Glycobiology, 2023, **33**, 11, 927–934
https://doi.org/10.1093/glycob/cwad063
Advance access publication date 27 July 2023
Communication

GlycoDraw: a python implementation for generating high-quality glycan figures

Jon Lundstrøm[1,2], James Urban[1,2], Luc Thomès[1,2], Daniel Bojar[1,2,*]

[1]Department of Chemistry and Molecular Biology, University of Gothenburg, Medicinaregatan 9C, 41390 Gothenburg, Västra Götaland, Sweden, [2]Wallenberg Centre for Molecular and Translational Medicine, University of Gothenburg, Medicinaregatan 9C, 41390 Gothenburg, Västra Götaland, Sweden

[*]Corresponding author: Wallenberg Centre for Molecular and Translational Medicine, University of Gothenburg, Medicinaregatan 9C, 41390 Gothenburg, Västra Götaland, Sweden. Email: daniel.bojar@gu.se

Glycans are essential to all scales of biology, with their intricate structures being crucial for their biological functions. The structural complexity of glycans is communicated through simplified and unified visual representations according to the Symbol Nomenclature for Glycans (SNFGs) guidelines adopted by the community. Here, we introduce GlycoDraw, a Python-native implementation for high-throughput generation of high-quality, SNFG-compliant glycan figures with flexible display options. GlycoDraw is released as part of our glycan analysis ecosystem, glycowork, facilitating integration into existing workflows by enabling fully automated annotation of glycan-related figures and thus assisting the analysis of e.g. differential abundance data or glycomics mass spectra.

Key words: glycan; glycobiology; python; SNFG; visualization.

Introduction

Glycans play a crucial role in the biology of all organisms. Structural complexity arises from the presence of a wide range of monosaccharides that can exist in different anomeric configurations and be connected through various linkages. Furthermore, unlike linear biological sequences such as DNA or protein, glycans usually contain branched structures (Varki 2017).

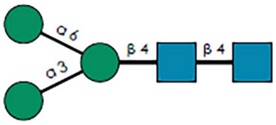
When choosing a nomenclature for representing glycan structures, researchers must compromise between achieving chemical accuracy while ensuring ease of interpretation, balancing the trade-offs, and advantages of each approach. As chemical compounds, all glycans can be described by the International Union of Pure and Applied Chemistry (IUPAC) notation (McNaught 1997). However, this notation proves impractical for daily use, due to the verbosity of the language. Other commonly used glycan nomenclatures include GlycoCT (Herget et al. 2008) and Web3 Unique Representation of Carbohydrate Structures (WURCS) (Tanaka et al. 2014; Matsubara et al. 2017) that were both developed with the aim of optimized machine readability in mind. In contrast, IUPAC-condensed, a simplified version of the standard IUPAC notation, offers excellent human readability and remains widely used for easy communication of glycan structures (McNaught 1997). Furthermore, glycans in the IUPAC-condensed nomenclature can even directly be utilized in computational workflows with glycowork (Thomès et al. 2021), where they are converted into graphs in the back-end, or processed on-the-fly into SMILES strings using software such as GlyLES (Joeres et al. 2023), for applications where exact chemical information is beneficial.

Still, for larger and more complicated glycan structures, the IUPAC-condensed notation is not immediately trivial to interpret, particularly for non-experts. To alleviate this issue, the glycobiology community has decided on standardized guidelines for graphical representation of glycan structures to promote efficient communication. The Symbol Nomenclature for Glycans (SNFGs) represents commonly occurring monosaccharides via a combination of colors and shapes, including rules for their assembly into glycan structures (Varki et al. 2015; Neelamegham et al. 2019). A comparative overview between the different mentioned glycan nomenclatures is shown in Table 1.

In recent years, several tools facilitating the generation of glycan figures have been published [exhaustively reviewed in (Lal et al. 2020) and summarized in Table 2]. A major focus of available software has been the development of easy-to-use tools with beginner-friendly graphical user interfaces (GUIs). Such GUI-based "click-and-drag" implementations facilitate easy construction of complex structures but are limited with regards to throughput, due to the requirement for manual input. Alternatively, tools such as CSDB/SNFG Editor (Bochkov and Toukach 2021), DrawGlycan-SNFG (Cheng et al. 2017, 2020), or GlycoGlyph (Mehta and Cummings 2020) allow for the generation of glycan figures directly from text (e.g. GlycoCT- or IUPAC-formatted glycans). However, web-based implementations still limit automatability and scalability, if they are not used via command line interfaces, which in turn requires some level of computational proficiency, especially when having to integrate different interfaces. Programmatical implementations exist e.g. the Python integration of DrawGlycan-SNFG or glypy (Klein and Zaia 2019), but existing workflow integration possibilities, such as automatic annotation of data figures, remain limited. In summary, we conclude that there are substantial benefits to filling the current need for a scalable, programmatic,

Received: May 20, 2023. **Revised:** July 14, 2023. **Accepted:** July 26, 2023

Table 1. Comparison of glycan nomenclatures.

