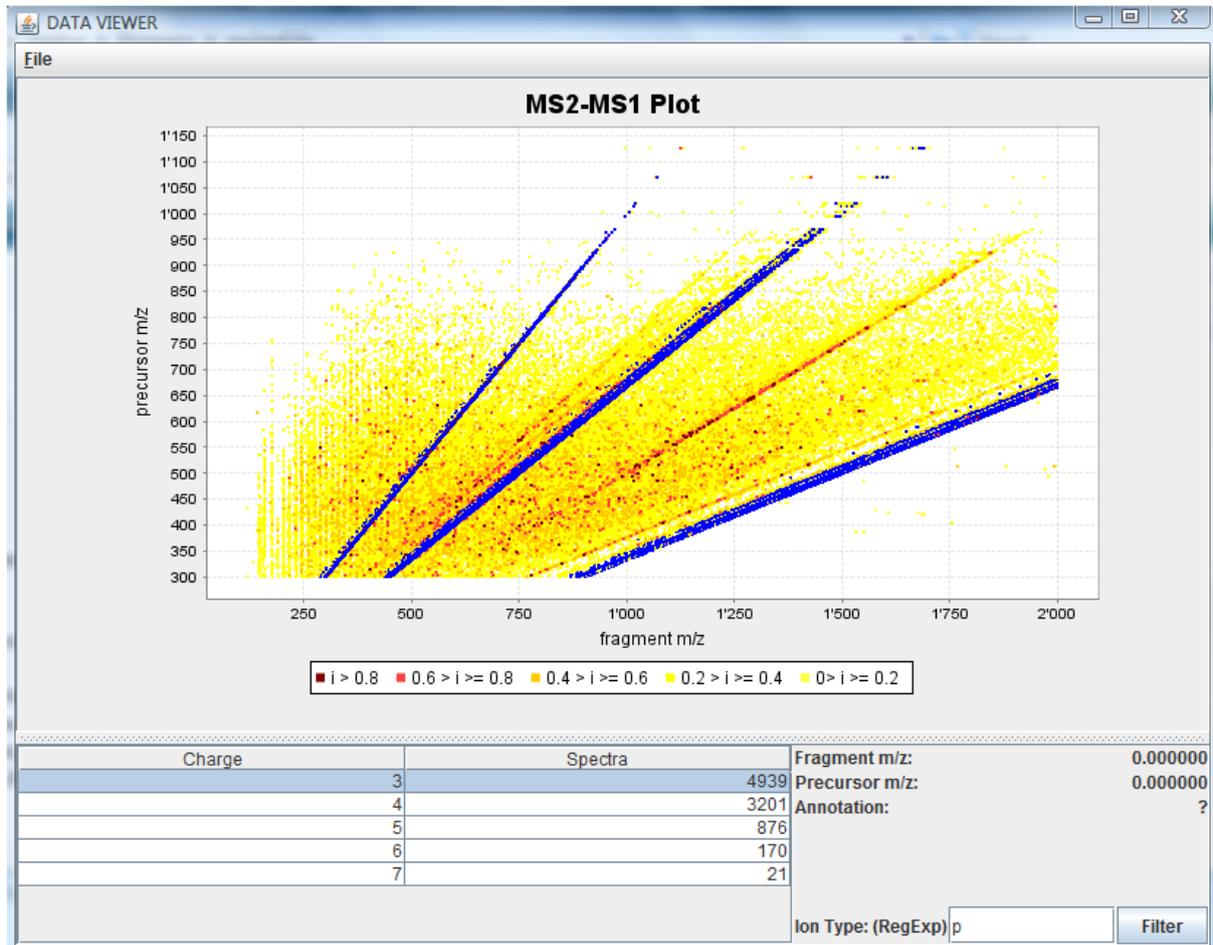
The first mandatory parameter is the path to the pepXML file. All PSMs with a Peptide Prophet probability higher than the one provided in the `--filter peptideprophet>0.95` option are written into the library (in future releases we will allow filters for all other scores in the peptide prophet file). Further PSMs can be selected according to their precursor m/z value and charge (`--minMZ`, `--maxMZ`, `--minCharge`, `--maxCharge`) and the MS level of the spectra (`--msLevel`). Spectra matching to the same peptide/charge pair are combined into a consensus spectrum. If the paths to the MS files provided in the pepXML file are not correct, you can specify the directory (`-d`, `--dir`) and the extension (`-e`, `--ext`) of the MS files.

