

# Glycompare

# GlyCompareCT GlyCompare command-line tool. GlyCompare is a novel method wherein glycans from glycomic data are decomposed to a minimal set of intermediate substructures, thus incorporating shared intermediate glycan substructures into all comparisons of glycans.

## Citation

Bao, Bokan, Benjamin P. Kellman, Austin WT Chiang, Yujie Zhang, James T. Sorrentino, Austin K. York, Mahmoud A. Mohammad, Morey W. Haymond, Lars Bode, and Nathan E. Lewis. **“Correcting for sparsity and interdependence in glycomics by accounting for glycan biosynthesis.”** Nature communications 12, no. 1 (2021): 1-14. <https://doi.org/10.1038/s41467-021-25183-5>

## Installation

First, please make sure you have `conda` installed. Version recommendation: `conda` 4.9.2 and later versions. - Install `conda` on Windows: <https://docs.conda.io/projects/conda/en/latest/user-guide/install/windows.html> - Install `conda` on Mac OS: <https://docs.conda.io/projects/conda/en/latest/user-guide/install/macos.html>

Please `git clone` the main branch to your target local directory.

```
# get the repo
git clone https://github.com/yuz682/GlyCompareCT.git
# enter the repo
cd GlyCompareCT
```

All dependencies required to run GlyCompareCT can be installed using `environment.yml`. A new `conda` environment is created with all dependencies installed. This step will take a while (10 - 15 minutes).

```
# Create the environment with all required dependencies installed.
conda env create -f environment.yml
```

Activate the new environment `glycompareCT`. Then the preprocessing is all done.

```
# Activate conda environment
conda activate glycompareCT
```

## Executables

Executables for Window, MacIntel, and Linux can be downloaded from the release or zenodo. The binary file is glyCompareCT (or glyCompareCT.exe). To use more conveniently, you can export the path to PATH variable by

```
export PATH="<path>/<to>/glyCompareCT/<directory>":$PATH
```

then

```
source ~/.bashrc
```

## User manual

Please refer to the GlyCompare wiki regarding input file format and more details about input parameters. Please ignore some inconsistent wording as the wiki was written for a web app.

## Quick start

Retreive example data

```
git clone https://github.com/LewisLabUCSD/GlyCompare.git
```

Glycopare decomposition of structural, linkage-specific HMO data with no normalization, 2 cores, integer substructure counting, epitope-based motif extraction

```
python glyCompare.py structure \
  -a GlyCompare/example_data/paper_hmo/source_data/abundance_table.csv \
  -v GlyCompare/example_data/paper_hmo/source_data/annotation.csv \
  -o output_hmo/ -p glycoCT -c 2 \
```

Glycopare decomposition of structural, linkage-specific HMO data with Probabilistic Quotient normalization, 2 cores, binary substructure counting, lactose-based motif extraction

```
python glyCompare.py structure \
  -a GlyCompare/example_data/paper_hmo/source_data/abundance_table.csv \
  -v GlyCompare/example_data/paper_hmo/source_data/annotation.csv \
  -o output_hmo/ -p glycoCT -n prob_quot \
  -m binary -c 2 -r lactose
```

## Structure data

```
python glyCompare.py structure -a <ABUNDANCE TABLE> -v <VARIABLE ANNOTATION>
-o <OUTPUT_DIRECTORY> -p <GLYCAN_DATA_TYPE> [-n <NORMALIZATION_MODE>,
```

`-m <SUBSTRUCTURE_ABUNDANCE_MULTIPLIER>`, `-c <NUMBER_OF_CORES>`, `-r <ROOT>`,  
`-u <CUSTOM_ROOT>`, `-d`, `-s`, `-b`, `-i`]

Required arguments:

Parameter	Description
<code>-a</code>	The file directory to the abundance table, in csv format
<code>-v</code>	The file directory to the variable annotation table, in csv format
<code>-o</code>	The directory to save the outputs, folder
<code>-p</code>	Glycan data type, choose from <code>&lt;'glycoCT', 'iupac_extended', 'linear_code', 'wurcs', 'glytoucan_id'&gt;</code>

Optional arguments:

Parameter	Default	Description
<code>-s</code>	None	Add this parameter if the input glycans don't have linkage information. The default assumes linkage information inclusion.
<code>-c</code>	1	The number of cores to use

Parameter	Default	Description
-n	'none'	Input glycans normalization within each glycoprofile, choose from <'none', 'min-max', 'prob-quot'>.'none': no normalization;'min-max': each element x is set to $(x - \min) / (\max - \min)$ ;'prob-quot': A commonly seen normalization method in biological data described in <i>Dieterle et al. 2006</i>
-b	None	Add this parameter to keep the absolute value of the substructure abundance. If not set, the substructure will be normalized by sum.
-m	'integer'	Substructure abundance multiplier, choose from <'binary', 'integer'>.'binary': 1 if the substructure exists in the glycan, 0 if not;'integer': the occurrence of the substructure in the glycan.

Parameter	Default	Description
-r	'epitope'	The root substructure of the substructure network, choose from <'epitope', 'N', 'O', 'lactose', 'custom'>."epitope": run every possible monosaccharide is a root;'N': the root for N-glycan, <b>GlcNAc</b> ;'O': the root for O-glycan, <b>GalNAc</b> ;'lactose': set the root as lactose, <b>Gal(b1-4)Glc</b> ;'custom': set custom root. You need to write your custom root in glycoCT format to a txt file and specify the file directory in -cr.
-u	"	The file directory to the txt file containing the custom root in glycoCT format. Only specify this if -r is set to 'custom'.
-d	None	Add this parameter if you want to draw the cluster map based on the output motif abundance table.

Parameter	Default	Description
-i	None	Add this parameter if you want to ignore unrecognized glycan structures and proceed the rest.

### Composition data

```
python glyCompare.py composition -a <ABUNDANCE TABLE> -v <VARIABLE ANNOTATION>
-o <OUTPUT_DIRECTORY> [-n <NORMALIZATION_MODE>, -i]
```

Required arguments:

Parameter	Description
-a	The file directory to the abundance table, in csv format
-v	The file directory to the variable annotation table, in csv format
-o	The directory to save the outputs, folder

Optional arguments:

Parameter	Default	Description
-n	'none'	Input glycans normalization within each glycoprofile, choose from <'none', 'min-max', 'prob-quot'>.'none': no normalization;'min-max': each element x
-i	None	Add this parameter if you want to ignore unrecognized glycan compositions and proceed the rest.