Nomenclature	
SNFG	
IUPAC-condensed	Man(a1-3)[Man(a1-6)]Man(b1-4)GlcNAc(b1-4)GlcNAc
IUPAC	alpha-D-manno-hexopyranosyl-(1- > 3)-[alpha-D-manno-hexopyranosyl-(1- > 6)]-beta-D-manno-hexopyranosyl-(1- > 4)-2-acetamido-2-deoxy-beta-D-gluco-hexopyranosyl-(1- > 4)-2-acetamido-2-deoxy-D-gluco-hexopyranose
SMILES	CC(=O)N[C@@H]1[C@H]([C@@H]([C@H](O[C@H]1O[C@@H]2[C@H](OC([C@@H]([C@H]2O)NC(=O)C)O)CO)CO)O[C@H]3[C@H]([C@H]([C@@H]([C@H](O3)CO[C@@H]4[C@H]([C@H]([C@@H]([C@H](O4)CO)O)O)O)O)O[C@@H]5[C@H]([C@H]([C@@H]([C@H](O5)CO)O)O)O)O)O
WURCS	WURCS = 2.0/4,5,4/[a2122h-1x_1-5_2*NCC/3 = O][a2122h-1b_1-5_2*NCC/3 = O][a1122h-1b_1-5][a1122 h-1a_1-5]/1-2-3-4-4/a4-b1_b4-c1_c3-d1_c6-e
GlycoCT	RES 1b:x-dglc-HEX-1:5 2 s:n-acetyl 3b:b-dglc-HEX-1:5 4 s:n-acetyl 5b:b-dman-HEX-1:5 6b:a-dman-HEX-1:5 7b:a-dman-HEX-1:5 LIN 1:1d(2 + 1)2n 2:1o(4 + 1)3d 3:3d(2 + 1)4n 4:3o(4 + 1)5d 5:5o(3 + 1)6d 6:5o(6 + 1)7d

high-quality implementation that can be used in a plug'n'play manner to augment existing workflows.

To overcome these limitations and realize the potential for this computational aspect of glycobiology, we set out to develop a Python-native implementation for high-throughput generation of publication-ready glycan figures, as vector graphics, according to the SNFG guidelines. Here, we describe GlycoDraw, its use in the automation of drawing glycan figures, and integration into existing workflows. GlycoDraw is available as a part of glycowork version 0.7, a Python-based ecosystem for glycan analysis we developed and maintain. This allows for a seamless integration with glycowork analysis workflows, without the need for engineering interfaces or using command line interfaces.

Results

The functionality of GlycoDraw is based on the computational framework of glycowork and the graphics engines of pycairo (version 1.23.0) and drawsvg (version 2.1.1). Glycan inputs in the IUPAC-condensed nomenclature (see Supplementary Note 1 for best practices) are converted to graph objects for calculating coordinates to place individual monosaccharides in relation to each other. The full glycan structure is split into the main chain (i.e. longest chain), branches, and nested branches, while connectivity information is retained based on the glycan graph object. In re-assembling the full glycan as a figure, each individual component is first placed horizontally at the correct x position, as described below, with subsequent adjustment to achieve the proper y position (Fig. 1A).

For the x position, the reducing end (initiating) monosaccharide of the main chain is placed at (0,0) in a coordinate system extending up and to the left. The remaining monosaccharides of the main chain are placed with an incrementally increasing x position (x + 1, 0). For branches, the horizontal positioning is calculated in a similar manner, with the coordinates of the initiating monosaccharide being x + 1, where x is the position of the monosaccharide that the branch is connected to. For each level of branching, the y position is increased by 1. Next, the y positions are fine-tuned in 3 steps: (i) any overlap caused by perpendicular monosaccharides (e.g. fucose or xylose) is fixed, (ii) at each branch point, the hub monosaccharide is placed centrally between the branch connections, and (iii) extra space between branches introduced by step (ii) is minimized where possible, to have no more than 1 unit of distance between branches (Fig. 1A). Finally, the glycan figure is rendered with drawsvg, where monosaccharide icons and chemical modifications are shown according to the guidelines described by the SNFG (described below). GlycoDraw supports inline viewing of the resulting figure when running in a Jupyter Notebook environment, in addition to directly saving figures as pdf or svg files.

In contrast to existing software, GlycoDraw is ideally suited for high-throughput and automatic generation of glycan figures, or situations where the user wishes to integrate figure generation into a computational workflow. Although the drawing algorithm scales non-linearly with glycan size and -complexity, the runtime for an average glycan remains around 2 ms, allowing for near-instantaneous rendering of hundreds of structures (Fig. 1B and C). In testing GlycoDraw on the entirety of SugarBase, the glycowork-internal database of nearly 50,000 glycans, we achieved a drawing coverage of 99.5%, only failing in the case of the presence of non-standard monosaccharides not defined in the SNFG. We further tested the performance of GlycoDraw by selecting 100 random

Table 2. Feature summary of GlycoDraw and other commonly used glycan visualization tools.

