

Package ‘glydraw’

July 25, 2026

Title Draw Beautiful Symbol Nomenclature for Glycans

Version 0.7.0

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 ggraph, knitr, rmarkdown, testthat (>= 3.0.0), tibble, vdiff

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

Depends R (>= 4.1)

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>

Repository CRAN

Date/Publication 2026-07-25 05:00:02 UTC

Contents

draw_cartoon	2
export_cartoons	3
geom_glycan	6
geom_node_glycan	9
glycanGrob	10
glydraw_style	11
guide_glycan	12
hjust_red_end	15
print.glydraw_cartoon	16
save_cartoon	16
scale_x_glycan	17
Index	20

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,  
  style = 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.
style	A <code>glydraw_style</code> object that supplies rendering options. Explicitly supplied rendering arguments override it.

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,
  style = 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 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.
style	A glydraw_style object that supplies rendering options. Explicitly supplied rendering arguments override it.

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()
)
```

geom_glycan

*Draw glycans at ggplot2 positions***Description**

geom_glycan() draws one glycan cartoon for each data row. Each cartoon is anchored at its mapped x and y position and retains the structure-derived dimensions and appearance used by draw_cartoon(). The optional size aesthetic scales the complete cartoon uniformly, including nodes, lines, text, and spacing, without changing their relative appearance. Like points and text, the cartoons do not expand the position scales beyond their anchor coordinates. Use scale expansion or explicit coordinate limits when the cartoons need more room around the panel edges. Unlike standalone cartoons returned by draw_cartoon(), cartoons in this layer have no output border or background.

Usage

```
geom_glycan(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  angle = 0,
  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,
  style = NULL,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by <code>ggplot2::aes()</code> . The x, y, and structure aesthetics are required.
data	The data to be displayed in this layer. When NULL, the default, the data is inherited from the plot data.
stat	The statistical transformation to use on the data for this layer. Defaults to "identity".

position	A position adjustment to use on the data for this layer. Defaults to "identity".
...	Other arguments passed to <code>ggplot2::layer()</code> .
angle	Rotation in degrees. Like <code>ggplot2::geom_text()</code> , this can be supplied as an aesthetic or a fixed layer value. It rotates each completed cartoon around its mapped position independently of orient. Defaults to 0.
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.
style	A <code>glydraw_style</code> object that supplies rendering options. Explicitly supplied rendering arguments override it.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	Logical. Should this layer be included in the legends?
inherit.aes	If FALSE, overrides the default aesthetics rather than combining with them.

Value

A `ggplot2` layer that can be added to a `ggplot2::ggplot()` object.

Aesthetics

`geom_glycan()` understands the following aesthetics. `x`, `y`, and `structure` are required; the remaining aesthetics are optional:

- `x`
- `y`
- `structure`, containing glycan structure strings or `glyrepr::glycan_structure()` values
- `size`, an optional whole-cartoon scale multiplier that defaults to 1. Mapped values are transformed by `ggplot2`'s size scale; use `ggplot2::scale_size_identity()` when the mapped values are literal multipliers. This is distinct from `node_size`, which changes residue size within the cartoon.
- `hjust`, an optional horizontal justification that defaults to 0.5. Numeric 0 aligns the cartoon content's left bound with `x`, and 1 aligns its right bound. Use `hjust_red_end()` to anchor a vertical cartoon at its reducing end.
- `vjust`, an optional vertical justification that defaults to 0.5. Numeric 0 aligns the cartoon content's bottom bound with `y`, including the end of a reducing-end annotation line, and 1 aligns its top bound. Use `vjust_red_end()` to anchor a horizontal cartoon at its reducing end.
- `angle`, an optional rotation in degrees that defaults to 0. Rotation is applied after the cartoon is drawn, independently of `orient`.

