glyXtoolMS Usermanual

Copyright Max-Planck institute of complex dynamical systems, Magdeburg,2018

This software is released under a three-clause BSD license:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
* Neither the name of any author or any participating institution may be used to endorse or promote products derived from this software without specific prior written permission.

For a full list of authors, refer to the file AUTHORS.

--------------------------------------------------------------------------

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANY OF THE AUTHORS OR THE CONTRIBUTINGINSTITUTIONS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

[1. Installation 4](#_Toc501624641)

[1.1. OpenMS 4](#_Toc501624642)

[1.2. Pyopenms 4](#_Toc501624643)

[1.3. glyXtoolMS 4](#_Toc501624644)

[2. Creating an OpenMS Workflow 4](#_Toc501624645)

[3. TOPPAS tools for glycopeptide analytics 5](#_Toc501624646)

[3.1. FeatureFinderMS 5](#_Toc501624647)

[3.2. FileBuilder 6](#_Toc501624648)

[3.3. glyXFilter 7](#_Toc501624649)

[3.4. Glycopeptide digest 9](#_Toc501624650)

[3.5. Glycan composition builder 10](#_Toc501624651)

[3.6. Glycopeptide Matcher 11](#_Toc501624652)

[3.7. Peptide Fragment Search 12](#_Toc501624653)

[3.8. glyxReporter 13](#_Toc501624654)

[4. glyXtool Evaluator 14](#_Toc501624655)

# Installation

# OpenMS

For installation of OpenMS visit <https://www.openms.de/> and follow the download/install or build instructions for your operating system.

After installation the following tools should be installed: TOPPAS and TOPPView.

# Python

To run glyXtoolMS, a python 2.7 installation is required, together with the package manager pip. The use of a virtual environment like virtualenvwrapper is recommended, to handle p

Install python 2.7 from <https://www.python.org/>. The package manager for python will then be installed, too. To check, open a console and type the command “pip”. If it has not been installed, follow the installation instructions on <https://pip.pypa.io/en/stable/installing/#do-i-need-to-install-pip>.

The use of a virtual environment is recommended, in case multiple python installations with different package setups are installed on the computer. For the installation of virtualenvwrapper, please refer to <https://virtualenvwrapper.readthedocs.io/en/latest/>

Virtualenvwrapper can be installed via:

pip install virtualenvwrapper

aftwerwards a fresh environment can be created using:

mkvirtualenv <envname>

switch into the environment using:

workon <envname>

# glyXtoolMS

glyXtoolMS can be installed using pip:

pip install glyXtoolMS

The dependencies canvasvg, configparser, lxml ,numpy,pyopenms, pyperclip, and xlwt should then be automatically downloaded and installed.

alternatively the .egg or .wheel can be downloaded from <https://test.pypi.org/project/glyxtoolms/>

or build manually from [www.github.com/mpioch/glyXtoolMS](http://www.github.com/mpioch/glyXtoolMS)

After the installation of glyXtoolMS, the glyXtoolMS Evaluator should acessable via the console command:

glyXtoolMS

# TOPPAS Script Setup using glyXtoolMS Evaluator

To get access to the glycopeptide scripts in TOPPAS, the scripts have to be copied into the OpenMS folder. Start the glyXtoolMS Evaluator, using the command glyXtoolMS in the command line. During startup, the path to the openMS installation folder will be requested. Afterwards the necessary files will be copied over from the glyXtoolMS python package into the OpenMS Scripts and Externals folder.

# Creating an OpenMS Workflow

An Example Data set can be download from [www.github.com/mpcio/glyxtoolms](http://www.github.com/mpcio/glyxtoolms), containing an IgG mass spectrometry file, the N-glycan database, the IgG Fasta file,

# TOPPAS tools for glycopeptide analytics

# FeatureFinderMS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pot. predecessor tools | → | FeatureFinderMS | → | pot. successor tools |
| MS centroid data in .mzML format  PeakPickerHighRes  PeakPickerWavelet  FileFilter | glyxFilter |

**Purpose**

Finds features around analytes containing at least one fragment spectrum.