	GlycoDraw	DrawGlycan-SNFG	GlycanBuilder2	GlycoGlyph	glypy	SugarSketcher
Interface	Command-line interface (CLI)	CLI, standalone & web GUI	Standalone GUI	Web GUI	CLI	Web GUI
Language	Python	MATLAB	Java	JavaScript	Python	JavaScript
Input	IUPAC-condensed, GlycoCT[a], WURCS[a]	IUPAC-condensed	BCSDB, CabosML, CarbBank, GlycoCT, GlycosuiteDB, GLYDE-II, LinearCode, LINUCS, IUPAC-condensed	Manual building, modified IUPAC-condensed	GlycoCT, LinearCode, IUPAC-condensed, WURCS	GlycoCT, manual building
Output	pdf, svg	jpg, with CLI: bmp, emf, eps, png, pdf, svg, tif	GlycoCT, GLYDE-II, LinearCode, LINUCS, WURCS, bmp, jpg, png, pdf	GlycoCT, svg	GlycoCT, LinearCode, IUPAC-condensed, WURCS, jpg, png, pdf, svg	GlycoCT, svg
Ambiguous structures	Yes	Yes	Yes	No	No	No
Code available	Yes	Yes	Yes	Yes	Yes	Yes
Motif templates	Yes	No	Yes	Yes	Yes	Yes
MS fragments	Yes	Yes	No	No	Yes	No
Reference	This work	(Cheng et al. 2017, 2020)	(Tsuchiya et al. 2017)	(Mehta and Cummings 2020)	(Klein and Zaia 2019)	(Alocci et al. 2018)

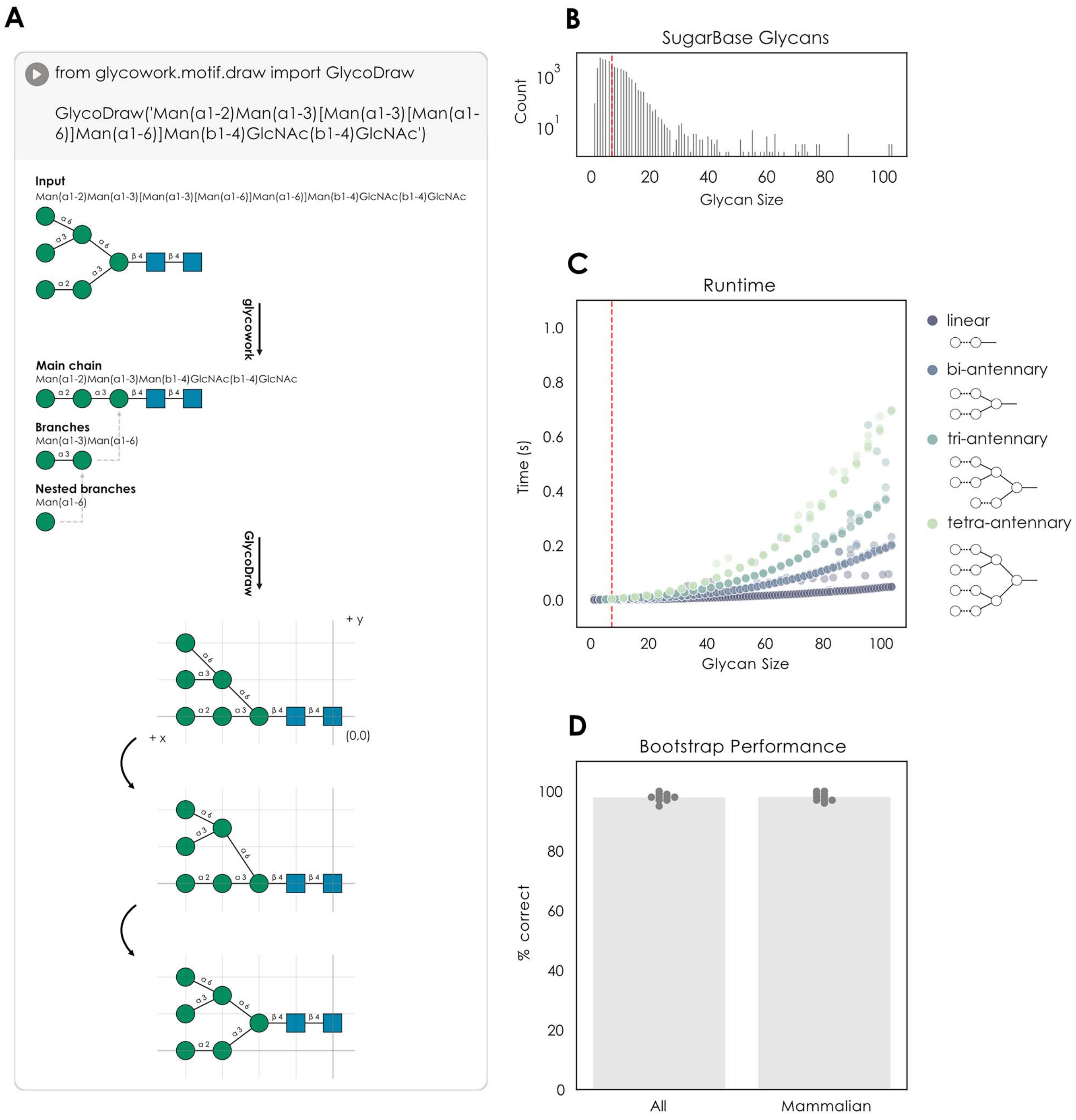
[a] With nomenclature conversion using glypy.