Examples

```
glycans <- data.frame(
  x = c(1, 3),
  y = c(1, 2),
  size = c(0.7, 1.1),
  structure = c(
    "Gal(b1-3)GalNAc(a1-",
    "Man(a1-3)[Man(a1-6)]Man(b1-4)GlcNAc(b1-"
  )
)

ggplot2::ggplot(
  glycans,
  ggplot2::aes(
    x = .data$x,
    y = .data$y,
    structure = .data$structure,
    size = .data$size
  )
) +
  geom_glycan() +
  ggplot2::scale_size_identity() +
  ggplot2::coord_cartesian(
    xlim = c(0, 4),
    ylim = c(0, 3),
    expand = FALSE
  )
```

```
# Bottom-align a row of vertical glycans with different heights.
ggplot2::ggplot(
  glycans,
  ggplot2::aes(x = .data$x, y = 1, structure = .data$structure)
) +
  geom_glycan(orient = "V", vjust = 0)
```

geom_node_glycan	<i>Draw glycans as ggraph nodes</i>
------------------	-------------------------------------

Description

geom_node_glycan() is a [ggraph](#) node layer backed by [geom_glycan\(\)](#). It supplies the node layout's x and y columns as the default position aesthetics and supports ggraph's filter aesthetic.

Usage

```
geom_node_glycan(
  mapping = NULL,
  data = NULL,
  position = "identity",
  show.legend = NA,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by ggplot2::aes() . The structure aesthetic is required. The x and y aesthetics default to the node coordinates calculated by ggraph.
data	The node data to display. When NULL, the default, the layer uses the node data calculated by ggraph::ggraph() .
position	A position adjustment. Defaults to "identity".
show.legend	Logical. Should this layer be included in legends?
...	Arguments passed to geom_glycan() , including fixed aesthetics and glycan rendering options.

Value

A ggplot2 layer that can be added to a [ggraph::ggraph\(\)](#) plot.

Aesthetics

geom_node_glycan() understands the same aesthetics as [geom_glycan\(\)](#). structure is required, while x and y default to the node layout coordinates. It additionally supports filter, a logical aesthetic that selects which nodes are drawn.

Examples

```
if (requireNamespace("ggraph", quietly = TRUE)) {
  graph <- igraph::make_ring(2)
  igraph::V(graph)$glycan <- c(
    "GalNAc(a1-",
    "Gal(b1-3)GalNAc(a1-"
  )

  ggraph::ggraph(graph, layout = "linear") +
    ggraph::geom_edge_link() +
    geom_node_glycan(ggplot2::aes(structure = .data$glycan))
}
```

glycanGrob

Construct a glycan grob

Description

glycanGrob() prepares the complete drawing specification for one glycan as a grid grob. It is the low-level drawing primitive used by [draw_cartoon\(\)](#).

Usage

```
glycanGrob(
  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,
  style = 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.
style	A <code>glydraw_style</code> object that supplies rendering options. Explicitly supplied rendering arguments override it.

Value

A `glycanGrob` object inheriting from `grid::gTree()`.

Examples

```
grob <- glycanGrob("Gal(b1-3)GalNAc(a1-")
grid::grid.draw(grob)
```

glydraw_style

Create a glycan drawing style

Description

`glydraw_style()` collects the rendering options shared by `glydraw`'s standalone drawings, grobs, `ggplot2` layers, guides, and `glycan` scales. Supply the result with `style =` to reuse a visual specification. Explicit rendering arguments supplied to a drawing function override the style.

Usage

```
glydraw_style(
  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

<code>show_linkage</code>	Whether to show glycosidic linkage annotations.
<code>orient</code>	Glycan drawing orientation: "H" for horizontal or "V" for vertical.
<code>fuc_orient</code>	Fuc-like triangle orientation: "flex" or "up".
<code>red_end</code>	Reducing-end annotation. Use "~" for a wave.
<code>edge_linewidth</code>	Linewidth of glycosidic linkages.
<code>node_linewidth</code>	Linewidth of node borders.
<code>node_size</code>	Multiplier for the default node size.
<code>colors</code>	Optional named character vector of monosaccharide fill-color overrides.

Value

A `glydraw_style` object.

Examples

```
vertical_style <- glydraw_style(orient = "V", show_linkage = FALSE)
draw_cartoon("Gal(b1-3)GalNAc(a1-)", style = vertical_style)
```

guide_glycan

Use glycan cartoons as legend labels

Description

`guide_glycan()` is a legend guide that replaces the usual text labels with glycan cartoons while retaining the legend keys drawn by the plot layers. Mapped discrete values and values returned by the scale's `labels` argument may be glycan structure strings supported by `glyparse::auto_parse()` or `glyrepr::glycan_structure()` vectors.

Usage