**Parameters**

* inMZML: Input mass spectra as centroid data in \*.mzML file format
* outFeature: Feature output file
* tolerance: Mass tolerance in Dalton
* mswindow: maximum mass range of the precursor isotope pattern in dalton
* precursorshift: maximum deviation of the precursor mass from the (average) precursor mass reported within the mass file in dalton
* rtwindow: maximum elution range of the analyte peak in seconds

**Possible Input Nodes**

The tool uses centroided MS1 data. Possible input nodes are the file input node, the various OpenMS Peakpicker nodes or the FileFilter node if the data have to be cropped to a certain elution or mass range

**Possible Output Nodes**

* glyxFilter

**Similar tools**

* FeatureFinderCentroided
* FeatureFinderisotopeWavelet

# FileBuilder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pot. predecessor tools | → | FileBuilder | → | pot. successor tools |
| FileFilter  PeakPicker | glyxFilter |
| FileFilter  PeakPicker |

**Purpose**

Replaces the given MS Level spectra in an experiment. In the context of glycopeptide analysis it is used to replace continuous MS2 fragment spectra with their centroided counterpart after peakpicking, while retaining continuous data in the MS1 domain. This is needed as input for the ‘glyXtoolMS Evaluator’ to visualize continuous MS1 data for the precursors.

**Parameters**

* inOriginal: File input of mass spectrometry data in \*.mzML format; All MS level are transferred to the output file except the level provided by the option ‘MSLevel’
* inReplace: File input of mass spectrometry data in \*.mzML format; The spectra matching the given MSLevel option are transferred to the output file
* out: File output in \*.mzML format
* MSLevel: MS level which will be replaced with data from the replacement file

**Possible Input Nodes**

* FileFilter
* PeakPicker
* File input node

**Possible Output Nodes**

* glyXfiler
* File output node

# glyXFilter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pot. predecessor tools | → | glyxFilter | → | pot. successor tools |
| FeatureFinderCentroided  FeatureFinderMS\*  FeatureFinderIsotopeWavelt | glyxReporter  Glycopeptide Matcher\*  File Output node (.xml) |
| MS centroid data in .mzML format  PeakPickerHighRes  PeakPickerWavelet  FileFilter  FileBuilder\* |

**Purpose**

The tool searches for glycopeptide evidence in MS2 spectra, based on oxonium ions and neutral losses from the precursor. Reported is a spectrum score between 0.0 and 10.0 for each MS2 spectrum where the lower score signifies a higher glycopeptide probability. The identified glycopeptide fragment spectra are then used to identify glycopeptide features in the FeatureMap. For easier data access in later stages of the analysis pipeline the tool then generates consenus spectra for all identified glycopeptide features. All generated information is finally stored in a \*.xml file.

**Parameters**

* inMZML: Input mass spectra as centroid data in \*.mzML file format
* inFeature: Input feature file as\*.featureXML
* outGlyML: Output file in \*.xml format, containing all scored fragment spectra and all identified glycopeptide features
* createFeatures: (false/true); if true a dummy feature will be generated, if no feature could be found within the provided feature map for a given fragment spectra
* hasFucose: (false/true); if true use predefined oxonium ions that contain fucose
* hasNANA: (false/true); if true use predefined oxonium ions that contain N-acetylneuraminic acid
* hasNGNA: (false/true); if true use predefined oxonium ions that contain N-glycolylneuraminic acid
* oxoniumions: Add additional oxonium ions to the search.   
  Format has to be like: (NeuAc)1(H2O)-1(H+)1 with comma separated oxonium ions
* tolerance: Mass tolerance in Dalton for the oxonium ion search
* ionthreshold: Ignores peaks with lower intensity than the given threshold. Set to 0 to include all peaks.
* scorethreshold: Threshold used to identify a fragment spectrum as a glycopeptide. Lower scores signify a higher glycopeptide probability.

**Possible Input Nodes**

* inMZML: needed are centroided MS2 data with sorted peaks after mass. Suitable is the FileBuilder, to generate a suitable input file for the glyXtool Evaulator; PeakPicker or FileFilter if the MS2 data are already centroid data and need to be sorted
* inFeature: All possible FeatureFinder tools

**Possible Output Nodes**

* glyxReporter:
* glycoPeptideMatcher: for matching peptide and glycan composition to the precursor masses of identified glycopeptide features

# Glycopeptide digest

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pot. predecessor tools | → | Glycopeptide Digest | → | pot. successor tools |
| Protein sequences in \*.fasta format | Glycopeptide Matcher |

**Purpose**

Generating possible peptide sequences with glycosylation sites from protein sequences via theoretical digest.

**Parameters**

* inFasta: Input file in \*.fasta format containing either protein sequences or peptide squences
* out: \*.xml file containing the generated peptides with glycosylation sites, their possible modifications and the monoisotopic mass of each peptide
* enzymes: The enzyme(s) used for the digest. Currently supported are trypsin, AspN, Unspecific and NoDigest. The option ‘Unspecific’ cuts after each aminoacid and uses the Nr of missedCleavageSites as the maximum length of the reported peptides. With the option ‘NoDigest’ the provided sequences from the \*.fasta file are used without digest, allowing the user to specify peptides.
* cystTreatment: (None, Iodacetic acid, Iodoacetamide). Fixed modification on the cysteine residue
* modifications: (None, Oxidation(M), Acrylamide Adduct, Carbamylation (N-term)), Variable modifications. For each peptide all possible permutations are generated. If e.g. a peptide contains two methionines, three peptides are generated: (0 Oxidations, 1 Oxidation on either methionine and 2 oxidations on both residues)
* glycosylation: (N-glycosylation, O-glycosylation). Select which glycosylation site should be checked. Uses the motif N(S|T)(^P) as consensus sequence for N-glycosylation and (S|T) for O-glycosylation
* missedCleavageSites: maximum nr of missed cleavage sites. In case of unspecific digest determines the maximum length of the peptide.

