GlycoBinder

version 1.0.0

Introduction

This document describes the use of the R script *GlycoBinder* and its key elements. *GlycoBinder* allows for streamlined data processing of multiplexed glycopeptide quantitative mass spectrometry data. It relies on usage of external tools (s. below) that are not distributed with the script and have to be requested and installed separately. *GlycoBinder* is a free software and distributed under GNU GPL v3.0 license (for details see the GNU General Public License [https://www.gnu.org/licenses/]). This license does not apply to external software *GlycoBinder* relies on. For this, different license terms may apply.

Programming Language

GlycoBinder is written using R programming language [https://www.r-project.org/], version 3.5.0. It also relies on freely available R-packages: *data.table* [https://github.com/Rdatatable/data.table], *dplyr* [https://github.com/tidyverse/dplyr], *future.apply* [https://github.com/HenrikBengtsson/future.apply], and *stringr* [https://github.com/tidyverse/stringr].

External tools

GlycoBinder combines the following external tools for processing mass spectrometry data:

1. RawTools, version 2.0.2 [https://github.com/kevinkovalchik/RawTools]

Kovalchik, K.A., Colborne, S., Spencer, S.E., Sorensen, P.H., Chen, D.D., Morin, G.B. and Hughes, C.S., 2018. RawTools: Rapid and Dynamic Interrogation of Orbitrap Data Files for Mass Spectrometer System Management. Journal of proteome research, 18(2), pp.700-708.

2. msconvert (ProteoWizard), version 3.0.19262 (0a01c36ac) [https://github.com/ProteoWizard/pwiz]

Chambers, M.C., Maclean, B., Burke, R., Amodei, D., Ruderman, D.L., Neumann, S., Gatto, L., Fischer, B., Pratt, B., Egertson, J. and Hoff, K., 2012. A cross-platform toolkit for mass spectrometry and proteomics. Nature biotechnology, 30(10), p.918.

3. pParse, version 2.0.8 [http://pfind.net/software/pParse/index.html]

Yuan, Z.F.E., Liu, C., Wang, H.P., Sun, R.X., Fu, Y., Zhang, J.F., Wang, L.H., Chi, H., Li, Y., Xiu, L.Y. and Wang, W.P., 2012. pParse: A method for accurate determination of monoisotopic peaks in high-resolution mass spectra. Proteomics, 12(2), pp.226-235.

4. pGlyco, version 2.2.0 [http://pfind.net/software/pGlyco/index.html]

Liu, M.Q., Zeng, W.F., Fang, P., Cao, W.Q., Liu, C., Yan, G.Q., Zhang, Y., Peng, C., Wu, J.Q., Zhang, X.J. and Tu, H.J., 2017. *pGlyco 2.0* enables precision N-glycoproteomics with comprehensive quality control and one-step mass spectrometry for intact glycopeptide identification. Nature communications, 8(1), p.438.

GlycoBinder does not provide those tools and a user needs to request and install the tools by himself prior to working with *GlycoBinder*. To our knowledge, the tools are freely available upon request.

Requirements for the processing environment

GlycoBinder was developed and tested on machines running on 64-bit platforms under Windows 10 and R programming language versions 3.5.0 or higher. Respectively, it requires an R programming language (versions 3.5.0 or above) to be installed on your machine including *data.table*, *dplyr*, *future.apply*, and *stringr* packages. In case those packages are not installed, *GlycoBinder* will make an attempt to install them.

