Package: glyvis (via r-universe)

May 31, 2026

Title Fast Visualization For Glycoverse

Version 0.6.0

Description Provides a unified interface for visualizing glycomics data within the 'glycoverse' ecosystem. Implements 'autoplot()' methods for various data structures including statistical results from 'glystats', experimental data from 'glyexp', and glycan structures from 'glyrepr'. Designed for rapid data exploration and quality assessment rather than publication-quality figure generation.

License MIT + file LICENSE

Suggests testthat (>= 3.0.0), vdiffr, glystats (>= 0.10.0), glyclean (>= 0.7.1), ggdendro, factoextra, GGally, enrichplot, EnhancedVolcano, ggseqlogo, UniProt.ws, pheatmap, knitr, rmarkdown, patchwork

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL https://glycoverse.github.io/glyvis/

Imports checkmate, cli, dplyr, ggplot2, ggplotify, ggrepel, glyexp (>= 0.10.0), magrittr, purrr, rlang, stringr, tibble

Depends R (>= 4.0)

VignetteBuilder knitr

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev

Repository https://glycoverse.r-universe.dev

Date/Publication 2026-05-01 03:58:36 UTC

RemoteUrl https://github.com/glycoverse/glyvis

RemoteRef v0.6.0

RemoteSha f6dadab709be927c3f33d22c5ae8c7cba7b0ae15

Contents

`autoplot.glyexp_experiment`

Plots for Experiments

Description

Visualization for `glyexp::experiment()` (`glyexp_experiment` object). Possible types of plots:

- "heatmap": (Default) Expression heatmap with columns as samples and rows as variables.
- "boxplot": Boxplot of expression values for each sample.

Usage

```
## S3 method for class 'glyexp_experiment'
autoplot(object, type = "heatmap", group_col = "group", ...)
```

Arguments

`object`	A `glyexp_experiment` object.
`type`	The type of plot, one of "heatmap" (default) or "boxplot".
`group_col`	A character string specifying where to find the group information. It should be a column in the sample information tibble. Defaults to "group". Only applicable to "boxplot".
`...`	Ignored.

Value

A ggplot object.

`autoplot.glystats_anova_res`

Plots for Multi-Group Differential Expression Analysis (DEA)

Description

Visualization for `glystats_anova_res` and `glystats_kruskal_res`. Draw a dotchart for p-values.

Usage

```
## S3 method for class 'glystats_anova_res'
autoplot(object, p_cutoff = 0.05, p_col = "p_adj", ...)

## S3 method for class 'glystats_kruskal_res'
autoplot(object, p_cutoff = 0.05, p_col = "p_adj", ...)
```

Arguments

`object`	A `glystats_anova_res` or `glystats_kruskal_res` object.
`p_cutoff`	The p-value cutoff. Defaults to 0.05.
`p_col`	The column name for p-value. Defaults to "p_adj". Can also be "p_val" (raw p-values without multiple testing correction).
`...`	Other arguments passed to underlying functions.

Value

A ggplot object.

`autoplot.glystats_cor_res`

Plots for Correlation Analysis

Description

Visualization for `glystats::gly_cor()` results (`glystats_cor_res` objects).

Usage

```
## S3 method for class 'glystats_cor_res'
autoplot(object, ...)
```

Arguments

`object`	A `glystats_cor_res` object.
`...`	Other arguments passed to `GGally::ggcorr()`.

Value

A ggplot object.

`autoplot.glystats_cox_res`

Plots for Cox Proportional Hazards Model

Description

Visualization for results from `glystats::gly_cox()` (`glystats_cox_res` objects). Draw a dotchart for p-values.

Usage

```
## S3 method for class 'glystats_cox_res'
autoplot(object, p_cutoff = 0.05, p_col = "p_adj", ...)
```

Arguments

`object`	A `glystats_cox_res` object.
`p_cutoff`	The p-value cutoff. Defaults to 0.05.
`p_col`	The column name for p-value. Defaults to "p_adj". Can also be "p_val" (raw p-values without multiple testing correction).
`...`	Ignored.

Value

A ggplot object.

