

Package ‘glydraw’

August 26, 2026

Title Draw Beautiful Symbol Nomenclature for Glycans

Version 0.9.0

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 ComplexHeatmap, ggraph, ggsketch (>= 2.0.0), knitr, ragg, rmarkdown, systemfonts, 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 (>= 1.0.0), rlang, dplyr, ggplot2 (>= 4.0.0), igraph (>= 2.2.0), purrr, png, glyparse (>= 0.7.1), 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 2026-08-26 09:30:02 UTC

Contents

anno_glycan	2
draw_cartoon	4
draw_cartoon_sketch	5
export_cartoons	7
geom_glycan	9
geom_node_glycan	12
glycanGrob	13
glydraw_colors	14
guide_glycan	14
hjust_red_end	16
print.glydraw_cartoon	17
save_cartoon	18
scale_x_glycan	19
style_glydraw	21

Index	24
--------------	-----------

anno_glycan	<i>Use glycan cartoons as ComplexHeatmap labels</i>
-------------	---

Description

anno_glycan() creates a [ComplexHeatmap::AnnotationFunction\(\)](#) that draws glycan cartoons in the order used by a ComplexHeatmap heatmap. The cartoons follow row or column clustering, reordering, and splitting.

Usage

```
anno_glycan(  
  structure,  
  which = c("column", "row"),  
  side = NULL,  
  size = 0.4,  
  angle = 0,  
  hjust = NULL,  
  vjust = NULL,  
  nudge_x = 0,  
  nudge_y = 0,  
  show_linkage = TRUE,  
  style = style_glydraw(),  
  width = NULL,  
  height = NULL,  
  show_name = FALSE,  
  red_end = NULL,  
  orient = NULL  
)
```

Arguments

structure	A character vector of glycan structure strings supported by <code>glyparse::auto_parse()</code> or a <code>glyrepr::glycan_structure()</code> vector. Its order must match the rows or columns of the heatmap matrix.
which	Whether the cartoons label heatmap "column" or "row" observations.
side	Side on which the annotation is placed. Column annotations accept "bottom" or "top"; row annotations accept "left" or "right". Defaults to the corresponding glycan scale position.
size	Positive scalar that uniformly scales each cartoon. Defaults to 0.4.
angle	Rotation in degrees applied to each cartoon independently of its drawing orientation. Defaults to 0.
hjust	Horizontal justification. NULL uses <code>hjust_red_end()</code> for a top or bottom annotation with a vertical orientation, 1 or 0 for a left or right annotation with a horizontal orientation, respectively, and 0.5 for other side-orientation combinations.
vjust	Vertical justification. NULL uses 0 for a top or bottom annotation with a vertical orientation, <code>vjust_red_end()</code> for a left or right annotation with a horizontal orientation, and 0.5 for other side-orientation combinations.
nudge_x	Horizontal adjustment of each cartoon, in millimetres. Positive values move cartoons to the right. Defaults to 0.
nudge_y	Vertical adjustment of each cartoon, in millimetres. Positive values move cartoons upward. Defaults to 0.
show_linkage	Whether to show glycosidic linkage annotations inside the cartoons. Defaults to TRUE.
style	A <code>style_glydraw()</code> object that controls the cartoons' visual appearance.
width	Optional <code>grid::unit()</code> width for a row annotation. NULL calculates the width from the rendered cartoons.
height	Optional <code>grid::unit()</code> height for a column annotation. NULL calculates the height from the rendered cartoons.
show_name	Whether ComplexHeatmap should show the annotation name. Defaults to FALSE because the cartoons normally replace row or column names.
red_end	Reducing-end annotation. NULL, the default, uses <code>red_end</code> from <code>style</code> . A non-NULL value overrides <code>style\$red_end</code> .
orient	Glycan drawing orientation. NULL, the default, selects the orientation from <code>side</code> : "up" for "bottom" or "top", "left" for "left", and "right" for "right".

Details

Column annotations use the visual defaults of `scale_x_glycan()`: vertical cartoons anchored at their reducing ends and aligned along their bottom bounds. Row annotations use the defaults of `scale_y_glycan()`: horizontal cartoons aligned along their right bounds and anchored at their reducing ends. `side` should match the heatmap annotation side and defaults to "bottom" for columns and "left" for rows.

The required annotation width or height is calculated from the largest rendered cartoon, including rotation and perpendicular nudging. Supply width for row annotations or height for column annotations to override the calculated size.

Value

A ComplexHeatmap AnnotationFunction object.