Since *GlycoBinder* relies on external tools, all of them should be installed and configured prior to using the script. All external tools have to be added to the system path of the machine *GlycoBnder* is working on. Later allows for envoking the tool without specifying an exact path to it that might differ from one computer to another.To do so, search for `Edit environment variables for your account` and then click on it. Under "User Variables" select "PATH" and click the `Edit` button (make sure you are changing the "PATH" variable for a user account you will be later working). Select `New` and then `Browse`. Navigate to the directory where the executable of the tool is located. Repeat the same procedure for all the tools. We also suggest to add the file path to the folder containing "Rscript.exe" file, which is needed to run *GlycoBinder* using the command line. "Rscript.exe" is typically located within the folder containing files belonging to R, e.g. *C:/Program Files/R/R-3.5.0/bin/x64/*. After the enviromental variables are configured, please check if the tools can be accessed from the command line directly. For this, open the command line and type one by one: `RScript`, `rawtools`, `msconvert`, `pparse`, `pglyco`. Hit `Enter` after each command. Make sure that system can find each tool and returns help information to the console. A tutorial how to configure environmental variables can be found here: https://github.com/kevinkovalchik/RawTools/wiki/Download-and-prepare-RawTools-for-Windows

Depending on the number of raw files and their size, *GlycoBinder* might require a large amount of RAM to process the data. Per default, it will use number_of_awailable_processors - 2 threads on your machine for processing the data (this number might be different for external tools). We recommend to reserve at least 1GB of free RAM per running process (e.g. for a machine with 8 cores, one should aim for at least 6 GB of free RAM space). If you would like to restrict the number of processors used by *GlycoBinder*, please, consult the following section regarding additional parameters to the script.

Processing steps in brief

GlycoBinder is designed for processing *.raw* files acquired on Thermo Fisher Orbitrap instruments. It allows for combination of MS spectra resulting from MS2 and SPS-MS3 scans and use of isobaric peptide labeling reagents, e.g. TMT, for quantification.

In brief, *GlycoBinder* makes following steps in the data processing:

1. *RawTools* is used for extracting quantitative information (reporter ion intensities) from Thermo *.raw* files and assigning MS3 scan to corresponding MS2 scans.

2. *msconvert* transforms *.raw* files into *.mgf* file format and centroids data by applying vendor peak picking algorithm. MS2 and MS3 scans are preserved in the *.mgf* file.

3. *pParse* recalibrates the monoisotopic peaks of precursors and outputs an *.mgf* file containing MS2 scans.

4. *GlycoBinder* combines ion intensities of matching MS2 and MS3 spectra as reported by *RawTools*. MS2 and MS3 spectra are extracted from *msconvert*-produced *.mgf* file and merged based on the specified ion tolerance window. *GlycoBinder* replaces MS2 spectra in the *pParse* output by combined MS2/MS3 spectra. Modified *pParse* output file is used as an input for *pGlyco 2.0*.

5. *pGlyco 2.0* uses the combined spectra to search for peptides and associated glycans. After the first *pGlyco 2.0*-search is finished, results are filtered based on a specified FDR cutoff.

6. Optionally, a second *pGlyco 2.0*-search is performed on a smaller protein data base. For this, only proteins containing modified peptides identified during the first *pGlyco 2.0*-search and passing the total FDR threshold are retained in the protein sequence database used for the second peptide search. Please, consult the section about additional parameters to *GlycoBinder* in order to disable the second *pGlyco 2.0* search.

7. *GlycoBinder* combines *pGlyco 2.0* search results and reporter ion intensities extracted by *RawTools*. Resulting table is used to prepare quantitative data at different levels: at the levels of glycosylated peptides, glycoforms, glycosites, and glycans.

Using GlycoBinder

To execute *GlycoBinder*, follow the steps:

1. Prepare a working directory containing *.raw* files to be processed and *.fasta* file containing protein sequences.
2. Open the command line
3. Specify the path to the *Rscript.exe* (or just "Rscript.exe" if the file path is set in environmental variables)
4. Specify the path to the *GlycoBinder.R*
5. Specify the path to the working directory using `--wd` flag
6. Specify peptide labeling reagent after `--reporter_ion` flag (values supported by *RawTools* are allowed: "TMT0", `TMT2`, `TMT6`, `TMT10`, `TMT11`, `iTRAQ4`, `iTRAQ8`), e.g. `--reporter_ion TMT6`
7. Specify additional arguments (s. below)

Suppose, *.raw* files, the *.fasta* file, and *GlycoBinder.R* script are located in *C:/data* folder, and peptides were labeled using TMT6plex reagents, the minimum required input would look like:

```
C:/data>Rscript.exe "GlycoBinder.R" --wd "C:/data" --reporter_ion TMT6
```

Additional parameters

Following parameters modify default *GlycoBinder* behavior if added as command line arguments:

1. `--verbose`
   Forces *GlycoBinder* to be more chatty.