`autoplot.glystats_go_ora_res`
Plots for Enrichment Analysis

Description

Visualization for results from `glystats::gly_enrich_go()`, `glystats::gly_enrich_kegg()`, and `glystats::gly_enrich_reactome()` (`glystats_go_ora_res`, `glystats_kegg_ora_res`, and `glystats_reactome_ora_res` objects). Possible types of plots:

- "dotplot" (default): Dotplot showing p-value, gene counts, and gene ratios.
- "barplot": Barplot showing p-values and gene counts.
- "network": Network plot showing terms and their relationships.

Usage

```
## S3 method for class 'glystats_go_ora_res'
autoplot(object, type = "dotplot", ...)

## S3 method for class 'glystats_kegg_ora_res'
autoplot(object, type = "dotplot", ...)

## S3 method for class 'glystats_reactome_ora_res'
autoplot(object, type = "dotplot", ...)
```

Arguments

`object`	A `glystats_go_ora_res`, `glystats_kegg_ora_res`, or `glystats_reactome_ora_res` object.
`type`	The type of plot, one of "dotplot" (default), "barplot", or "network".
`...`	Other arguments passed to underlying functions (`enrichplot::dotplot()` for "dotplot", `enrichplot::barplot.enrichResult()` for "barplot", and `enrichplot::emapplot()` for "network")

Value

A ggplot object.

See Also

`enrichplot::dotplot()`, `enrichplot::barplot.enrichResult()`, `enrichplot::emapplot()`

`autoplot.glystats_hclust_res` *Plots for Hierarchical Clustering*

Description

Visualization for results from `glystats::gly_hclust()` (`glystats_hclust_res` object). Draw a dendrogram.

Usage

```
## S3 method for class 'glystats_hclust_res'
autoplot(object, ...)
```

Arguments

`object`	A `glystats_hclust_res` object.
`...`	Additional arguments passed to `ggdendro::ggdendrogram()`.

Value

A ggplot object.

See Also

`ggdendro::ggdendrogram()`

`autoplot.glystats_limma_res` *Plots for Limma Result*

Description

Visualization for results from `glystats::gly_limma()` (`glystats_limma_res` objects). Only 2-group comparison is supported, and a volcano plot is drawn.

Usage

```
## S3 method for class 'glystats_limma_res'
autoplot(
  object,
  contrast = NULL,
  log2fc_cutoff = 1,
  p_cutoff = 0.05,
  p_col = "p_adj",
  ...
)
```

Arguments

`object`	A `glystats_limma_res` object.
`contrast`	A character string specifying the contrast to plot, in the format of "group1_vs_group2". Must be one of the contrasts in the result. When there is only one contrast (two-group comparison), it can be NULL (default).
`log2fc_cutoff`	The log2 fold change cutoff. Defaults to 1.
`p_cutoff`	The p-value cutoff. Defaults to 0.05.
`p_col`	The column name for p-value. Defaults to "p_adj". Can also be "p_val" (raw p-values without multiple testing correction).
`...`	Other arguments passed to underlying functions.

Value

A ggplot object.

`autoplot.glystats_oplsda_res`

Plots for OPLS-DA

Description

Visualization for results from `glystats::gly_oplsda()` (`glystats_oplsda_res` object). Possible types of plots:

- "scores" (default): Plot scores for samples.
- "loadings": Plot loadings for variables.
- "vip": Plot VIP scores for variables.
- "variance": Plot explained variance for each component.
- "s-plot": Plot the correlation between p1 and pcorr1.

Usage

```
## S3 method for class 'glystats_oplsda_res'
autoplot(
  object,
  type = "scores",
  y_type = "o1",
  groups = NULL,
  group_col = NULL,
  ...
)
```

Arguments

`object`	A `glystats_oplsda_res` object.
`type`	The type of plot, one of "loadings", "scores", "vip", "variance", or "s-plot". Defaults to "scores".
`y_type`	What to plot on the y-axis when `type` is "scores" or "loadings". Either "p2" or "o1". Defaults to "o1".
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group. Only applicable to "scores" and "loadings" types.
`group_col`	A character string specifying where to find the group information. If you uses `glystats::gly_oplsda()` on a `glyexp::experiment()` to get the result, sample information has already been added to the result. In this case, you can specify the column name in the sample information tibble to be used for coloring. If not provided, this function will try "group".
`...`	Ignored.