The fragments considered for the peak annotation can be configured by means of the `-f`, `-frags` option as a comma separated list of fragment types (e.g a,b,y,i,p). Neutral losses can be configured. The (`-l`, `--loss`) option accounts for the H<sub>2</sub>O and NH<sub>3</sub> losses. The `--loss-phosphate` option will annotate all peaks that correspond to a loss of phosphate (79.966 Da) or phosphate and water (97.977). In order to reduce the complexity and the number of erroneous annotations, the number of combined losses per fragment is limited to `--max-loss-sites`. Fragment ion masses can be calculated as monoisotopic (default) or average (`-a`, `--average`). All annotations matching a peak within the m/z tolerance defined by the `-tol` option are considered. For the moment, we write only one annotation

per peak (most of the tools take only one annotation into account) and in case of several possible annotations their ranking becomes important. The ranking of the ion types can be calculated based on the specified priority rule (`-r, --priority-rule`), for example by `y>b>p` (future releases will support more complex ranking rules).

The resulting spectrum library is written in the `.sptxt` format to the directory specified by the (`-o, --output`) option. The precision of the fragment ion  $m/z$  values can be specified (`-p, --precision`).

## MS1MS2Viewer: Visualizing Spectral Libraries



The *MS1MS2Viewer* plots the fragment  $m/z$  values of all library spectrum peaks on the x-axis and their precursor  $m/z$  values on the y-axis. The intensity of the peaks is color coded. Different charge states are depicted in different views, which can be activated by selecting the appropriate entry in the Charge/Spectra table. The user can zoom into the views (dragging the mouse while keeping the left mouse button pressed). By moving the cursor on a peak he/she can display the annotation of this peak. The tool also allows highlighting specific peak annotations by defining a regular expression string that matches annotation string. Here we list some examples:

- p : will highlight all precursor ions
- c (byz) : will highlight all b(byz)-ions
- c2 : will highlight all c2-ions
- ? : will highlight all peak that have no annotation
- -18 : will highlight all -18 neutral losses

- `p[0-9]{1,2}\^2` : will highlight all doubly charged precursor ions
- `p.*-17` : will highlight all precursor ions with -17 neutral loss

## DeLiberator: Adding Decoy Spectra to Spectral Libraries

In order to calculate the false discovery rate in the list of matched query spectra we create decoy peptide fragmentation spectra using the *DeLiberator* tool. Each spectrum in a .sptxt file is parsed and a randomized version of the spectrum is calculated. The command `java -jar DeLiberator.jar -h` reveals all program options.

```
usage: Deliberator <mslib> [-a] [-c] [-d <arg>] [--decoy-tag] [-h] [-i
    <arg>] [--log <arg>] [-o <arg>] [-p <arg>] [-q] [-r <arg>] [-s <arg>] [-t
    <arg>] [-v] [-w <arg>]
-a,--average                set the average mass mode for
                             peptide mass calculation
                             by default: monoisotopic.
-c,--concat-libs            concatenate libs
                             by default: false.
-d,--dp-threshold <arg>   define the dot-product threshold
                             ([0-1[] for spectrum shuffling
                             by default: 0.5.
    --decoy-tag              set this decoy tag in 'Comments' of
                             decoy spectra
                             by default: No tag.
-h,--help                  print this message.
-i,--setting-file <arg>   give a property file with all input
                             settings.
    --log <arg>             define the log file.
-o,--output <arg>         set the output filename.
-p,--precision <arg>     define the number of fractional
                             digits for output
                             by default: 2.
-q,--quiet                 quiet mode (verbose off)
                             by default: false.
-r,--render-dir <arg>    render NA peak histograms
                             (render-dir/hist) and original + decoy
                             spectra (render-dir/scan).
warning:
-s,--sampling-prob <arg>  define the probability of sampling
                             non-annotated peaks ([0-1[]
                             by default: -1.0.
-t,--tol <arg>           define the tolerance for mz fragment
                             peak comparison
                             by default: 0.1.
-v,--version              print the version info.
-w,--sampling-interval-width <arg> define the bin width of
                             non-annotated (NA) peaks histograms for
                             sampling
                             by default: 100.
```

