

Package: glydraw (via r-universe)

July 14, 2026

Title Draw Beautiful Symbol Nomenclature for Glycans

Version 0.6.3

Description A 'ggplot2'-native plotting engine for drawing reproducible beautiful Symbol Nomenclature for Glycans (SNFG) glycan cartoons from glycan structure objects or text notations, with support for batch export, structural highlighting, and deep appearance customization. It follows the SNFG specification described at <https://www.ncbi.nlm.nih.gov/glycans/snfg.html>.

License MIT + file LICENSE

Suggests knitr, rmarkdown, testthat (>= 3.0.0), vdiffr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.3

URL <https://glycoverse.github.io/glydraw/>,
<https://github.com/glycoverse/glydraw>

Imports glyrepr (>= 0.10.0), rlang, dplyr, ggplot2 (>= 4.0.0), igraph, purrr, png, glyparse, cli, checkmate, scales, fs

VignetteBuilder knitr

BugReports <https://github.com/glycoverse/glydraw/issues>

NeedsCompilation no

Author Bin Fu [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0001-8567-2997>>), Xisong Feng [aut]
(ORCID: <<https://orcid.org/0009-0000-4057-1295>>)

Maintainer Bin Fu <23110220018@m.fudan.edu.cn>

Depends R (>= 4.1.0)

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-14 16:30:02 UTC

RemoteUrl <https://github.com/cran/glydraw>

RemoteRef HEAD

RemoteSha d6ad911298db51979b555e1482f7f40eb1415fd3

Contents

draw_cartoon	2
export_cartoons	3
print.glydraw_cartoon	5
save_cartoon	6
Index	7

draw_cartoon	<i>Draw a Symbol Nomenclature For Glycan (SNFG)</i>
--------------	---

Description

Draw a Symbol Nomenclature For Glycan (SNFG)

Usage

```
draw_cartoon(  
    structure,  
    ...,  
    show_linkage = TRUE,  
    orient = c("H", "V"),  
    fuc_orient = c("flex", "up"),  
    red_end = "",  
    edge_linewidth = 0.8,  
    node_linewidth = 0.8,  
    node_size = 1,  
    colors = NULL,  
    highlight = NULL  
)
```

Arguments

structure	A glyrepr::glycan_structure() scalar, or a string of any glycan structure text nomenclatures supported by glyparse::auto_parse() .
...	Ignored.
show_linkage	Show glycosidic linkage annotations or not. Default is TRUE. Substituent annotations are always shown.
orient	The orientation of glycan structure. "H" for horizontal, "V" for vertical. Default is "H"
fuc_orient	Fuc-like triangle orientation. "flex" points non-reducing Fuc-like residues toward their rendered linkage direction, while "up" draws all Fuc-like triangles pointing upward. Reducing-end Fuc-like residues always point upward. Defaults to "flex".

red_end	Reducing-end annotation. The default "" keeps the current reducing-end line. Use "~" to add a wavy reducing end, or any other string to draw that string at the reducing end.
edge_linewidth	Numeric scalar controlling the linewidth of linkage lines. Defaults to the current value, 0.8.
node_linewidth	Numeric scalar controlling the linewidth of node borders. Defaults to the current value, 0.8.
node_size	Numeric scalar used as a multiplier for the default node size. Defaults to 1, which keeps the current size. Linkage annotations are moved farther from larger nodes, and are hidden with a warning when node_size is too large to leave enough annotation space. Values larger than 2 are rejected because residues overlap.
colors	Optional named character vector of custom monosaccharide fill colors. Names must be supported monosaccharide names, such as "Gal" or "GlcNAc". User-provided colors overwrite the default SNFG colors, while unprovided monosaccharides keep their default colors. Defaults to NULL.
highlight	An integer vector specifying the node indices to highlight. This argument is applicable only when structure is a <code>glyrepr::glycan_structure()</code> . Note that for a <code>glyrepr::glycan_structure()</code> , the node indices correspond exactly to the monosaccharides in its printed IUPAC nomenclature. For example, given <code>glyrepr::as_glycan_structure("Gal(b1-3)[GlcNAc(b1-6)]GalNAc(a1-")</code> , setting <code>highlight = c(1, 3)</code> will highlight the "Gal" and "GalNAc" nodes.

Value

a ggplot2 object

Examples

```
draw_cartoon("Gal(b1-3)GalNAc(a1-")
```

export_cartoons

Export all glycan structures to figures

Description

Draw and save one cartoon for each glycan structure in x.