Value

A ggplot object.

`autoplot.glystats_pca_res`

Plots for Principal Component Analysis (PCA)

Description

Visualization for `glystats::gly_pca()` results (`glystats_pca_res` object). Possible types of plots:

- "screeplot": Scree plot to see the contribution of each PC.
- "individual" (default): Plot samples as individuals.
- "variables": Plot loadings for variables.
- "biplot": Biplot showing both samples and variables.

Usage

```
## S3 method for class 'glystats_pca_res'
autoplot(object, type = "individual", groups = NULL, group_col = NULL, ...)
```

Arguments

`object`	A `glystats_pca_res` object.
`type`	The type of plot, one of "screeplot", "individual", "variables", or "biplot".
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group. Only applicable to "individual" and "biplot" types. Passed to the `habillage` paramter of `factoextra::fviz_pca_ind()` or `factoextra::fviz_pca_biplot()`.
`group_col`	A character string specifying where to find the group information. If you uses `glystats::gly_pca()` on a `glyexp::experiment()` to get the result, sample information has already been added to the result. In this case, you can specify the column name in the sample information tibble to be used for coloring. If not provided, this function will try "group".
`...`	Other arguments passed to factoextra functions. • For "screeplot", see `factoextra::fviz_screeplot()`. • For "individual", see `factoextra::fviz_pca_ind()`. • For "variables", see `factoextra::fviz_pca_var()`. • For "biplot", see `factoextra::fviz_pca_biplot()`.

Value

A ggplot object.

See Also

`factoextra::fviz_screeplot()`, `factoextra::fviz_pca_ind()`, `factoextra::fviz_pca_var()`, `factoextra::fviz_pca_biplot()`

`autoplot.glystats_plsda_res`

Plots for PLS-DA

Description

Visualization for results from `glystats::gly_plsda()` (`glystats_plsda_res` object). Possible types of plots:

- "scores" (default): Plot scores for samples.
- "loadings": Plot loadings for variables.
- "vip": Plot VIP scores for variables.
- "variance": Plot explained variance for each component.

Usage

```
## S3 method for class 'glystats_plsda_res'
autoplot(object, type = "scores", groups = NULL, group_col = NULL, ...)
```

Arguments

`object`	A `glystats_plsda_res` object.
`type`	The type of plot, one of "loadings", "scores", "vip", or "variance". Defaults to "scores".
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group. Only applicable to "scores" and "loadings" types.
`group_col`	A character string specifying where to find the group information. If you uses `glystats::gly_plsda()` on a `glyexp::experiment()` to get the result, sample information has already been added to the result. In this case, you can specify the column name in the sample information tibble to be used for coloring. If not provided, this function will try "group".
`...`	Ignored.

Value

A ggplot object.

`autoplot.glystats_roc_res`

Plots for ROC Analysis

Description

Visualization for results from `glystats::gly_roc()` (`glystats_roc_res` objects). At most 10 variables can be plotted.

Usage

```
## S3 method for class 'glystats_roc_res'
autoplot(object, type = "dotplot", auc_cutoff = 0.5, ...)
```

Arguments

`object`	A `glystats_roc_res` object.
`type`	"dotplot" or "roc". Defaults to "dotplot". If "roc", at most 10 variables can be plotted.
`auc_cutoff`	The AUC cutoff. Defaults to 0.5. Only used if type is "dotplot".
`...`	Ignored.

Value

A ggplot object.

`autoplot.glystats_tsne_res`
Plots for t-SNE

Description

Visualization for results from `glystats::gly_tsne()` (`glystats_tsne_res` objects). Draw a scatter plot of t-SNE coordinates.

Usage

```
## S3 method for class 'glystats_tsne_res'
autoplot(object, groups = NULL, group_col = NULL, ...)
```

Arguments

`object`	A `glystats_tsne_res` object.
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group.
`group_col`	A character string specifying where to find the group information. If you uses `glystats::gly_tsne()` on a `glyexp::experiment()` to get the result, sample information has already been added to the result. In this case, you can specify the column name in the sample information tibble to be used for coloring. If not provided, this function will try "group".
`...`	Ignored.

Value

A ggplot object.