*Deliberator* either concatenates the decoy spectra to the original spectrum library (`-c, --concat-libs`) or writes them into a separate file. The fragment  $m/z$  values of the decoy spectra are calculated in monoisotopic or average mode (`-a, --average`) with a precision specified in the (`-p, --precision`) option. A decoy prefix (`--decoy-tag`) inserted at the beginning of the protein names in the comment line of the `.sptxt` files indicates the decoy character of these spectra. The path pointing to the output `.sptxt` file can be defined in `-o, --output` option.

Several technical parameters used in the decoy spectrum creation algorithm can be set. The most important parameter is the maximal allowed dot-product similarity between decoy and original spectrum (`-d, --dp-threshold`), where two peaks match if their  $m/z$  values are within a given  $m/z$  tolerance (`-t, --tol`). Other parameters deal with the sampling of not annotated peaks (`-s, --sampling-prob, -w, --sampling-interval-width`), which we will not discuss here.

## QuickMod: Searching Modified Peptides in a Spectrum Library

*QuickMod* matches a query peptide fragmentation spectrum against all library spectra that have a precursor mass within  $\pm\Delta$  of the precursor mass of the query spectrum. It aligns the query spectrum to all candidate library spectra, scores the alignment and positions the modification. It exports the spectrum matches, their scores and information about the modification positioning in a simple text file. Information about the statistical scores is also exported in a pdf file. The command `java -jar QM.jar -h` gives the program options:

```
usage: Please specify QuickMod options
-h          Print QuickMod help
-i <arg>    Absolute path to parameters file, (Required)
...
-v          verbose mode
```

*QuickMod* reads a parameter file (described below), which contains the search parameters and the paths to the spectrum library and query spectra files (mandatory `-i` option).

```
## Run Search from cmd line: java -Xmx 2048M -jar QuickMod.jar -i search.params
### QuickMod Search Parameters

# Search space
precursorMzTol=0.01
fragmentMzTol=0.4
modificationMassTol=100.0
targetedModificationMasses=-79.966:79.966

# Pre-processing
peaksPerWindow=2
```

```

windowSize=10
    ## Data filtering keep X peaks in window of Y m/z
    ## Default peaksPerWindow=1, windowSize=10
minPrecCharge=2
maxPrecCharge=3

#Scoring
oms=yes
    ## Enable Open Modification Scoring and Modification Site Assignment
minSpectraCorrelation=0.0
scoreWeights =1:0:0
    ## Weight assigned to each scoring feature
dotProduct:allIonCoverage:sequnceIonCoverage
    ## Sum must equal 1
    ## Default 1:0:0
intensityTransformation=3
    ## intensity transformation of spectral peaks for dotProduct score
calculation
    ## 0)NONE, 1)RANK, 2)LOG, 3)SQRT

# Results
candidatesPerSpectrum=1
    ## Number of library entry candidates listed per spectrum listed in results
file.
scoreThreshold=0
    ## Score cut-off on Spectrum-Spectrum Matches listed in results file.
    ## Default 0
fdrCutOff=0.05
    ## False Discovery Rate cut-off on Spectrum-Spectrum Matches listed in
results file.
    ## Default 0.05
resultsDirectory=C:\\QMTutorial\\results\\
    ## Absolute path to output directory of search results files
createFDRPlots=yes
    ## Create plots displaying the score distribution and FDR vs. #SSM per charge
state
    ## as well as a DeltaMass Histogram displaying the number of SSM per
modification mass.
pepXmlExport=yes
    ## export search results in PepXML format
resultsFileLabel=FOXO_OMS
    ## Specifiy output file label
    ## Default yourExpDataFileName_to_yourLibraryFileName

# Data partitioning
libraryBatchSize=1000
queryBatchSize=1000
    ## If experiencing memory problems limit the number of query and library
spectra read into memory.
    ## Default libraryBatchSize=30000, queryBatchSize=30000
    ## Also try increasing the upper memory limit of the Java virtual machine.
eg. java -Xmx 2048M -jar QuickMod.jar

# Display
enableProgressbar=yes

#Input data (.sptxt, .msp, .mgf, .mzxml, .mzml, .hms file formats are accepted)
    ## If you wish to search multiple files, specify name of the directory
containing all files to be searched.
    ## Specify absolute path.
expDataPath=C:\\QMTutorial\\mgf\\B06-1151_p.mgf
libraryPath=C:\\QMTutorial\\lib\\FOXO_pullndown_lib_loss_0.2.sptxt