```
guide_glycan(
  title = ggplot2::waiver(),
  theme = NULL,
  position = NULL,
  direction = NULL,
  override.aes = list(),
  nrow = NULL,
  ncol = NULL,
  reverse = FALSE,
  order = 0,
  size = 0.4,
  orient = c("H", "V"),
  hjust = 0,
  vjust = vjust_red_end(),
  show_linkage = TRUE,
  red_end = "",
  fuc_orient = c("flex", "up"),
  edge_linewidth = 0.8,
  node_linewidth = 0.8,
  node_size = 1,
  colors = NULL,
  style = NULL
)
```

Arguments

title	A character string or expression indicating a title of guide. If NULL, the title is not shown. By default (waiver()), the name of the scale object or the name specified in labs() is used for the title.
theme	A theme object to style the guide individually or differently from the plot's theme settings. The theme argument in the guide partially overrides, and is combined with, the plot's theme. Arguments that apply to a single legend are respected, most of which have the legend-prefix. Arguments that apply to combined legends (the legend box) are ignored, including <code>legend.position</code> , <code>legend.justification.*</code> , <code>legend.location</code> and <code>legend.box.*</code> .
position	A character string indicating where the legend should be placed relative to the plot panels. One of "top", "right", "bottom", "left", or "inside".
direction	A character string indicating the direction of the guide. One of "horizontal" or "vertical".
override.aes	A list specifying aesthetic parameters of legend key. See details and examples.
nrow, ncol	The desired number of rows and column of legends respectively.
reverse	logical. If TRUE the order of legends is reversed.
order	positive integer less than 99 that specifies the order of this guide among multiple guides. This controls the order in which multiple guides are displayed, not the contents of the guide itself. If 0 (default), the order is determined by a secret algorithm.

size	Positive scalar that uniformly scales each legend-label cartoon. Defaults to 0.4.
orient	Glycan drawing orientation, either "H" for horizontal or "V" for vertical. Defaults to "H".
hjust	Horizontal cartoon justification between 0 and 1, or <code>hjust_red_end()</code> . It defaults to 0 for horizontal cartoons and <code>hjust_red_end()</code> for vertical cartoons.
vjust	Vertical cartoon justification between 0 and 1, or <code>vjust_red_end()</code> . It defaults to <code>vjust_red_end()</code> for horizontal cartoons and 0.5 for vertical cartoons.
show_linkage	Whether to show glycosidic linkage annotations inside the cartoons. Defaults to TRUE.
red_end	Reducing-end annotation passed to <code>glycanGrob()</code> . Use "~" for a wave, or another string to display that text. Defaults to "".
fuc_orient	Fuc-like triangle orientation passed to <code>glycanGrob()</code> .
edge_linewidth	Linkage linewidth passed to <code>glycanGrob()</code> .
node_linewidth	Node-border linewidth passed to <code>glycanGrob()</code> .
node_size	Node-size multiplier passed to <code>glycanGrob()</code> .
colors	Optional named character vector of monosaccharide fill colors passed to <code>glycanGrob()</code> .
style	A <code>glydraw_style</code> object that supplies rendering options. Explicitly supplied rendering arguments override it.

Value

A ggplot2 legend guide that draws glycan cartoons in place of text labels.

Examples

```
glycans <- data.frame(
  structure = c(
    "Gal(b1-3)GalNAc(a1-",
    "Man(a1-3)[Man(a1-6)]Man(b1-4)GlcNAc(b1-"
  ),
  abundance = c(12, 8)
)

ggplot2::ggplot(
  glycans,
  ggplot2::aes(
    x = .data$structure,
    y = .data$abundance,
    fill = .data$structure
  )
) +
  ggplot2::geom_col() +
  ggplot2::scale_fill_discrete(guide = guide_glycan())
```

hjust_red_end*Anchor glycan cartoons at their reducing ends*

Description

These helpers align the reducing end of every glycan cartoon with its anchor in `geom_glycan()`, `scale_x_glycan()`, `scale_y_glycan()`, or `guide_glycan()`. Use `hjust_red_end()` for horizontal alignment when `orient = "V"`, and use `vjust_red_end()` for vertical alignment when `orient = "H"`. Because the required justification is calculated separately from each cartoon's rendered bounds, the helpers also work for collections of glycans with different asymmetric branches. Glycan scales and guides use reducing-end justification by default along the axis perpendicular to the drawing orientation; `geom_glycan()` remains centered by default.

Usage