`autoplot.glystats_ttest_res`
Plots for 2-Group Differential Expression Analysis (DEA)

Description

Visualization for results from `glystats::gly_ttest()` and `glystats::gly_wilcox()` (`glystats_ttest_res` and `glystats_wilcox_res` objects). Draw a volcano plot.

Usage

```
## S3 method for class 'glystats_ttest_res'
autoplot(object, log2fc_cutoff = 1, p_cutoff = 0.05, p_col = "p_adj", ...)

## S3 method for class 'glystats_wilcox_res'
autoplot(object, log2fc_cutoff = 1, p_cutoff = 0.05, p_col = "p_adj", ...)
```

Arguments

`object`	A `glystats_ttest_res` or `glystats_wilcox_res` object.
`log2fc_cutoff`	The log2 fold change cutoff. Defaults to 1.
`p_cutoff`	The p-value cutoff. Defaults to 0.05.
`p_col`	The column name for p-value. Defaults to "p_adj". Can also be "p_val" (raw p-values without multiple testing correction).
`...`	Ignored.

Value

A ggplot object.

`autoplot.glystats_umap_res` *Plots for UMAP*

Description

Visualization for results from `glystats::gly_umap()` (`glystats_umap_res` objects). Draw a scatter plot of UMAP coordinates.

Usage

```
## S3 method for class 'glystats_umap_res'
autoplot(object, groups = NULL, group_col = NULL, ...)
```

Arguments

`object`	A `glystats_umap_res` object.
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group.
`group_col`	A character string specifying where to find the group information. If you uses `glystats::gly_umap()` on a `glyexp::experiment()` to get the result, sample information has already been added to the result. In this case, you can specify the column name in the sample information tibble to be used for coloring. If not provided, this function will try "group".
`...`	Ignored.

Value

A ggplot object.

`plot_boxplot` *Boxplot*

Description

Draw a boxplot. Currently supported data types:

- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. Boxplots of log2-transformed expression values are plotted, grouped by the group column.

Usage

```
plot_boxplot(x, ...)

## S3 method for class 'glyexp_experiment'
plot_boxplot(x, group_col = "group", ...)
```

Arguments

`x`	An object to be plotted.
`...`	Ignored.
`group_col`	A character string specifying the column name in sample information that contains group labels. Default is "group".

Value

A ggplot object.

`plot_corrplot` *Correlation Plot*

Description

Draw a correlation matrix heatmap. Currently supported data types:

- `glystats_cor_res`: Result from `glystats::gly_cor()`.
- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. Correlation analysis is first performed using `glystats::gly_cor()`, then the result is plotted.

Usage

```
plot_corrplot(x, ...)

## S3 method for class 'glystats_cor_res'
plot_corrplot(x, ...)

## S3 method for class 'glyexp_experiment'
plot_corrplot(x, stats_args = list(), ...)
```

Arguments

`x`	An object to be plotted.
`...`	Additional arguments passed to `GGally::ggcorr()`.
`stats_args`	A list of keyword arguments to pass to `glystats::gly_cor()`.

Value

A ggplot object.

`plot_enrich` *Plot Enrichment Analysis Results*

Description

Draw a enrichment analysis result plot. Currently supported data types:

- `glystats_go_ora_res`: Result from `glystats::gly_enrich_go()`.
- `glystats_kegg_ora_res`: Result from `glystats::gly_enrich_kegg()`.
- `glystats_reactome_ora_res`: Result from `glystats::gly_enrich_reactome()`.
- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. Enrichment analysis is first performed using `glystats::gly_enrich_go()`, `glystats::gly_enrich_kegg()`, or `glystats::gly_enrich_reactome()`, then the result is plotted.

Usage

```
plot_enrich(x, type = "dotplot", ...)

## S3 method for class 'glystats_go_ora_res'
plot_enrich(x, type = "dotplot", ...)

## S3 method for class 'glystats_kegg_ora_res'
plot_enrich(x, type = "dotplot", ...)

## S3 method for class 'glystats_reactome_ora_res'
plot_enrich(x, type = "dotplot", ...)

## S3 method for class 'glyexp_experiment'
plot_enrich(x, type = "dotplot", enrich_type = "go", stats_args = list(), ...)
```

Arguments

`x`	An object to be plotted.
`type`	The type of plot, one of "dotplot" (default), "barplot", or "network".
`...`	Additional arguments passed to underlying functions: • "dotplot": `enrichplot::dotplot()`

- "barplot": `enrichplot::barplot.enrichResult()`
- "network": `enrichplot::emapplot()`

`enrich_type`	The type of enrichment analysis, one of "go" (default), "kegg", or "reactome".
`stats_args`	A list of keyword arguments to pass to `glystats::gly_enrich_go()`, `glystats::gly_enrich_kegg()`, or `glystats::gly_enrich_reactome()`.