2. `--tol_unit`
   Specify tolerance unit used for matching ions from corresponding MS2 and MS3 spectra. Supported values are `ppm` and `Th`, e.g. `--tol_unit ppm` (default).

3. `--match_tol`
   Specify tolerance for matching ions from corresponding MS2 and MS3 spectra. Integer numbers are supported, e.g. `--match_tol 1` (default). Default tolerance widow for ion matching is 1 ppm. It means, if two ions in the matching spectra have an absolute mass difference smaller than 1 ppm, those peptides will be considered the same and their intensities will be summed.

4. `--pglyco_fdr_threshold`
   Specify total FDR cutoff for *pGlyco 2.0* search results, e.g. `--pglyco_fdr_threshold 0.02` (default) sets maximum total FDR to 2%.

5. `--no_second_search`
   Prevent *GlycoBinder* from running second *pGlyco 2.0* search on reduced data base.

6. `--report_intermediate_results`
   Forces *GlycoBinder* to keep intermediate files (after *pGlyco 2.0* search).

7. `--nr_threads`
   Specify number of available processors for *GlycoBinder* processing. It can take values between 1 and the number of available processors - 2 (default).

8. `--seq_wind_size`
   The parameter specifies the number of amino acids around the modification site. It is applied to extract sequence window around modification site from protein sequences. Sequence windows are needed to combine quantitative information on glycoform level. Default paramter is 7, e.g. `--seq_wind_size 7`. Seven amino acid before the modified site and seven amino acids after the modified site will be extracted, resulting in the 15 amino acids long sequence window.

Default parameters for external tools

Per default, external tools are started using parameters listed below. The majority of these parameters are fixed. However, one can execute those tools outside of *GlycoBinder* using a different parameter set and then supply the output files into the respective folder within the *GlycoBinder* working directory (specified after `--wd` flag while running the script). In this case, *GlycoBinder* skips execution of a respective tool.

1. *RawTools*
   `rawtools -parse -d [input directory] -out [output directory] -q -r [reporter ions type] -R -u`
   RawTools output one *_Matrix.txt* file per *.raw* file. Output file names are created by appending *_Matrix.txt* to the *.raw* file name including extension (example: "raw_file.raw" becomes "raw_file.raw_Matrix.txt"). *RawTools* output files are located in *./rawtools_output* folder within the *GlycoBinder* working directory (location of the *.raw* files). One can process *raw* files externally and then copy the resulting *_Matrix.txt* files into the *./rawtools_output* folder. If every *.raw* file has a corresponding *_Matrix.txt* file, *GlycoBinder* will skip *RawTools* processing.

2. *msconvert*
   `msconvert [file] --outdir [output directory] --mgf --ignoreUnknownInstrumentError --filter "peakPicking vendor" --filter "defaultArrayLength 1-" --filter "titleMaker <RunId>.<ScanNumber>.<ScanNumber>.<ChargeState>"`
   Similar to *RawTools*, *msconvert* outputs one *.mgf* file per *.raw* file in the *GlycoBinder* working directory. Output file names consist of the original file name without *.raw* extension substituted by *.mgf*. *msconvert* output files are located in *./msconvert_ouput* folder within *GlycoBinder* working directory. If all *.mgf* files are present in there, *GlycoBinder* skips *msconvert* processing step. For correct processing of *.mgf* files generated by *msconvert*, each scan within an *.mgf* file should contain a line starting with "TITLE=" and containing a scan number flanked by dots, e.g. ".355.".