```

The MS2 peptide spectrum library formats *QuickMod* can read are .sptxt and .msp. The library is defined in libraryPath and the query spectra in expDataPath. The user

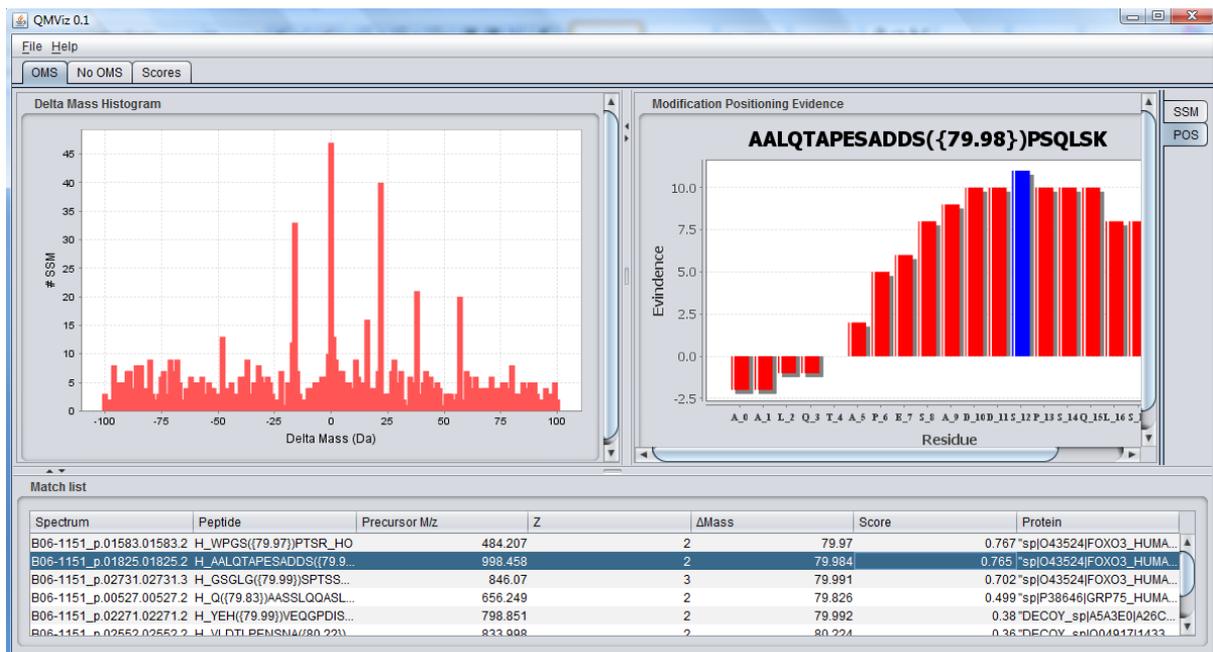
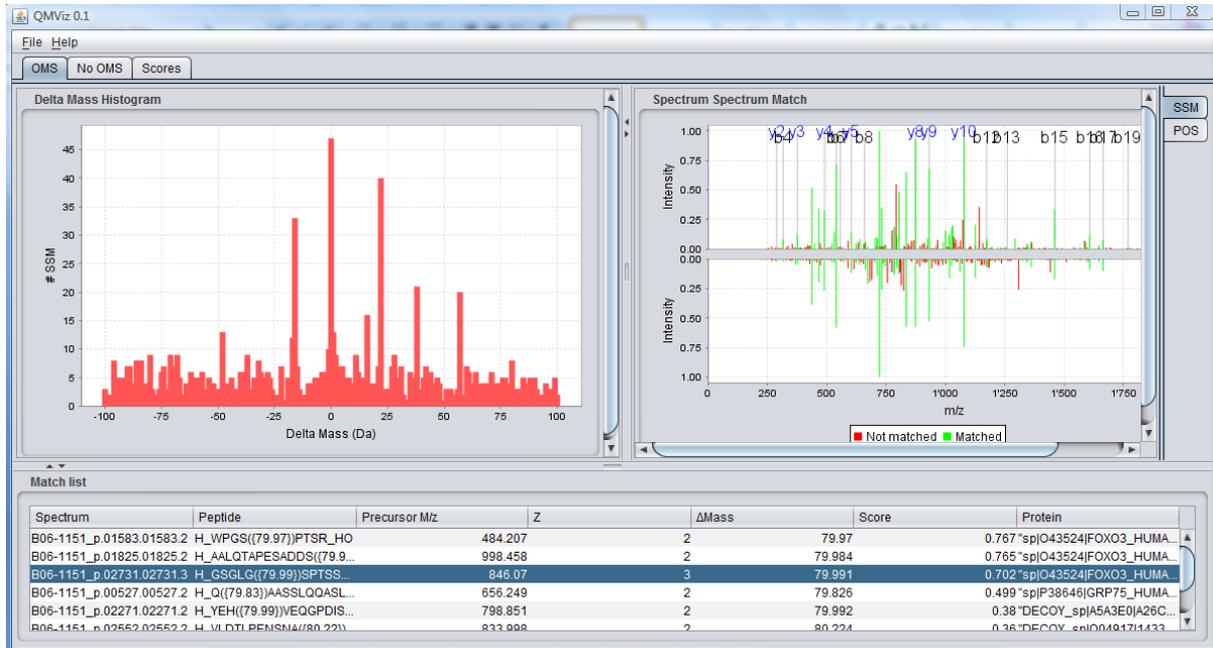
can choose the directory where all the output files are written (`resultsDirectory`). The library files as well as the MS files containing the query spectra are converted to binary files for performance reasons. By default, *QuickMod* takes the name of the original MS files and changes the extension to the extension of the binary file (.hms). The name of the result files is defined by the `resultsFileLabel`. The list of spectrum-spectrum matches in the result file can be filtered by both FDR (`fdrCutOff`) and score (`scoreThreshold`) thresholds. Only the highest ranking library spectrum matches to a query spectrum are written to the results file (`candidatesPerSpectrum`).

*QuickMod* can run in both OMS and standard mode (`oms=yes` and `oms=no`). The `precursorMzTol` is the absolute precursor ion m/z tolerance for the standard search and a maximum modification mass (`modificationMassTol`) is used for the OMS search. *QuickMod* is also able to search for a limited list of mass shifts (targeted modification search TMS), which is useful if the modifications of interest are known and speeds up the search considerably. The targeted modification masses (`targetedModificationMasses`) are separated by a colon (":"). If some masses are specified, then the search is run in TMS mode and the OMS mode is ignored. If no masses are specified and `oms=yes`, the search is executed in OMS mode.