Value

A ggplot object.

`plot_heatmap` *Heatmap*

Description

Draw a heatmap from a `glyexp::experiment()`. Heatmap of log2-transformed expression values is plotted. Before plotting, zero-variance rows and columns are filtered out to ensure robust clustering.

Usage

```
plot_heatmap(x, ...)

## S3 method for class 'glyexp_experiment'
plot_heatmap(x, ...)
```

Arguments

`x`	A `glyexp::experiment()` object.
`...`	Other arguments passed to `pheatmap::pheatmap()`.

Value

A ggplot object.

`plot_logo` *Logo Plot*

Description

Draw a logo plot for all glycosites. Positions with insufficient flanking amino acids will be padded with "X". Currently supported data types:

- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. Logo plot of glycosylation sites is plotted.

Usage

```
plot_logo(x, n_aa = 5L, fasta = NULL, tax_id = 9606L, ...)

## S3 method for class 'glyexp_experiment'
plot_logo(x, n_aa = 5L, fasta = NULL, tax_id = 9606L, ...)
```

Arguments

`x`	An object to be plotted.
`n_aa`	The number of amino acids to the left and right of the glycosylation site. For example, if `n_aa = 5`, the resulting sequence will contain 11 amino acids.
`fasta`	The path to the FASTA file containing protein sequences. If `glyclean::add_site_seq()` has been called on the experiment, this argument can be omitted. When `site_sequence` is missing and `fasta` is `NULL`, UniProt.ws is used to fetch protein sequences automatically.
`tax_id`	The NCBI taxonomy ID used for UniProt.ws lookups. Defaults to 9606.
`...`	Additional arguments passed to `ggseqlogo::ggseqlogo()`.

Value

A ggplot object.

See Also

`ggseqlogo::ggseqlogo()`

`plot_oplsda` *OPLS-DA Plot*

Description

Draw a OPLS-DA scores plot. Currently supported data types:

- `glystats_oplsda_res`: Result from `glystats::gly_oplsda()`.
- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. OPLS-DA analysis is first performed using `glystats::gly_oplsda()`, then the result is plotted.

Usage

```
plot_oplsda(
  x,
  type = "scores",
  y_type = "o1",
  groups = NULL,
  group_col = NULL,
  ...
)

## S3 method for class 'glystats_oplsda_res'
plot_oplsda(
  x,
  type = "scores",
  y_type = "o1",
  groups = NULL,
  group_col = NULL,
  ...
)

## S3 method for class 'glyexp_experiment'
plot_oplsda(
  x,
  type = "scores",
  y_type = "o1",
  groups = NULL,
  group_col = NULL,
  stats_args = list(),
  ...
)
```

Arguments

`x`	An object to be plotted.

`type`	The type of plot, one of "loadings", "scores", "vip", "variance", or "s-plot". Defaults to "scores".
`y_type`	What to plot on the y-axis when `type` is "scores" or "loadings". Either "p2" or "o1". Defaults to "o1".
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group. Only applicable to "scores" and "loadings" types.
`group_col`	A character string specifying where to find the group information. If you uses `glystats::gly_oplsda()` on a `glyexp::experiment()` to get the result, sample information has already been added to the result. In this case, you can specify the column name in the sample information tibble to be used for coloring. If not provided, this function will try "group".
`...`	Ignored.
`stats_args`	A list of keyword arguments to pass to `glystats::gly_oplsda()`.

Value

A ggplot object.

`plot_pca` *PCA Plot*

Description

Draw a PCA scores plot. Currently supported data types:

- `glystats_pca_res`: Result from `glystats::gly_pca()`.
- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. PCA analysis is first performed using `glystats::gly_pca()`, then the result is plotted.