```
hjust_red_end()
```

```
vjust_red_end()
```

Value

A reducing-end justification marker accepted by the glycan layer, scales, and guide.

Examples

```
glycan <- data.frame(  
  x = 0,  
  y = 0,  
  structure = "Man(a1-3)[Man(a1-6)]Man(b1-4)GlcNAc(b1-"  
)  
  
ggplot2::ggplot(  
  glycan,  
  ggplot2::aes(x = .data$x, y = .data$y, structure = .data$structure)  
) +  
  geom_glycan(orient = "V", hjust = hjust_red_end())  
  
ggplot2::ggplot(  
  glycan,  
  ggplot2::aes(x = .data$x, y = .data$y, structure = .data$structure)  
) +  
  geom_glycan(orient = "H", vjust = vjust_red_end())
```

```
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

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

Value

The original glycan cartoon, invisibly.

```
save_cartoon Save fixed-size glycan cartoon image to local device.
```

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"))
```

scale_x_glycan	<i>Use glycan cartoons as axis labels</i>
----------------	---

Description

`scale_x_glycan()` and `scale_y_glycan()` are discrete position scales that replace text tick labels with compact glycan cartoons. Mapped discrete values and values returned by the scale's `labels` argument may be glycan structure strings supported by `glyparse::auto_parse()` or `glyrepr::glycan_structure()` vectors. X-axis cartoons are vertical and bottom-aligned by default, while y-axis cartoons are horizontal and right-aligned by default. The cartoon orientation and alignment adapt to the displayed axis, including when the axes are swapped by `ggplot2::coord_flip()`.

Usage

```
scale_x_glycan(
  name = ggplot2::waiver(),
  ...,
  palette = seq_len,
  expand = ggplot2::waiver(),
  position = "bottom",
  sec.axis = ggplot2::waiver(),
  continuous.limits = NULL,
  size = 0.4,
  angle = 0,
  hjust = hjust_red_end(),
  vjust = 0,
```

```

    nudge_x = 0,
    nudge_y = 0,
    show_linkage = TRUE,
    red_end = "",
    fuc_orient = c("flex", "up"),
    edge_linewidth = 0.8,
    node_linewidth = 0.8,
    node_size = 1,
    colors = NULL,
    style = NULL
  )