Examples

```
if (requireNamespace("ComplexHeatmap", quietly = TRUE)) {
  mat <- matrix(
    seq_len(9),
    nrow = 3,
    dimnames = list(paste0("row", 1:3), paste0("column", 1:3))
  )
  structures <- c(
    "GlcNAc(??-",
    "Gal(??-?)GlcNAc(??-",
    "Neu5Ac(??-?)Gal(??-?)GlcNAc(??-"
  )
  ComplexHeatmap::Heatmap(
    mat,
    show_row_names = FALSE,
    show_column_names = FALSE,
    left_annotation = ComplexHeatmap::rowAnnotation(
      glycan = anno_glycan(structures, which = "row")
    ),
    bottom_annotation = ComplexHeatmap::HeatmapAnnotation(
      glycan = anno_glycan(structures, which = "column")
    )
  )
}
```

draw_cartoon

Draw a Symbol Nomenclature For Glycan (SNFG)

Description

Draw a Symbol Nomenclature For Glycan (SNFG)

Usage

```
draw_cartoon(
  structure,
  ...,
  show_linkage = TRUE,
  orient = c("left", "right", "up", "down"),
```

```

    highlight = NULL,
    style = style_glydraw(),
    red_end = 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	Direction in which the glycan extends from its reducing end: one of "left", "right", "up", or "down". Defaults to "left".
highlight	An integer vector specifying the node indices to highlight. This argument is applicable only when structure is a glyrepr::glycan_structure() . Note that for a glyrepr::glycan_structure() , 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 style_glydraw() object that controls the cartoon's visual appearance.
red_end	Reducing-end annotation. NULL, the default, uses <code>red_end</code> from style. A non-NULL value overrides <code>style\$red_end</code> . Ignored when <code>style\$red_end_length</code> is 0 or the glycan is an alditol. To annotate an amino-acid sequence, tag its single glycosite as, for example, "ABC<site>D</site>EFG".

Value

a ggplot2 object

Examples

```

draw_cartoon("Gal(b1-3)GalNAc(a1-")
draw_cartoon(
  "Gal(b1-3)GalNAc(a1-",
  style = style_glydraw(font_family = "serif")
)

```

draw_cartoon_sketch	<i>Draw a sketch-style glycan cartoon</i>
---------------------	---

Description

`draw_cartoon_sketch()` uses [ggsketch::ggsketch-package](#) geoms to give a glycan cartoon hand-drawn strokes and patterned residue fills. Its glycan layout, annotations, orientation, and sizing are otherwise identical to [draw_cartoon\(\)](#).

Usage

```
draw_cartoon_sketch(
    structure,
    ...,
    show_linkage = TRUE,
    orient = c("left", "right", "up", "down"),
    highlight = NULL,
    style = style_glydraw(),
    roughness = 1,
    bowing = 1,
    n_passes = 2L,
    seed = NULL,
    fill_style = "pencil_shade",
    hatchure_angle = 45,
    hatchure_gap = 0.03,
    fill_weight = 0.5,
    medium = NULL,
    red_end = 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. Defaults to TRUE. Substituent annotations are always shown.
orient	Direction in which the glycan extends from its reducing end: one of "left", "right", "up", or "down". Defaults to "left".
highlight	An integer vector specifying the node indices to highlight. This argument is applicable only when structure is a glyrepr::glycan_structure() . Note that for a glyrepr::glycan_structure() , the node indices correspond exactly to the monosaccharides in its printed IUPAC nomenclature. For example, given glyrepr::as_glycan_structure("Gal(b1-3)[GlcNAc(b1-6)]GalNAc(a1-") , setting <code>highlight = c(1, 3)</code> will highlight the "Gal" and "GalNAc" nodes.
style	A style_glydraw() object that controls the cartoon's visual appearance.
roughness	Non-negative roughness of the hand-drawn strokes. Zero produces straight strokes. Hex circle outlines are automatically softened to keep their curved borders smooth.
bowing	Non-negative multiplier controlling how much strokes bow.
n_passes	Number of times each sketch stroke is drawn.
seed	An optional integer seed for reproducible sketch strokes. When NULL, <code>ggsketch</code> uses <code>getOption("ggsketch.seed", 1L)</code> .
fill_style	Residue fill pattern. Defaults to "pencil_shade". See ggsketch::geom_sketch_polygon() for the available styles.
hatchure_angle	Angle of patterned fill lines in degrees.

hachure_gap	Gap between patterned fill lines as a proportion of the node diameter. Defaults to 0.03.
fill_weight	Stroke weight of patterned fill lines.
medium	Optional drawing medium for linkage and reducing-end strokes. See ggsketch::sketch_media() for the available media.
red_end	Reducing-end annotation. NULL, the default, uses red_end from style. A non-NULL value overrides style\$red_end. Ignored when style\$red_end_length is 0 or the glycan is an alditol. To annotate an amino-acid sequence, tag its single glycosite as, for example, "ABC<site>D</site>EFG".