Usage

```
export_cartoons(
  x,
  dirname,
  ...,
  file_ext = "png",
  dpi = 300,
```

```

scale = 1,
show_linkage = TRUE,
orient = c("H", "V"),
fuc_orient = c("flex", "up"),
red_end = "",
edge_linewidth = 0.8,
node_linewidth = 0.8,
node_size = 1,
colors = NULL
)

```

Arguments

x	A <code>glyrepr::glycan_structure()</code> vector, or a character vector of any glycan structure text nomenclatures supported by <code>glyparse::auto_parse()</code> .
dirname	Directory name to save the cartoons. If it does not exist, it is created.
...	Ignored.
file_ext	File extension supported by <code>ggplot2::ggsave()</code> . Defaults to "png".
dpi	Deprecated and ignored. Use <code>scale</code> to change the output size.
scale	Numeric output-size multiplier passed to <code>save_cartoon()</code> .
show_linkage	Show glycosidic linkage annotations or not. Default is TRUE. Substituent annotations are always shown.
orient	The orientation of glycan structure. "H" for horizontal, "V" for vertical. Default is "H"
fuc_orient	Fuc-like triangle orientation. "flex" points non-reducing Fuc-like residues toward their rendered linkage direction, while "up" draws all Fuc-like triangles pointing upward. Reducing-end Fuc-like residues always point upward. Defaults to "flex".
red_end	Reducing-end annotation. The default "" keeps the current reducing-end line. Use "~" to add a wavy reducing end, or any other string to draw that string at the reducing end.
edge_linewidth	Numeric scalar controlling the linewidth of linkage lines. Defaults to the current value, 0.8.
node_linewidth	Numeric scalar controlling the linewidth of node borders. Defaults to the current value, 0.8.
node_size	Numeric scalar used as a multiplier for the default node size. Defaults to 1, which keeps the current size. Linkage annotations are moved farther from larger nodes, and are hidden with a warning when <code>node_size</code> is too large to leave enough annotation space. Values larger than 2 are rejected because residues overlap.
colors	Optional named character vector of custom monosaccharide fill colors. Names must be supported monosaccharide names, such as "Gal" or "GlcNAc". User-provided colors overwrite the default SNFG colors, while unprovided monosaccharides keep their default colors. Defaults to NULL.

Value

The function returns the list of cartoons implicitly.

File names

IUPAC-condensed nomenclatures are used as file names. If `x` is a named character vector or named `glyrepr::glycan_structure()` vector, the vector names are used as file names.

Why not width and height?

The familiar `ggplot2::ggsave()` interface uses width, height, and dpi because ordinary ggplot2 plots are drawn into a user-chosen device size. glydraw cartoons are different: the natural width and height are calculated from the glycan structure so residues, linkages, labels, and borders stay comparable across different glycans. If users supplied arbitrary width and height, glydraw would either distort that structure-derived layout or need to guess how to reconcile one requested size with the other.

dpi is also not the right control here because changing it alters how point- and inch-based ggplot2 elements are rasterized relative to the fixed cartoon canvas. glydraw therefore keeps an internal fixed design scale and uses `scale` as a single multiplier for the final pixel dimensions. This preserves the cartoon's aspect ratio and relative appearance while still allowing larger or smaller output files.

Examples

```
export_cartoons(
  c(
    "Man(a1-3)Man(b1-4)GlcNAc(b1-",
    "Gal(b1-4)GlcNAc(b1-"
  ),
  tempdir()
)
```

```
print.glydraw_cartoon  Print glycan cartoon
```

Description

Print glycan cartoon

Usage

```
## S3 method for class 'glydraw_cartoon'
print(x, ..., newpage = is.null(vp), vp = NULL)
```

Arguments

<code>x</code>	A ggplot2 object returned by <code>draw_cartoon()</code> .
<code>...</code>	Ignored.
<code>newpage</code>	Draw the plot on a new page.
<code>vp</code>	A grid viewport object or viewport name.

Value

The original glycan cartoon, invisibly.

save_cartoon	<i>Save fixed-size glycan cartoon image to local device.</i>
--------------	--

Description

This function saves the glycan cartoon to a file, with a suitable size.

Usage

```
save_cartoon(cartoon, file, ..., dpi = 300, scale = 1)
```

Arguments

cartoon	A ggplot2 object returned by draw_cartoon() .
file	File name of glycan cartoon.
...	Ignored.
dpi	Deprecated and ignored. Use scale to change the output size.
scale	Numeric output-size multiplier. The default 1 saves the cartoon at its natural fixed size; 2 saves the same cartoon with twice the pixel width and height.

Value

Path of the saved cartoon file.

Why not width and height?

The familiar `ggplot2::ggsave()` interface uses width, height, and dpi because ordinary ggplot2 plots are drawn into a user-chosen device size. glydraw cartoons are different: the natural width and height are calculated from the glycan structure so residues, linkages, labels, and borders stay comparable across different glycans. If users supplied arbitrary width and height, glydraw would either distort that structure-derived layout or need to guess how to reconcile one requested size with the other.

dpi is also not the right control here because changing it alters how point- and inch-based ggplot2 elements are rasterized relative to the fixed cartoon canvas. glydraw therefore keeps an internal fixed design scale and uses scale as a single multiplier for the final pixel dimensions. This preserves the cartoon's aspect ratio and relative appearance while still allowing larger or smaller output files.

Examples

```
cartoon <- draw_cartoon("Gal(b1-3)GalNAc(a1-")
save_cartoon(cartoon, tempfile(fileext = ".png"))
```

Index

`draw_cartoon`, [2](#)
`draw_cartoon()`, [5](#), [6](#)
`export_cartoons`, [3](#)
`ggplot2::ggsave()`, [4-6](#)
`glyparse::auto_parse()`, [2](#), [4](#)
`glyrepr::glycan_structure()`, [2-5](#)
`print.glydraw_cartoon`, [5](#)
`save_cartoon`, [6](#)
`save_cartoon()`, [4](#)