

Package: bixverse.plots (via r-universe)

October 21, 2026

Title Plotting functions for bixverse

Version 0.3.0

Description This package contains various plotting helpers for the core functions of the bixverse package.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports bixverse (>= 0.8.0), checkmate, data.table, fields, ggExtra, ggplot2 (>= 4.0.0), ggraph, ggrepel, grDevices, hexbin, igraph, magrittr, MASS, MetBrewer, patchwork, purrr, RColorBrewer, S7, scales, scattermore, stats, stringr, viridis, visNetwork, zeallot, rlang

Suggests ComplexHeatmap, fgsea, knitr, manifoldsR, pheatmap, quarto, UpSetR, tinytest

Config/roxygen2/version 8.1.0

URL <https://gregorlueg.github.io/bixverse.plots/>

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libglpk-dev make libicu-dev libuv1-dev libxml2-dev libssl-dev xz-utils zlib1g-dev libclang-dev

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

Date/Publication 2026-10-21 11:35:46 UTC

RemoteUrl <https://github.com/GregorLueg/bixverse.plots>

RemoteRef v0.3.0

RemoteSha 3c6dae363fa96c68aa565e343de55bc476168e72

Contents

density_plot_sc.CellQc	2
density_plot_sc.data.table	3
dot_plot_sc	4
embedding_plot_sc	5

enrichment_map_gsea	6
enrichment_map_oae	7
feature_plot_sc	8
feature_scatter_plot_sc	10
get_gsea_enrichment_data	11
heatmap_plot_sc	12
joint_plot_sc.CellQc	13
joint_plot_sc.data.table	14
label_centroids	15
milo_nhood_plot_sc	16
paga_plot_sc	18
params_plots	20
params_volcano	21
plot_binary_heatmap	22
plot_blitzgsea_es_null	23
plot_blitzgsea_null	24
plot_enrichment_map_ggraph	25
plot_enrichment_map_visnetwork	26
plot_gse_dotplot	26
plot_gsea_enrichment	27
prepare_mitch_scores	28
save_plot	29
save_plot_ls	29
scale_color_bx	30
scale_color_bx_c	31
scale_fill_bx	31
scale_fill_bx_c	32
stacked_violin_plot_sc	32
theme_bx	33
violin_plot_sc.CellQc	34
violin_plot_sc.data.table	35
volcano_plot	36
wrap_and_truncate	37
Index	38

density_plot_sc.CellQc

Per-metric density plots from a CellQc object

Description

Requires grouped data; outlier groups are read from per_group_stats.

Usage

```
## S3 method for class 'CellQc'
density_plot_sc(x, adjust_position_label = 0, palette = BX_PALETTES, ...)
```

Arguments

x	A CellQc object.
adjust_position_label	Numeric. x-offset for the labels (default: 0).
palette	String. Discrete palette for the group fills and labels. One of c("main", "sequential", "diverging", "viridis", "spectral"), see bx_colors() .
...	Ignored.

Value

A named list of ggplot objects, one per metric.

density_plot_sc.data.table

Density plot from a data.table

Description

Group-level outliers are recomputed from per-group medians.

Usage

```
## S3 method for class 'data.table'
density_plot_sc(
  x,
  grouping_column,
  variable,
  direction = c("twosided", "below", "above"),
  threshold = 3,
  var_name = NULL,
  log_scale = TRUE,
  adjust_position_label = 0,
  palette = BX_PALETTES,
  ...
)
```

Arguments

x	data.table. Input data containing the QC metric.
grouping_column	Character. Column used to group the densities.
variable	Character. Numeric column to plot on the x-axis.
direction	Character. One of "twosided", "below", "above".
threshold	Numeric. Number of MADs for outlier detection (default: 3).

var_name	Character. x-axis label (default: NULL).
log_scale	Logical. Apply a log10 x-axis (default: TRUE).
adjust_position_label	Numeric. x-offset for the labels (default: 0).
palette	String. Discrete palette for the group fills and labels. One of c("main", "sequential", "diverging", "viridis", "spectral"), see bx_colors() .
...	Ignored.

Value

A [ggplot](#) object.

dot_plot_sc	<i>Dot plot of marker gene expression across groups</i>
-------------	---

Description

Dot plot of marker gene expression across groups

Usage

```
dot_plot_sc(  
  object,  
  features,  
  grouping_variable,  
  feature_labels = NULL,  
  feature_grouping = NULL,  
  scale_exp = TRUE,  
  modality = c("rna", "adt"),  
  cluster_groups = TRUE,  
  palette = c("sequential", "spectral", "viridis", "diverging")  
)
```

Arguments

object	A single cell class.
features	Character vector. Gene IDs to plot.
grouping_variable	String. Obs column to group by.
feature_labels	Optional named character vector mapping gene ids to display labels (default: NULL).
feature_grouping	Optional named character vector mapping gene ids to grouping labels, e.g. cell type labels. If feature_labels is provided, the character vectors should contain the mapping of feature display labels to their respective groups (e.g. c(CD3E = "T cell", CD8A = "T cell", MS4A1 = "B cell", ...). (default: NULL).

scale_exp	Boolean. Whether to min-max scale mean expression per gene.
modality	String. One of c("rna", "adt").
cluster_groups	Boolean. Use hierarchical clustering on the grouping variable to re-order the group labels based on expression similarity.
palette	String. Continuous palette for the scaled expression. One of c("sequential", "spectral", "viridis", "diverging"), see bx_colors() .