glycans from SugarBase and manually evaluating the results for correct formatting. Averaged over 10 iterations (i.e. 1,000 glycans in total), the structure is drawn correctly in 97.9% of cases for all glycans and 98.1% of cases for mammalian glycans (Fig. 1D), with the rare occurrence of minor aesthetical deviations in particularly exotic or complex structures.

As GlycoDraw is implemented within glycowork, it requires Python 3.8+. The GlycoDraw input format is IUPAC-condensed, and the software can handle all canonical SNFG-defined monosaccharides (Fig. 2A). For broad applicability, we also included the *canonicalize_iupac* function (glycowork.motif.processing; version 0.7), which is able to streamline many irregularities or non-canonical formatting of IUPAC-condensed glycans. In addition to the standard glycan figure nomenclature as defined by the SNFG, GlycoDraw supports rendering of glycan fragment ions following the Domon–Costello nomenclature as described in Domon and Costello (1988) and implemented in GlycoWorkBench (Ceroni et al. 2008) (Fig. 2A). In combination with in silico fragmentation, this allows for automated annotation of experimental mass spectra as described below. Monosaccharide modifications are shown according to SNFG guidelines; a furanose ring configuration is indicated by an italicized *f* in the center of the monosaccharide symbol. Likewise, the enantiomeric configuration (D or L) can be shown in the same manner. Chemical carbohydrate modifications are indicated above the monosaccharide symbol, in text, by the number of the modified carbon (or, if unknown, the atom linking the modification), followed by the abbreviated chemical substituent (Fig. 2A).

GlycoDraw supports the drawing of a list of predefined glycan structures and -motifs by name (e.g. "LewisX"; *motif_list*, imported from glycowork.glycan_data.loader, contains the full list of possible structures) (Fig. 2B). Furthermore, several optional arguments are available to modify the resulting glycan figure. "vertical = True" will draw the glycan in a portrait format rather than the default horizontal format. "show_linkage = False" will hide all linkage information and display a skeletonized version of the glycan backbone, while "compact = True" additionally minimizes the bond length to show the structure as compact as possible (Fig. 2B). Ambiguous glycan structures e.g. when fucose is present but the exact location in the glycan is unknown, or sialylation of one of multiple branches in a complex *N*-glycan, are also supported by GlycoDraw. The ambiguous part of the structure is enclosed within curly brackets "{Fuc(a1-3)}" and written before the glycan. Best practices dictate that the ordering of multiple ambiguous components follows that of branch ordering i.e. longest > lowest linkage > alphabetical (Fig. 2B). The output of GlycoDraw can be viewed inline when working in a notebook format e.g. Google Colaboratory or Jupyter Notebooks. Image files can be saved by specifying "output = 'filename.pdf'". Files can be exported as pdf or svg by indicating the appropriate file extension in the name (Fig. 2B), and can be freely edited as vector graphics in downstream applications. A quick-start guide describing the installation and general use of GlycoDraw can be found online in the documentation of glycowork (https://bojarlab.github.io/glycowork/).

The Python implementation of glycan figure generation allows for easy integration with existing glycan analysis and processing workflows. With GlycoDraw, any list of glycans can be effortlessly converted into figures and saved as high-quality vector graphics. Furthermore, automated workflows can be established, in which glycan-related svg figures generated using the Python matplotlib library are modified to include glycan graphics instead of text labels by using the *annotate_figure* function (glycowork.motif.draw; version 0.7) (Fig. 3A). The integration of GlycoDraw with example workflows is described below.



```
from glycowork.motif.draw import GlycoDraw

GlycoDraw('Man(a1-2)Man(a1-3)[Man(a1-3)[Man(a1-
6)]Man(a1-6)]Man(b1-4)GlcNAc(b1-4)GlcNAc')
```

Fig. 1. GlycoDraw algorithm and characterization. A) Overview of the drawing algorithm. A glycan string in IUPAC-condensed nomenclature is first split between main chain, branches, and nested branches. Then, monosaccharide coordinates are calculated, and the final figure is rendered. B) Size distribution of glycans in the SugarBase database. The red dotted line indicates the average glycan size (in number of monosaccharides) C) runtime of GlycoDraw with different glycan topologies and sizes. An average glycan (red dotted line) is rendered in ~2 ms. D) Bootstrap performance of GlycoDraw; 100 glycans from SugarBase (either the full database or limited to mammalian glycans) were selected at random, visualized with GlycoDraw, and the results manually evaluated. This was repeated 10 times for each group of glycan structures.