Query spectra can be processed using a sliding windows approach, where only the `peaksPerWindow` highest peaks in a window of width `windowSize` are retained. A range of possible precursor charges can also be selected (`minPrecCharge`, `maxPrecCharge`). Peak intensities can be transformed as defined in `intensityTransformation`.

If the volume of all library spectra is too large to be loaded into the computer memory at once, it is possible to read the library and query spectra in batches where the batch size can be configured (`libraryBatchSize` and `queryBatchSize`).

## QMViz: Visualizing QuickMod results



QMViz reads a *QuickMod* configuration file and retrieves the links to the result and spectral files. OMS and standard search are performed independently in *QuickMod* and their results are shown in different tabs. The OMS tab shows the histogram of the mass differences between the neutral precursor masses of library and query spectrum. The user can zoom into the histogram by holding the left mouse button down and dragging the cursor to the right. Dragging the cursor to the left reverses the zoom and displays the original histogram. The mass differences are binned into 1 Da bins. Clicking onto one of these bins displays all the spectra matching with the selected mass shift in the lower panel. The matching

library/query spectra together with their annotations are depicted in the panel on the right hand side.

The No OMS tab displays a list of all spectra matching to library spectra without precursor mass shift. Selecting a spectrum entry in the table displays both library and query spectra in the panel on the right.

The Score tab displays the histograms of the score distribution for the different search modes and precursor charge states. Score histograms of decoy entries are given together with the calculated FDR.

**At the moment the viewer does not work if several query MS/MS file were used.**

## Examples

In this example section we describe a scenario where a sequence search was already run without modifications and the *QuickMod* tools are used to find modified variants of the peptides already identified in the sequence search. The data are taken from a protein interaction study (Rinner et al., 2007, Nat. Biotechnol., 25, 345-352), where human FOXO transcription factor and its interaction partners were isolated and analyzed with MS2 mass spectrometry. The identification was done with XTandem and the TPP.

1. Create a spectrum library based on the pepXML file and the mgf files. To avoid that warning messages are printed on the terminal during program execution, you can specify `2>error_log.txt` as a command line argument, which redirects the warning messages to the `error_log.txt` file.

```
java -Xmx1400m -Xms256m -jar Liberator.jar
  C:\QMTutorial\pepXML\FOXO_pulldown.pep.xml
  -d C:\QMTutorial\mgf
  --msLevel 2
  --tol 0.2
  -l
  --loss-phosphate
  -o C:\QMTutorial\lib\
  FOXO_pulldown_lib_loss_0.2.sptxt 2>error_log.txt
```

2. Visualize the spectral libraries : Double click on `ms2ms1Viz.jar`, which opens the `ms2ms1Viz` UI. First open the `ETD_library.sptxt` file in the `lib` folder. Highlight the parent masses and other masses of interest. Visualize the `IT` library to see the difference.
3. Add decoy spectra to the created spectral library:

```
java -Xmx1400m -Xms256m -jar Deliberator.jar
  C:\QMTutorial\results\FOXO_pulldown_lib_0_2.sptxt
  -c
  -decoy-tag DECOY_ 2>error_log.txt
```

4. Search the spectrum library with *QuickMod* in the OMS mode and the `FOXO_QM_search.params` parameter file in the `bin` folder:

```
java -Xmx1400m -Xms256m -jar QM-0.4.jar
  -i C:\QMTutorial\bin\FOXO_QM_search.params 2>error_log.txt
```

5. Visualize the results with the *QMViz* tool:

```
java -Xmx1400m -Xms256m -jar QMViz-0.1.jar 2>error_log.txt
```

See whether you can find any phosphorylations of the FOXO transcription factor. What are the most abundant modifications? Which other biological modifications are present?