Usage

```
plot_pca(x, type = "individual", groups = NULL, group_col = NULL, ...)

## S3 method for class 'glystats_pca_res'
plot_pca(x, type = "individual", groups = NULL, group_col = NULL, ...)

## S3 method for class 'glyexp_experiment'
plot_pca(
  x,
  type = "individual",
  groups = NULL,
  group_col = NULL,
  stats_args = list(),
  ...
)
```

Arguments

`x`	An object to be plotted.
`type`	The type of plot, one of "screeplot", "individual", "variables", or "biplot".
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group. Only applicable to "individual" and "biplot" types. Passed to the `habillage` paramter of `factoextra::fviz_pca_ind()` or `factoextra::fviz_pca_biplot()`.
`group_col`	A character string specifying where to find the group information.
`...`	Additional arguments passed to underlying factoextra functions: • type = "screeplot": `factoextra::fviz_screeplot()`. • type = "individual": `factoextra::fviz_pca_ind()`. • type = "variables": `factoextra::fviz_pca_var()`. • type = "biplot": `factoextra::fviz_pca_biplot()`.
`stats_args`	A list of keyword arguments to pass to `glystats::gly_pca()`.

Value

A ggplot object.

See Also

`factoextra::fviz_screeplot()`, `factoextra::fviz_pca_ind()`, `factoextra::fviz_pca_var()`, `factoextra::fviz_pca_biplot()`

`plot_plsda` *PLS-DA Plot*

Description

Draw a PLS-DA scores plot. Currently supported data types:

- `glystats_plsda_res`: Result from `glystats::gly_plsda()`.
- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. PLS-DA analysis is first performed using `glystats::gly_plsda()`, then the result is plotted.

Usage

```
plot_plsda(x, type = "scores", groups = NULL, group_col = NULL, ...)

## S3 method for class 'glystats_plsda_res'
plot_plsda(x, type = "scores", groups = NULL, group_col = NULL, ...)

## S3 method for class 'glyexp_experiment'
plot_plsda(
  x,
```

```
  type = "scores",
  groups = NULL,
  group_col = NULL,
  stats_args = list(),
  ...
)
```

Arguments

`x`	An object to be plotted.
`type`	The type of plot, one of "loadings", "scores", "vip", or "variance". Defaults to "scores".
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group. Only applicable to "scores" and "loadings" types.
`group_col`	A character string specifying where to find the group information. If you uses `glystats::gly_plsda()` on a `glyexp::experiment()` to get the result, sample information has already been added to the result. In this case, you can specify the column name in the sample information tibble to be used for coloring. If not provided, this function will try "group".
`...`	Ignored.
`stats_args`	A list of keyword arguments to pass to `glystats::gly_plsda()`.

Value

A ggplot object.

`plot_roc` *ROC plot*

Description

Draw ROC curves. Only two groups are allowed. At most 10 variables can be plotted. Currently supported data types:

- `glystats_roc_res`: Result from `glystats::gly_roc()`.
- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. ROC analysis is first performed using `glystats::gly_roc()`, then the result is plotted.

Usage

```
plot_roc(x, type = "roc", auc_cutoff = 0.5, ...)

## S3 method for class 'glystats_roc_res'
plot_roc(x, type = "roc", auc_cutoff = 0.5, ...)

## S3 method for class 'glyexp_experiment'
plot_roc(x, type = "roc", auc_cutoff = 0.5, stats_args = list(), ...)
```

Arguments

`x`	An object to be plotted.
`type`	The type of plot, one of "dotplot" or "roc". Default is "roc".
`auc_cutoff`	The AUC cutoff. Default is 0.5.
`...`	Ignored.
`stats_args`	A list of keyword arguments to pass to `glystats::gly_roc()`.

Value

A ggplot object.

`plot_tsne` *t-SNE Plot*

Description

Draw a t-SNE scores plot. Currently supported data types:

- `glystats_tsne_res`: Result from `glystats::gly_tsne()`.
- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. t-SNE analysis is first performed using `glystats::gly_tsne()`, then the result is plotted.