3. *pParse* `pParse.exe -D [file] -O [output directory] -p, 0`
   pParse output files are located in *./pparse_output* folder and named as original *.raw* files with *.raw* file extension substituted by *_[Type of Detector, e.g. CDFT of ITFT].mgf*. Similarly, *GlycoBinder* processing is skipped if all output files are found within the *./pparse_output* folder. After merging of MS2 and MS3 spectra, MS2 spectra within *pParse* output files are substituted by the combined MS2/MS3 spectra. The modified *pParse* output files are renamed to *base_raw_file_name_pParse_mod.mgf* files and saved in the same *./pparse_output* folder. If all *_pParse_mod.mgf* are found in the *./pparse_output* folder, *pParse* processing and merging of the MS2 and MS3 spectra are skipped.

4. *pGlyco 2.0*
   `pGlycodb.exe [pglyco configuration file] && pGlycoFDR.exe -p [pglyco configuration file] -r [output file name] && pGlycoProInfer.exe`

pGlyco 2.0 workflow consist of three programs, *pGlycodb.exe*, *pGlycoFDR.exe*, and *pGlycoProInfer.exe* that are executed one after another and rely upon configuration file that should be created before run. If *GlycoBinder* does not find any file with a name *pGlyco_task.pglyco* in the working directory, it will create a configuration file with default parameters. One can create its own configuration file, e.g. using GUI of *pGlyco 2.0*, name it as *pGlyco_task.pglyco* and then copy it to the working directory of *pGlyco 2.0*. In this case, *pGlyco 2.0* will utilize the existing parameter file for glycopeptide search. Following parameters are used per default and can be changed when supplying a GUI-created *pGlyco_task.pglyco* file to the *GlycoBinder* working directory:

- `enzyme=Trypsin_KR-C`
- `max_miss_cleave=2`
- `max_peptide_len=40`
- `min_peptide_len=6`
- `max_peptide_weight=4000`
- `min_peptide_weight=600`
- ```
  [modification]
  fix_total=3
  fix1=Carbamidomethyl[C]
  fix2=TMT6plex[K]
  fix3=TMT6plex[AnyN-term]
  max_var_modify_num=3
  var_total=1
  var1=Oxidation[M]
  ```
- ```
  [search]
  search_precursor_tolerance=10
  search_precursor_tolerance_type=ppm
  search_fragment_tolerance=20
  search_fragment_tolerance_type=ppm
  ```

Other parameters are fixed or will be overwritten irrespectively of the origin of the configuration file. Furthermore, same parameter file will be applied in the second *pGlyco 2.0* search, with exception that the protein database file will be changed to the reduced version of the *.fasta* file. The output file is *pGlycoDB-GP-FDR-Pro.txt* for the first *pGlyco 2.0* search and *pGlycoDB-GP-FDR-Pro2.txt* for the second search, respectively. Both files are located in the *./pglyco_output* folder. If the file *pGlycoDB-GP-FDR-Pro.txt* exists (or *pGlycoDB-GP-FDR-Pro2.txt* exists and `--no_second_search` flag was not used), *GlycoBinder* will skip the first (or first and second) *pGlyco 2.0* search, respectively.

Special case: MS2 data

After processing with *RawTools*, files that were identified as not containing MS3 scans will not be subjected to *msconvert* processing. The MS2/MS3 spectra merging step is skipped. After *pParse* processing, original *pParse* output files are renamed to *_pParse_mod.mgf* files for consistency and used as input for *pGlyco* directly.

Merging of MS2/MS3 spectra

GlycoBinder combines MS2 and MS3 spectra based on MS2 and MS3 spectra scan number pairs in the *RawTools* output files (*MS2ScanNumber* and *MS3ScanNumber* columns within *_Matrix.txt* file). First, ions from MS2/MS3 scan pairs are roughly matched using 1 Th tolerance window. Initially matching ions are then tested to satisfy the specified tolerance window (1 ppm per default, it can be changed by specifying `--tol_unit` and `--match_tol` arguments). If several ions matches the same ion, the ions with the minimal absolute mass diference are considered as a matching ion pair. Intensities of matched ions are summed.