Glycowork contains functions to analyze glycan abundance and visualize the results with heatmaps. With the version 0.7 update introducing GlycoDraw, we additionally included functionality to modify such heatmaps by automatic annotation with glycan figures instead of text labels (Fig. 3B). As an alternative to heatmaps, differential glycan (motif) abundance can be visualized with volcano plots, as widely used in the field of RNA-seq and differential expression analysis. Our implementation additionally allows for annotation of volcano plots by adding glycan figures to selected data points, scaled by significance, resulting in easy visual interpretation (Fig. 3C).

Another example of the benefit of automatable glycan figure generation includes the rendering of biosynthetic networks, as described in Thomès et al. (2023). With glycowork, theoretical biosynthetic networks including all possible intermediate structures can be generated based on the observed glycan abundance of an analyzed sample. Using GlycoDraw, such networks can be annotated to include glycan figures. In the example graph (Fig. 3D), each substructure is shown as the network nodes, and the enzymatic reaction is indicated on the edge connecting 2 intermediates.

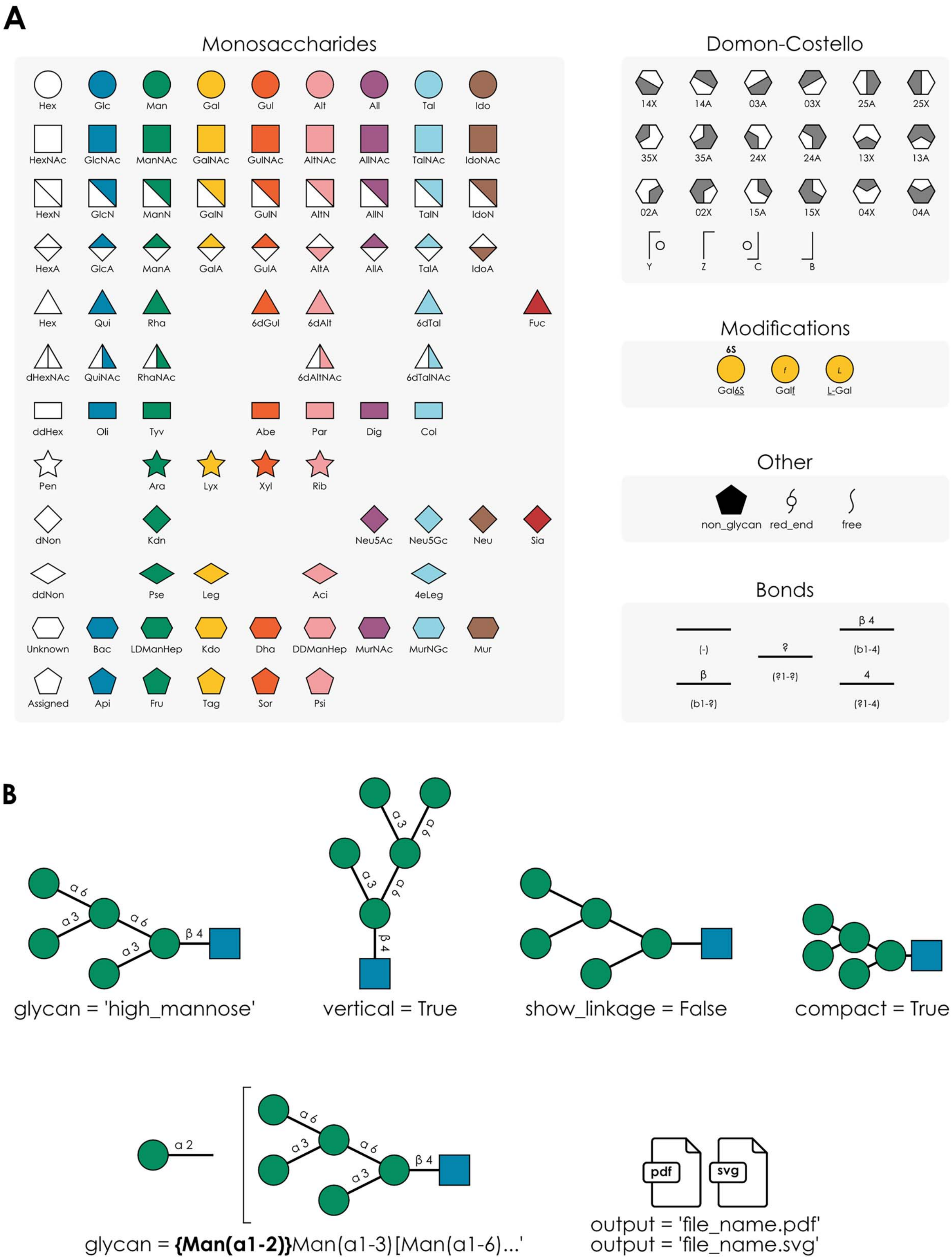


Fig. 2. GlycoDraw functionality. A) Overview of SNFG monosaccharide and Domon–Costello fragment icons supported by GlycoDraw. Modifications are shown with the example of galactose and extend to any monosaccharide. Bonds are shown with the example of beta-linkage. B) Optional arguments to GlycoDraw allow modification of the output as well as saving the figure as an image file.