Usage

```
plot_tsne(x, ...)

## S3 method for class 'glystats_tsne_res'
plot_tsne(x, groups = NULL, group_col = NULL, ...)

## S3 method for class 'glyexp_experiment'
plot_tsne(x, groups = NULL, group_col = NULL, stats_args = list(), ...)
```

Arguments

`x`	An object to be plotted.
`...`	Ignored.
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group.
`group_col`	A character string specifying where to find the group information. If you uses `glystats::gly_tsne()` on a `glyexp::experiment()` to get the result, sample information has already been added to the result. In this case, you can specify the column name in the sample information tibble to be used for coloring. If not provided, this function will try "group".
`stats_args`	A list of keyword arguments to pass to `glystats::gly_tsne()`.

Value

A ggplot object.

`plot_umap` *UMAP Plot*

Description

Draw a UMAP scores plot. Currently supported data types:

- `glystats_umap_res`: Result from `glystats::gly_umap()`.
- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. UMAP analysis is first performed using `glystats::gly_umap()`, then the result is plotted.

Usage

```
plot_umap(x, ...)

## S3 method for class 'glystats_umap_res'
plot_umap(x, groups = NULL, group_col = NULL, ...)

## S3 method for class 'glyexp_experiment'
plot_umap(x, groups = NULL, group_col = NULL, stats_args = list(), ...)
```

Arguments

`x`	An object to be plotted.
`...`	Ignored.
`groups`	A factor or character vector specifying group membership for each sample. If provided, the plot will be colored by group.
`group_col`	A character string specifying where to find the group information. If you uses `glystats::gly_umap()` on a `glyexp::experiment()` to get the result, sample information has already been added to the result. In this case, you can specify the column name in the sample information tibble to be used for coloring. If not provided, this function will try "group".
`stats_args`	A list of keyword arguments to pass to `glystats::gly_umap()`.

Value

A ggplot object.

`plot_volcano` *Volcano Plot*

Description

Draw a volcano plot. Currently supported data types:

- `glystats_ttest_res`: Result from `glystats::gly_ttest()`.
- `glystats_wilcox_res`: Result from `glystats::gly_wilcox()`.
- `glystats_limma_res`: Result from `glystats::gly_limma()`. Only two-group comparisons are supported.
- `glyexp_experiment`: Experiment created by `glyexp::experiment()`. Differential expression analysis is first performed using `glystats::gly_ttest()`, `glystats::gly_wilcox()`, or `glystats::gly_limma()` (controlled by the `test` argument), then the result is plotted.

Usage

```
plot_volcano(x, log2fc_cutoff = 1, p_cutoff = 0.05, p_col = "p_adj", ...)

## S3 method for class 'glyexp_experiment'
plot_volcano(
  x,
  log2fc_cutoff = 1,
  p_cutoff = 0.05,
  p_col = "p_adj",
  test = "limma",
  stats_args = list(),
  ...
)

## S3 method for class 'glystats_ttest_res'
plot_volcano(x, log2fc_cutoff = 1, p_cutoff = 0.05, p_col = "p_adj", ...)

## S3 method for class 'glystats_wilcox_res'
plot_volcano(x, log2fc_cutoff = 1, p_cutoff = 0.05, p_col = "p_adj", ...)

## S3 method for class 'glystats_limma_res'
plot_volcano(
  x,
  log2fc_cutoff = 1,
  p_cutoff = 0.05,
  p_col = "p_adj",
  contrast = NULL,
  ...
)
```

Arguments

`x`	An object to be plotted.
`log2fc_cutoff`	The log2 fold change cutoff. Defaults to 1.
`p_cutoff`	The p-value cutoff. Defaults to 0.05.
`p_col`	The column name for p-value. Defaults to "p_adj". Can also be "p_val" (raw p-values without multiple testing correction).
`...`	Other arguments passed to `EnhancedVolcano::EnhancedVolcano()`.
`test`	"ttest", "wilcox", or "limma". Default is "limma".
`stats_args`	A list of keyword arguments to pass to `glystats::gly_ttest()`, `glystats::gly_wilcox()`, or `glystats::gly_limma()`.
`contrast`	A character string specifying the contrast to plot, in the format of "group1_vs_group2". Must be one of the contrasts in the result. When there is only one contrast (two-group comparison), it can be NULL (default).

Value

A ggplot object.

Index