Remaining MS3 ions that do not have matching MS2 ions are simply added to the MS2 spectra. *pParse .mgf* file then will output merged MS2/MS3 spectra. *GlycoBinder* matches spectra in the *pParse* output file to the merged MS2/MS3 spectra based on the scan number. While scan number is unique for merged MS2/MS3 spectra, several spectra in the *pParse* output can refer to the same scan number. For all of them, the spectrum will be substituted by the respective merged MS2/MS3 spectrum. Spectra that do no share scan number with merged MS2/MS3 spectra will be kept unchanged.

GlycoBinder output

As a last step in the data processing, *GlycoBinder* combines peptide information identified by *pGlyco 2.0* and reporter ion quantities extracted by *RawTools*. It combines intensity information at different levels by summing the respective ion intensities. All *GlycoBinder* output files are located in the *./pglyco_output* folder within the *GlycoBinder* working directory.

1. `pglyco_quant_results.txt`
   Combination of *pGlyco 2.0* output (*pGlycoDB-GP-FDR-Pro.txt* or *pGlycoDB-GP-FDR-Pro2.txt*) and all *RawTools* output files (*_Matrix.txt* files). Quantitative information from *RawTools* output is merged with *pGlyco 2.0* output file based on the *.raw* file name and MS2 scan number. Column descriptions can be found in the documentation for *pGlyco 2.0* and *RawTools*.

2. `pGlyco_Scans.txt`
   Same `pglyco_quant_results.txt` file filtered based on the total FDR cutoff (lesser than 2% FDR per default, can be changed when specifying `--pglyco_fdr_threshold` parameter).

3. `pGlyco_modified_peptides.txt`
   Based on `pGlyco_Scans.txt`, respective reporter ion intensities are combined (summed) for each peptide carrying specific glycan structure, e.g. reporter ion intensities are combined for the same glycopeptide identified in different *.raw* files or carrying additional variable modifications (apart from glycosylation). Corresponding precursor information is concatenated using default *pGlyco 2.0* separator ("/") and is preserved within the columns `pGlyco_ids`, `RawName`, `Scan`, `PrecursorMZ`, `Charge`, `Mod`, `ParentPeakArea`. `pGlyco_ids` column refers to `id` column in the `pGlyco_Scans.txt` table. Columns `Peptide`, `GlySite`, `Glycan(H,N,A,G,F)`, `GlyID`, `PlausibleStruct`, `GlyFrag`, `GlyMass`, `Proteins`, `ProSite`, keep the information about the glycan structure and possible protein assignment. `Leading_Protein` and `Leading_ProSite` reports the selected protein and corresponding site based on criteria discussed below.

4. `pGlyco_glycoforms.txt`
   The table is based on `pGlyco_modified_peptides.txt` table. It combines quantitative information based on sequence window and a particular glycan structure. Sequence windows are first extracted from the amino acid sequences of corresponding proteins. Per default, +/-7 amino acids are extracted around the modification site (can be changed if specifying `--seq_wind_size` parameter). Peptides are grouped based on modification site they share. Sequence windows extracted from proteins that could potentially contribute to those peptides are ranked based on the number of peptides in the group each sequence window can explain. Ties are broken by using protein ranking (s. description below). Peptides shared among several sequence windows are assigned to the sequence window that encompasses the majority of the peptides within the peptide group. If there are peptides that cannot be explained by the leading sequence window, those peptides are distributed between other sequence windows accordingly. Intensity information is then combined based on sequence window and glycan structure (reported in `seq_win` and `Glycan(H,N,A,G,F)` columns, respectively). Columns `modpept_ids`, `Scan`, `pGlyco_ids`, `Peptide`, `GlySite`, `GlyID` represent a concatenation of entries in respective columns in the `pGlyco_modified_peptides.txt` table. ";" is used as a separator by concatenation. `modpept_ids` refers to the `id` column in the `pGlyco_modified_peptides.txt` table. It contains the respective ids of the peptides that were combined by particular sequence window and glycan structure.