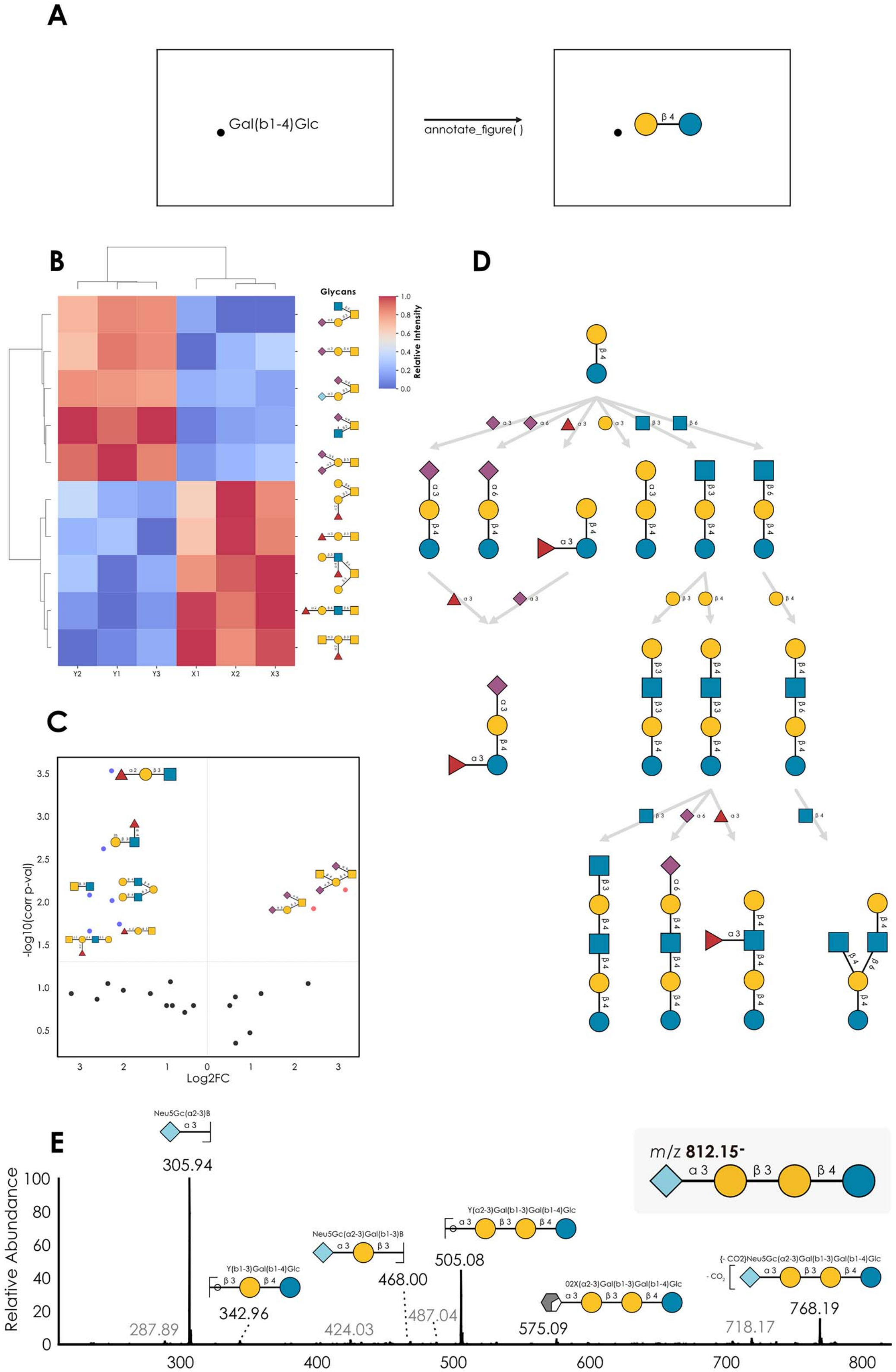


Fig. 3. Example applications of GlycoDraw. A) The *annotate_figure* function modifies matplotlib-generated svg figures by replacing text labels with glycan images B) annotated heatmap visualization of glycan abundance between 2 groups based on simulated example data. C) Annotated volcano plot visualization of differential glycan abundance quantification based on simulated example data. D) Example milk oligosaccharide biosynthetic network showing the relation between enzymatic reactions and selected structures. E) MS/MS spectrum of Neu5Gc(a2-3)gal(b1-3)gal(b1-4)Glc from (Jin et al. 2023), annotated with CandyCrumbs (Urban et al. 2023), showing the integration of the Domon–Costello nomenclature with GlycoDraw. Low-abundance peaks in grey were not assigned any fragment structure. Code for generating these figures can be found in Supplementary Note 2.