scale_y_glycan(
  name = ggplot2::waiver(),
  ...,
  palette = seq_len,
  expand = ggplot2::waiver(),
  position = "left",
  sec.axis = ggplot2::waiver(),
  continuous.limits = NULL,
  size = 0.4,
  angle = 0,
  hjust = 1,
  vjust = vjust_red_end(),
  nudge_x = 0,
  nudge_y = 0,
  show_linkage = TRUE,
  red_end = "",
  fuc_orient = c("flex", "up"),
  edge_linewidth = 0.8,
  node_linewidth = 0.8,
  node_size = 1,
  colors = NULL,
  style = NULL
)

```

Arguments

<code>name</code>	The name of the scale, displayed as the axis title. Use <code>NULL</code> to remove the title.
<code>...</code>	Arguments passed to <code>ggplot2::scale_x_discrete()</code> or <code>ggplot2::scale_y_discrete()</code> , including breaks and labels.
<code>palette</code>	A palette function that returns discrete position values.
<code>expand</code>	Expansion applied to the discrete position scale.
<code>position</code>	Position of the axis.
<code>sec.axis</code>	A secondary axis specification.
<code>continuous.limits</code>	Continuous limits used to position the discrete scale.

size	Positive scalar that uniformly scales each axis-label cartoon. Defaults to 0.4.
angle	Rotation in degrees applied to each axis-label cartoon, independently of the cartoon orientation. Defaults to 0.
hjust	Horizontal justification. Vertical x-axis cartoons default to <code>hjust_red_end()</code> , while horizontal y-axis cartoons default to 1.
vjust	Vertical justification. Vertical x-axis cartoons default to 0, while horizontal y-axis cartoons default to <code>vjust_red_end()</code> .
nudge_x	Horizontal adjustment of each cartoon, in millimetres. Positive values move cartoons to the right. When this moves cartoons toward or away from a y-axis title, the title moves with them to preserve the gap. Defaults to 0.
nudge_y	Vertical adjustment of each cartoon, in millimetres. Positive values move cartoons upward. When this moves cartoons toward or away from an x-axis title, the title moves with them to preserve the gap. Defaults to 0.
show_linkage	Whether to show glycosidic linkage annotations inside the cartoons. Defaults to TRUE.
red_end	Reducing-end annotation passed to <code>glycanGrob()</code> . Use "~" for a wave, or another string to display that text. Defaults to "".
fuc_orient	Fuc-like triangle orientation passed to <code>glycanGrob()</code> .
edge_linewidth	Linkage linewidth passed to <code>glycanGrob()</code> .
node_linewidth	Node-border linewidth passed to <code>glycanGrob()</code> .
node_size	Node-size multiplier passed to <code>glycanGrob()</code> .
colors	Optional named character vector of monosaccharide fill colors passed to <code>glycanGrob()</code> .
style	A <code>glydraw_style</code> object that supplies rendering options. Its orientation is ignored because each scale determines the axis direction. Explicitly supplied rendering arguments override it.

Value

A `ggplot2` discrete position scale.

Examples

```
glycans <- data.frame(
  structure = c(
    "Gal(b1-3)GalNAc(a1-",
    "Man(a1-3)[Man(a1-6)]Man(b1-4)GlcNAc(b1-"
  ),
  abundance = c(12, 8)
)

ggplot2::ggplot(glycans, ggplot2::aes(x = structure, y = abundance)) +
  ggplot2::geom_col() +
  scale_x_glycan()
```

Index

`draw_cartoon`, [2](#)
`draw_cartoon()`, [6](#), [10](#), [16](#)

`export_cartoons`, [3](#)

`geom_glycan`, [6](#)
`geom_glycan()`, [9](#), [15](#)
`geom_node_glycan`, [9](#)
`ggplot2::aes()`, [6](#), [9](#)
`ggplot2::coord_flip()`, [17](#)
`ggplot2::geom_text()`, [7](#)
`ggplot2::ggplot()`, [7](#)
`ggplot2::ggsave()`, [4](#), [5](#), [17](#)
`ggplot2::layer()`, [7](#)
`ggplot2::scale_size_identity()`, [8](#)
`ggplot2::scale_x_discrete()`, [18](#)
`ggplot2::scale_y_discrete()`, [18](#)
`ggraph`, [9](#)
`ggraph::ggraph()`, [9](#)
`glycanGrob`, [10](#)
`glycanGrob()`, [14](#), [19](#)
`glydraw_style`, [11](#)
`glyparse::auto_parse()`, [2](#), [4](#), [10](#), [12](#), [17](#)
`glyrepr::glycan_structure()`, [2–5](#), [7](#), [8](#),
[10–12](#), [17](#)
`grid::gTree()`, [11](#)
`guide_glycan`, [12](#)
`guide_glycan()`, [15](#)

`hjust_red_end`, [15](#)
`hjust_red_end()`, [8](#), [14](#), [19](#)

`labs()`, [13](#)

`print.glydraw_cartoon`, [16](#)

`save_cartoon`, [16](#)
`save_cartoon()`, [4](#)
`scale_x_glycan`, [17](#)
`scale_x_glycan()`, [15](#)
`scale_y_glycan (scale_x_glycan)`, [17](#)

`scale_y_glycan()`, [15](#)

`theme`, [13](#)

`vjust_red_end (hjust_red_end)`, [15](#)
`vjust_red_end()`, [8](#), [14](#), [19](#)

`waiver()`, [13](#)