**Possible Input Nodes**

* Input node with \*.fasta file

**Possible Output Nodes**

* Glycopeptide Matcher

# Glycan composition builder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pot. predecessor tools | → | Glycan Composition Builder | → | pot. successor tools |
| File with glycan compositions  Or no file node | Glycopeptide Matcher |

**Purpose**

Provides glycan compositions for the ‘Glycopeptide Matcher’ tool. A given list of glycan compositions can be filtered by the provided ranges if the ‘useAsFilter’ option is set to true, otherwise a list of glycan compositions is calculated in-silico with the given ranges.

**Parameters**

* in: File input
* out: Output file, containing the filtered glycan compostions in an \*.txt file
* useAsFilter: (false/true); If true, filters the glycan compositions from the input file according to the provided monomer ranges. If false disregards content of the input file and calculates all compositon permutations from the given monomer ranges.
* rangeHex: range of hexose within the glycan composition
* rangeHexNAc: range of N-acetylhexoseamine within the glycan composition
* rangeFuc: range of fucose within the glycan composition
* rangeNeuAc: range of N-acetylneuraminic acid within the glycan composition
* rangeNeuGc: range of N-glycolylneuraminic acid within the glycan composition

**Possible Input Nodes**

* File input node. In case the ‘useAsFilter’ is set to false, the content of the file input node is neglected, since the glycan compostion permutations are calculated based on the given ranges. However TOPPAS expects each tool to have an input node to run, thus some file has to be provided to the tool.

**Possible Output Nodes**

* Glycopeptide Matcher

# Glycopeptide Matcher

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pot. predecessor tools | → | Glycopeptide Matcher | → | pot. successor tools |
| glyxFilter\* | glyxReporter\*  File Output node (.xml) |
| Glycopeptide Digest\* |
| Glycan Composition Builder  File with glycan compositions |

**Purpose**

Matches a given list of peptides and glycan compositions to precursor masses of glycopeptide features.

**Parameters**

* out: Output file, Appends new collected information to the given inAnalysis file
* inAnalysis: Input file containing a glyML analysis file with scored glycopeptide features
* inGlycan: Input file containing a list of glycan compositions to match against
* inPeptide: Input file containing a list of peptide to match against
* accuracy: Mass tolerance in Dalton of the precursor matching. Used on the deconvoluted precursor mass

**Possible Input Nodes**

1. glyxFilter
2. Glycopeptide Digest
3. Glycan Composition Builder

File with glycan compositions

**Possible Output Nodes**

* glyxReporter
* Peptide Fragment Search

# Peptide Fragment Search

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pot. predecessor tools | → | Peptide Fragment Search | → | pot. successor tools |
| Glycopeptide Matcher | glyxReporter\*  File Output node (.xml) |

**Purpose**

Annotates the consensus spectra of glycopeptide features with peptide fragments based on the theoretical fragments of the peptide sequence suggested by the Glycopeptide Matcher tool.

**Parameters**

* inGlyML: Input analysis file in glyML format
* outGlyML: Output analysis file in glyML format
* tolerance: maximum mass deviation in Dalton for matching fragment spectra peaks to theoretical peptide fragment masses
* ionthreshold: Intensity threshold for annotating fragment spectra peaks. Set to Zero to ignore intensity.

**Possible Input Nodes**

* Glycopeptide Matcher

**Possible Output Nodes**

* glyxReporter

# glyxReporter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pot. predecessor tools | → | glyxReporter | → | pot. successor tools |
| glyxFilter  Glycopeptide Matcher  Fragment Search | File Output node (.xls) |

**Purpose**

Converts the collected information stored in the glyML file from the glycopeptide analysis tools into excel sheets.

**Parameters**

* inAnalysis: Input file in glyML format
* outReport: Output as \*.xls file

**Possible Input Nodes**

* glyxFilter
* Glycopeptide Matcher
* Fragment Search

**Possible Output Nodes**

* File output (\*.xls)

# glyXtool Evaluator

* Setup
* Changing Plot Parameter Setup
* Loading Data
* Feature Tab
* Identification Tab
* Annotations