The analysis of glycomics mass spectrometry data routinely includes expert assignment and manual annotation of spectra and fragment mass peaks. For this purpose, glycan fragments are usually annotated using the Domon–Costello nomenclature, a text-based system that can be visualized by specific icons that represent possible fragmentation ions. In addition to standard IUPAC-condensed, GlycoDraw supports this extended fragmentation nomenclature. An example workflow would include computationally obtaining all possible fragments of a specific glycan, cross-checking their masses against a list of observed mass peaks from a glycomics experiment, and generating the corresponding figures where glycan text labels are replaced by glycan images for automatable annotation of glycomics mass spectra (Fig. 3E).

Discussion

With the introduction of GlycoDraw, we present a Python-native implementation for scalable, high-throughput generation of high-quality, SNFG-compliant glycan figures. GlycoDraw has several advantages over existing tools for generating glycan figures, including scalability for high-throughput generation of images, and importantly, our Python implementation additionally allows for integration into existing workflows, such as the annotation of glycan-related figures.

Although we use IUPAC-condensed inputs as the preferred glycan nomenclature in our workflows, we recognize the widespread usage of WURCS and GlycoCT in the community. As a workaround, we refer users to already existing functionality to convert between glycan nomenclatures online (glycosmos.org/glycans/converters, Tsuchiya et al. 2019) or in Python using glypy (Klein and Zaia 2019). As mentioned previously, within different IUPAC "dialects," such as the formatting of glycan sequences in the CFG array database (www.functionalglycomics.org/static/consortium/CFGnomenclature.pdf), the restructured *canonicalize_iupac* function within glycowork version 0.7 can be used to harmonize inputs. Furthermore, although GlycoDraw functions as expected for the vast majority of glycan structures, we note the existence of very rare cases, particularly exotic/highly complex structures, where manual intervention may be necessary. Fortunately, in such cases, the glycan images can be exported as vector graphics and easily adjusted with appropriate image editing software.

GlycoDraw is a valuable tool for generating publication-ready glycan figures that comply with SNFG guidelines. Its scalability and programmability make it ideal for high-throughput analysis. We will continue to develop GlycoDraw to improve its capabilities and usability, and we envision that it will contribute to advancing the field of glycobiology.

Author contributions

Conceptualization: JL, DB, Funding Acquisition: DB, Resources: DB, Software: JL, JU, LT, DB, Supervision: DB, Visualization: JL, JU, LT, DB, Writing—Original Draft Preparation: JL, DB, Writing—Review & Editing: JL, JU, LT, DB.

CRediT author statement

Jon Lundstrøm (Conceptualization [equal], Software [lead], Visualization [lead], Writing—original draft [lead], Writing—review & editing [lead]), James Urban (Software [supporting], Visualization [supporting], Writing—review & editing [supporting]), Luc Thomès (Software [supporting], Visualization [supporting], Writing—review & editing [supporting]), and Daniel Bojar (Conceptualization [equal], Funding acquisition [lead], Resources [lead], Software [supporting], Supervision [lead], Visualization [supporting], Writing—original draft [supporting], Writing—review & editing [supporting]).

Supplementary material

Supplementary material is available at *Glycobiology Journal* online.

Funding

This work was funded by a Branco Weiss Fellowship—Society in Science awarded to DB, by the Knut and Alice Wallenberg Foundation, and the University of Gothenburg, Sweden.

Conflict of interest statement. The authors declare no competing interests.

Data availability

All used code and data can be found at https://github.com/BojarLab/glycowork/, with particular emphasis on the motif.draw module.

References

Alocci D, Suchánková P, Costa R, Hory N, Mariethoz J, Vařeková RS, Toukach P, Lisacek F. SugarSketcher: quick and intuitive online glycan drawing. *Molecules*. 2018:**23**(12):3206. https://doi.org/10.3390/molecules23123206.

Bochkov AY, Toukach PV. CSDB/SNFG structure editor: an online glycan builder with 2D and 3D structure visualization. *J Chem Inf Model*. 2021:**61**(10):4940–4948. https://doi.org/10.1021/acs.jcim.1c00917.

Ceroni A, Maass K, Geyer H, Geyer R, Dell A, Haslam SM. GlycoWorkbench: a tool for the computer-assisted annotation of mass spectra of Glycans. *J Proteome Res*. 2008:**7**(4):1650–1659. https://doi.org/10.1021/pr7008252.

Cheng K, Zhou Y, Neelamegham S. DrawGlycan-SNFG: a robust tool to render glycans and glycopeptides with fragmentation information. *Glycobiology*. 2017:**27**(3):200–205. https://doi.org/10.1093/glycob/cww115.