Details

Sketch cartoons always use a handwriting font, ignoring font_family in style. They prefer an installed sketch-style font that contains Greek alpha, Greek beta, and decimal digits so all linkage labels use one font.

Value

A glydraw_cartoon ggplot2 object.

Examples

```
if (requireNamespace("ggsketch", quietly = TRUE)) {
  draw_cartoon_sketch("Gal(b1-3)GalNAc(a1-", seed = 1)
}
```

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("left", "right", "up", "down"),
  style = style_glydraw(),
  red_end = NULL
)
```

Arguments

<code>x</code>	A <code>glyrepr::glycan_structure()</code> vector, or a character vector of any glycan structure text nomenclatures supported by <code>glyparse::auto_parse()</code> .
<code>dirname</code>	Directory name to save the cartoons. If it does not exist, it is created.
<code>...</code>	Ignored.
<code>file_ext</code>	File extension supported by <code>ggplot2::ggsave()</code> . Defaults to "png".
<code>dpi</code>	Deprecated and ignored. Use <code>scale</code> to change the output size.
<code>scale</code>	Numeric output-size multiplier passed to <code>save_cartoon()</code> .
<code>show_linkage</code>	Show glycosidic linkage annotations or not. Default is TRUE. Substituent annotations are always shown.
<code>orient</code>	Direction in which the glycan extends from its reducing end: one of "left", "right", "up", or "down". Defaults to "left".
<code>style</code>	A <code>style_glydraw()</code> object that controls the cartoon's visual appearance.
<code>red_end</code>	Reducing-end annotation. NULL, the default, uses <code>red_end</code> from <code>style</code> . A non-NULL value overrides <code>style\$red_end</code> . Ignored when <code>style\$red_end_length</code> is 0 or the glycan is an alditol. To annotate an amino-acid sequence, tag its single glycosite as, for example, "ABC<site>D</site>EFG".

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("left", "right", "up", "down"),
  highlight = NULL,
  style = style_glydraw(),
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  red_end = NULL
)

```

Arguments

mapping	Set of aesthetic mappings created by ggplot2::aes() . 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	Direction in which the glycan extends from its reducing end: one of "left", "right", "up", or "down". Defaults to "left".
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>style_glydraw()</code> object that controls the cartoon's visual appearance.
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.
red_end	Reducing-end annotation. NULL, the default, uses red_end from style. A non-NULL value overrides style\$red_end. Ignored when style\$red_end_length is 0 or the glycan is an alditol. To annotate an amino-acid sequence, tag its single glycosite as, for example, "ABC<site>D</site>EFG".

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`.
- `alpha`, an optional whole-cartoon transparency. The default, `NA`, is opaque. Each cartoon is composited before `alpha` is applied so nodes continue to occlude the linkage edges beneath them. Non-opaque values require a graphics device that supports alpha masks and transformations; unsupported devices throw an error rather than render an inaccurate cartoon.

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 = "up", 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, drawing controls, and style.

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("left", "right", "up", "down"),
  highlight = NULL,
  style = style_glydraw(),
  red_end = 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	Direction in which the glycan extends from its reducing end: one of "left", "right", "up", or "down". Defaults to "left".
highlight	An integer vector specifying the node indices to highlight. This argument is applicable only when structure is a glyrepr::glycan_structure() . Note that for a glyrepr::glycan_structure() , the node indices correspond exactly to the monosaccharides in its printed IUPAC nomenclature. For example, given glyrepr::as_glycan_structure("Gal(b1-3)[GlcNAc(b1-6)]GalNAc(a1-") , setting <code>highlight = c(1, 3)</code> will highlight the "Gal" and "GalNAc" nodes.
style	A style_glydraw() object that controls the cartoon's visual appearance.
red_end	Reducing-end annotation. NULL, the default, uses red_end from style. A non-NULL value overrides style\$red_end. Ignored when style\$red_end_length is 0 or the glycan is an alditol. To annotate an amino-acid sequence, tag its single glycosite as, for example, "ABC<site>D</site>EFG".