5. `pGlyco_glycosites.txt`
   The table is based on `pGlyco_modified_peptides.txt` table. The combination is based on sequence window information only, irrespective of the glycan structure. Accordingly, it contains `seq_win` column with sequence window information, `modpept_id` column that refers to `id` column in the `pGlyco_modified_peptides.txt`. Columns `Scan`, `pGlyco_ids`, `Peptide`, `GlySite`, `GlyID`, `Glycan(H,N,A,G,F)`, `GlyMass` are concatenaions of respective columns in `pGlyco_modified_peptides.txt` using ";" as a separator. `Leading_Protein` and `Leading_ProSite` are selected according to protein rank. Proteins are ranked based on the number of unique peptides (highest priority), number of all peptides, number of glycoforms, whether it is a swiss prot entry, and whether it is an isoform (lowest priority). Proteins that have greater number of unique peptides/total peptides/glycoforms, annotated in SwissProt data base and are not isoforms, receive a higher rank. The highest rank is 1. The rank is unique and ties, if occur, are broken by alphabetic order.

6. `pGlyco_glycans.txt`
   The table is based on `pGlyco_modified_peptides.txt` table. The combination is based on glycan structure only, irrespective of the peptide sequence (using `Glycan(H,N,A,G,F)` column). As before, `modpept_id` column refers to `id` column in the `pGlyco_modified_peptides.txt`. Columns `pGlyco_ids`, `Scan`, `Leading_Protein`, `Leading_ProSite` are concatenations of respective columns in `pGlyco_modified_peptides.txt` using ";" as a separator.

Potential Problems / Special use cases

1. *GlycoBinder* does not find one of the external tools or cannot execute it
   Check that the tool is accessible through the command line. If not, check if the path to the tool is saved in the system paths (s. `Requirements for the processing environment` section). Try to process the files using command line and same arguments as described for that tool. If correct output is created, copy it to a respective output folder within the *GlycoBinder* working directory. Re-run *GlycoBinder*. It should skip the problematic step and continue with the next one. If the tool does not return a correct output, consult the help page of the tool.

2. *GlycoBinder* is suspended in one of the steps and does not continue with other steps
   First, make sure that you gave *GlycoBinder* enough time to finish the task. Check, if external tools have created a proper output. If the output is created, stop *GlycoBinder* by closing the command line window and try to restart it. If the output is not complete, try to use the the tool outside of *GlycoBinder*, as described in the point 1.

3. *GlycoBinder* cannot find working directory
   Check that the file path specified after `--wd` argument does not contain white spaces or properly enquoted. Same applies when specifying the location of the script itself.

4. *GlycoBinder* cannot find *.raw* files
   Make sure that *.raw* files are located in the specified working directory and have *.raw* extension.

5. Minimal requirements to run *GlycoBinder* are:

- Installed and properly configured environment (R and respective packages, external tools, configured file paths)
- Directory containing *.raw* files and a *.fasta* file. The path to the directory is specified after `--wd` flag in the command line
- Specifying which labeling reagent was used for quantification by using `--reporter_ion` flag.

6. Use of external tools with different parameters
   Output of all external tools can be created outside of *GlycoBinder* workflow and then copied into respective output folder within the *GlycoBinder* working directory. In this case, *GlycoBinder* will skip the respective processing step if it can find respective files.

For *pGlyco 2.0* there is an option to pre-configure a parameter file by using *pGlyco 2.0* GUI and save the file in the *GlycoBinder* working directory under the name "pGlyco_task.pglyco". *GlycoBinder* will utilize it instead of default settings.