Cheng K, Pawlowski G, Yu X, Zhou Y, Neelamegham S. DrawGlycan-SNFG and gpAnnotate: rendering glycans and annotating glycopeptide mass spectra. *Bioinformatics*. 2020:**36**(6):1942–1943. https://doi.org/10.1093/bioinformatics/btz819.

Domon B, Costello CE. A systematic nomenclature for carbohydrate fragmentations in FAB-MS/MS spectra of glycoconjugates. *Glycoconj J*. 1988:**5**(4):397–409. https://doi.org/10.1007/BF01049915.

Herget S, Ranzinger R, Maass K, Lieth C-WVD. GlycoCT-a unifying sequence format for carbohydrates. *Carbohydr Res*. 2008:**343**(12): 2162–2171. https://doi.org/10.1016/j.carres.2008.03.011.

Jin C, Lundstrøm J, Korhonen E, Luis AS, Bojar D. Breast milk oligosaccharides contain immunomodulatory glucuronic acid and LacdiNAc. *bioRxiv*. 2023. https://doi.org/10.1101/2023.01.16.524336.

Joeres R, Bojar D, Kalinina OV. GlyLES: grammar-based parsing of glycans from IUPAC-condensed to SMILES. *J Chem*. 2023:**15**(1):37. https://doi.org/10.1186/s13321-023-00704-0.

Klein J, Zaia J. Glypy: an open source glycoinformatics library. *J Proteome Res*. 2019:**18**(9):3532–3537. https://doi.org/10.1021/acs.jproteome.9b00367.

Lal K, Bermeo R, Perez S. Computational tools for drawing, building and displaying carbohydrates: a visual guide. *Beilstein J Org Chem*. 2020:**16**:2448–2468. https://doi.org/10.3762/bjoc.16.199.

Matsubara M, Aoki-Kinoshita KF, Aoki NP, Yamada I, Narimatsu H. WURCS 2.0 update to encapsulate ambiguous carbohydrate structures. *J Chem Inf Model*. 2017:**57**(4):632–637. https://doi.org/10.1021/acs.jcim.6b00650.

McNaught AD. Nomenclature of carbohydrates (recommendations 1996). *Adv Carbohydr Chem Biochem*. 1997:**52**:43–177.

Mehta AY, Cummings RD. GlycoGlyph: a glycan visualizing, drawing and naming application. *Bioinformatics*. 2020:**36**(11):3613–3614. https://doi.org/10.1093/bioinformatics/btaa190.

Neelamegham S, Aoki-Kinoshita K, Bolton E, Frank M, Lisacek F, Lütteke T, O'Boyle N, Packer NH, Stanley P, Toukach P, et al. Updates to the symbol nomenclature for glycans guidelines. *Glycobiology*. 2019:**29**(9):620–624. https://doi.org/10.1093/glycob/cwz045.

Tanaka K, Aoki-Kinoshita KF, Kotera M, Sawaki H, Tsuchiya S, Fujita N, Shikanai T, Kato M, Kawano S, Yamada I, et al. WURCS: the Web3 unique representation of carbohydrate structures. *J Chem Inf Model*. 2014:**54**(6):1558–1566. https://doi.org/10.1021/ci400571e.

Thomès L, Burkholz R, Bojar D. Glycowork: a python package for glycan data science and machine learning. *Glycobiology*. 2021:**31**(10): 1240–1244. https://doi.org/10.1093/glycob/cwab067.

Thomès L, Karlsson V, Lundstrøm J, Bojar D. Mammalian milk glycomes: connecting the dots between evolutionary conservation and biosynthetic pathways. *Cell Rep*. 2023:**42**(7):112710. https://doi.org/10.1016/j.celrep.2023.112710.

Tsuchiya S, Aoki NP, Shinmachi D, Matsubara M, Yamada I, Aoki-Kinoshita KF, Narimatsu H. Implementation of GlycanBuilder to draw a wide variety of ambiguous glycans. *Carbohydr Res*. 2017:**445**:104–116. https://doi.org/10.1016/j.carres.2017.04.015.

Tsuchiya S, Yamada I, Aoki-Kinoshita KF. GlycanFormatConverter: a conversion tool for translating the complexities of glycans. *Bioinformatics*. 2019:**35**(14):2434–2440. https://doi.org/10.1093/bioinformatics/bty990.

Urban J, Jin C, Thomsson KA, Karlsson NG, Ives CM, Fadda E, Bojar D. Predicting glycan structure from tandem mass spectrometry via deep learning. *bioRxiv*. 2023. https://doi.org/10.1101/2023.06.13.544793.

Varki A. Biological roles of glycans. *Glycobiology*. 2017:**27**(1):3–49. https://doi.org/10.1093/glycob/cww086.

Varki A, Cummings RD, Aebi M, Packer NH, Seeberger PH, Esko JD, Stanley P, Hart G, Darvill A, Kinoshita T, et al. Symbol nomenclature for graphical representations of glycans. *Glycobiology*. 2015:**25**(12):1323–1324. https://doi.org/10.1093/glycob/cwv091.