Value

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

Examples

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

glydraw_colors	<i>Default color palette in glydraw</i>
----------------	---

Description

This palette uses the SNFG standard defined in <https://www.ncbi.nlm.nih.gov/glycans/snfg.html>.

Usage

```
glydraw_colors()
```

Value

A named character vector of colors.

Examples

```
glydraw_colors()
```

guide_glycan	<i>Use glycan cartoons as legend labels</i>
--------------	---

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("left", "right", "up", "down"),
  hjust = 0,
  vjust = vjust_red_end(),
  show_linkage = TRUE,
  style = style_glydraw(),
  red_end = 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	Direction in which the glycan extends from its reducing end: one of "left", "right", "up", or "down". Defaults to "left".

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.
style	A <code>style_glydraw()</code> object that controls the cartoons' visual appearance.
red_end	Reducing-end annotation. NULL, the default, uses <code>red_end</code> from <code>style</code> . A non-NULL value overrides <code>style\$red_end</code> .

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` is "up" or "down", and use `vjust_red_end()` for vertical alignment when `orient` is "left" or "right". 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 = "up", hjust = hjust_red_end())

ggplot2::ggplot(
  glycan,
  ggplot2::aes(x = .data$x, y = .data$y, structure = .data$structure)
) +
  geom_glycan(orient = "left", 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	<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"))
```

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,  
  style = style_glydraw(),  
  red_end = NULL,  
  orient = 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,
    style = style_glydraw(),
    red_end = NULL,
    orient = NULL
  )

```

Arguments

name	The name of the scale, displayed as the axis title. Use NULL to remove the title.
...	Arguments passed to <code>ggplot2::scale_x_discrete()</code> or <code>ggplot2::scale_y_discrete()</code> , including breaks and labels.
palette	A palette function that returns discrete position values.
expand	Expansion applied to the discrete position scale.
position	Position of the axis.
sec.axis	A secondary axis specification.
continuous.limits	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. When omitted, cartoons on a top or bottom axis with a vertical orientation use <code>hjust_red_end()</code> . Cartoons on a left or right axis with a horizontal orientation use 1 or 0, respectively. Other position-orientation combinations use 0.5.
vjust	Vertical justification. When omitted, cartoons on a top or bottom axis with a vertical orientation use 0. Cartoons on a left or right axis with a horizontal orientation use <code>vjust_red_end()</code> . Other position-orientation combinations use 0.5.
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.
style	A <code>style_glydraw()</code> object that controls the cartoons' visual appearance.
red_end	Reducing-end annotation. NULL, the default, uses <code>red_end</code> from <code>style</code> . A non-NULL value overrides <code>style\$red_end</code> .
orient	Glycan drawing orientation. NULL, the default, selects the orientation from the displayed axis position: "up" for "bottom" or "top", "left" for "left", and "right" for "right".

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()
```

style_glydraw	<i>Glycan drawing styles</i>
---------------	------------------------------

Description

Style constructors collect the rendering options shared by glydraw's standalone drawings, grobs, ggplot2 layers, guides, and glycan scales. `style_glydraw()` provides glydraw's default appearance, while the other constructors provide presets matching common glycan-drawing conventions. Supply a returned style with `style =` to reuse its visual specification.

Usage

```
style_glydraw(
  fuc_orient = "flex",
  red_end = "",
  red_end_length = 0.6,
  red_end_size = 6,
  edge_linewidth = 0.8,
  node_linewidth = 0.8,
  node_size = 1,
  font_family = "",
  colors = glydraw_colors()
)

style_glygen(
  fuc_orient = "flex",
  red_end = "~",
  red_end_length = 1,
  red_end_size = 6,
  edge_linewidth = 0.8,
  node_linewidth = 0.8,
```

```

    node_size = 1,
    font_family = "arial",
    colors = glydraw_colors()
)

style_snfg(
    fuc_orient = "up",
    red_end = "",
    red_end_length = 1,
    red_end_size = 6,
    edge_linewidth = 1.5,
    node_linewidth = 0.8,
    node_size = 1.15,
    font_family = "arial",
    colors = glydraw_colors()
)

style_glycoworkbench(
    fuc_orient = "flex",
    red_end = "~",
    red_end_length = 1,
    red_end_size = 6,
    edge_linewidth = 0.8,
    node_linewidth = 0.8,
    node_size = 1,
    font_family = "arial",
    colors = c(glyWhite = "#FFFFFF", glyBlue = "#0000FF", glyGreen = "#5AC54B", glyYellow =
        "#FFFF54", glyOrange = "#F7EAD7", glyPink = "#FFC0CB", glyPurple = "#800080",
        glyLightBlue = "#ADD8E6", glyBrown = "#8B4513", glyRed = "#FF0000")
)

```

Arguments

<code>fuc_orient</code>	Fuc-like triangle orientation: "flex" or "up".
<code>red_end</code>	Reducing-end annotation. Use "~" for a wave, any other string for custom text, or tag one amino-acid site as "ABC<site>D</site>EFG". Ignored when <code>red_end_length</code> is 0 or the glycan is an alditol.
<code>red_end_length</code>	Length of the reducing-end line in plot coordinate units. Set to 0 to omit the line and any <code>red_end</code> wave or custom text while retaining the axis-aligned core anomer annotation. Ignored for alditols.
<code>red_end_size</code>	Size of custom text passed to <code>red_end</code> . The "~" wave is not affected. Ignored for alditols.
<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>font_family</code>	A length-one character string naming the font family used for linkage, substituent, and reducing-end text annotations. Portable choices are "sans", "serif",

	and "mono". Other family names, such as installed system fonts, are graphics-device dependent. The default "" uses the graphics device's default font.
colors	A named character vector of SNFG colors in the format returned by <code>glydraw_colors()</code> . Names must be complete and match that palette.

Value

A glydraw_style object.

Functions

- `style_glydraw()`: Use glydraw's default style.
- `style_glygen()`: Use a GlyGen-style preset.
- `style_snfg()`: Use a Symbol Nomenclature for Glycans preset.
- `style_glycoworkbench()`: Use a GlycoWorkbench-style preset.

Examples

```
serif_style <- style_glydraw(font_family = "serif")
draw_cartoon("Gal(b1-3)GalNAc(a1-", style = serif_style)

draw_cartoon("Gal(b1-3)GalNAc(a1-", style = style_glygen())
draw_cartoon("Gal(b1-3)GalNAc(a1-", style = style_snfg())
draw_cartoon("Gal(b1-3)GalNAc(a1-", style = style_glycoworkbench())
```

Index

anno_glycan, [2](#)

ComplexHeatmap::AnnotationFunction(),
[2](#)

draw_cartoon, [4](#)
draw_cartoon(), [5](#), [9](#), [13](#), [17](#), [18](#)
draw_cartoon_sketch, [5](#)

export_cartoons, [7](#)

geom_glycan, [9](#)
geom_glycan(), [12](#), [16](#)
geom_node_glycan, [12](#)
ggplot2::aes(), [9](#), [12](#)
ggplot2::coord_flip(), [19](#)
ggplot2::geom_text(), [10](#)
ggplot2::ggplot(), [10](#)
ggplot2::ggsave(), [8](#), [18](#)
ggplot2::layer(), [10](#)
ggplot2::scale_size_identity(), [10](#)
ggplot2::scale_x_discrete(), [20](#)
ggplot2::scale_y_discrete(), [20](#)
ggraph, [12](#)
ggraph::ggraph(), [12](#)
ggsketch::geom_sketch_polygon(), [6](#)
ggsketch::ggsketch-package, [5](#)
ggsketch::sketch_media(), [7](#)
glycanGrob, [13](#)
glydraw_colors, [14](#)
glydraw_colors(), [23](#)
glyparse::auto_parse(), [3](#), [5](#), [6](#), [8](#), [13](#), [14](#),
[19](#)
glyrepr::glycan_structure(), [3](#), [5](#), [6](#), [8](#),
[10](#), [13](#), [14](#), [19](#)
grid::gTree(), [14](#)
grid::unit(), [3](#)
guide_glycan, [14](#)
guide_glycan(), [16](#)

hjust_red_end, [16](#)

hjust_red_end(), [3](#), [10](#), [16](#), [20](#)

labs(), [15](#)

print.glydraw_cartoon, [17](#)

save_cartoon, [18](#)
save_cartoon(), [8](#)
scale_x_glycan, [19](#)
scale_x_glycan(), [3](#), [16](#)
scale_y_glycan(scale_x_glycan), [19](#)
scale_y_glycan(), [3](#), [16](#)
style_glycoworkbench(style_glydraw), [21](#)
style_glydraw, [21](#)
style_glydraw(), [3](#), [5](#), [6](#), [8](#), [10](#), [13](#), [16](#), [20](#)
style_glygen(style_glydraw), [21](#)
style_snfg(style_glydraw), [21](#)

theme, [15](#)

vjust_red_end(hjust_red_end), [16](#)
vjust_red_end(), [3](#), [11](#), [16](#), [20](#)

waiver(), [15](#)