Value

A [ggplot](#) object.

embedding_plot_sc	<i>Embedding plot coloured by an obs column</i>
-------------------	---

Description

Embedding plot coloured by an obs column

Usage

```
embedding_plot_sc(
  object,
  embedding,
  colour_by,
  label_by = NULL,
  discrete = NULL,
  embd_modality = c("rna", "adt", "wnn"),
  point_size = NULL,
  point_alpha = 0.5,
  raster = NULL,
  raster_dpi = c(512, 512),
  palette = NULL,
  label_size = 3,
  label_color = "black",
  label_font = "bold"
)
```

Arguments

object	A single cell class.
embedding	String. Name of the embedding (e.g. "umap").
colour_by	String. Obs column to colour by.
label_by	String. Optional obs column to label by. (default: NULL).
discrete	Optional boolean. Force a discrete scale by coercing colour_by to a factor. NULL (default) picks the scale from the column type.

embd_modality	String. Modality the embedding is pulled from. One of <code>c("rna", "adt", "wnn")</code> . Use "wnn" for WNN-derived embeddings.
point_size	Optional numeric. Defines the point size. If not provided, will be auto-determined.
point_alpha	Numeric. Defines the alpha.
raster	Optional boolean. Shall the plot be rasterised. If NULL and number of cells is larger than 1e5, defaults to TRUE.
raster_dpi	Two numerics. Pixel resolution for rasterized plots, passed to <code>geom_scattermore()</code> . Default is <code>c(512, 512)</code> .
palette	Optional string. Palette for the colour scale, see <code>bx_colors()</code> . NULL (default) resolves to "main" if colour_by is discrete and "sequential" if it is continuous.
label_size	Numeric. Size of the labels
label_color	String. Color fo the labels.
label_font	String. Font of the labels. <code>bixverse::extract_embedding_data()</code> .

Value

A `ggplot` object.

enrichment_map_gsea	<i>Generate enrichment map igragh (for GSEA)</i>
---------------------	--

Description

Helper function to generate an enrichment map based on GSEA results. Similar enriched gene sets are clustered together via their Jaccard similarity (alternatively overlap coefficient) and the function returns an `igraph` object for subsequent visualisations.

Usage

```
enrichment_map_gsea(
  res,
  threshold,
  pathways,
  overlap_coefficient = FALSE,
  min_sim = 0.2,
  resolution = 1,
  layout_func = igraph::layout_with_fr,
  ...
)
```

Arguments

<code>res</code>	data.table with the enrichment results. Needs to have the columns <code>c("pathway_name", "nes", "fdr")</code> , i.e. the output of <code>bixverse::calc_fgsea()</code> or <code>bixverse::calc_blitzgsea()</code> . A <code>geneset_name</code> column is accepted with a deprecation warning.
<code>threshold</code>	Numeric. The FDR threshold you wish to filter for.
<code>pathways</code>	Named list. The original pathway list used for the calculation of the overenrichment analysis.
<code>overlap_coefficient</code>	Boolean. Shall the overlap coefficient be used instead of the Jaccard similarity.
<code>min_sim</code>	Numeric. Minimum similarity between two gene sets to be connected.
<code>resolution</code>	Numeric. The resolution parameter for the Louvain clustering.
<code>layout_func</code>	Layout function. Please see <code>igraph::add_layout_()</code> for options. This one will be used to layout the graph.
<code>...</code>	Further parameters to forward to <code>wrap_and_truncate()</code> .

Value

igraph object representing the enrichment map.

<code>enrichment_map_oae</code>	<i>Generate enrichment map igraph (for overenrichment tests)</i>
---------------------------------	--

Description

Helper function to generate an enrichment map based on overenrichment results. Similar enriched gene sets are clustered together via their Jaccard similarity (alternatively overlap coefficient) and the function returns an igraph object for subsequent visualisations.

Usage

```
enrichment_map_oae(
  res,
  threshold,
  pathways,
  overlap_coefficient = FALSE,
  min_sim = 0.2,
  resolution = 1,
  layout_func = igraph::layout_with_fr,
  ...
)
```

Arguments

<code>res</code>	data.table with the enrichment results. Needs to have the columns <code>c("gene_set_name", "fdr")</code> .
<code>threshold</code>	Numeric. The FDR threshold you wish to filter for.
<code>pathways</code>	Named list. The original pathway list used for the calculation of the overenrichment analysis.
<code>overlap_coefficient</code>	Boolean. Shall the overlap coefficient be used instead of the Jaccard similarity.
<code>min_sim</code>	Numeric. Minimum similarity between two gene sets to be connected.
<code>resolution</code>	Numeric. The resolution parameter for the Louvain clustering.
<code>layout_func</code>	Layout function. Please see <code>igraph::add_layout_()</code> for options. This one will be used to layout the graph.
<code>...</code>	Further parameters to forward to <code>wrap_and_truncate()</code> .

Value

igraph object representing the enrichment map.

<code>feature_plot_sc</code>	<i>Feature plot over an embedding</i>
------------------------------	---------------------------------------

Description

Plots the expression of one or more features over an embedding. By default every feature gets its own panel and its own colour bar (`scale_mode = "free"`), so a weakly expressed gene is not flattened by whatever the loudest gene in the set happens to be. Use `scale_mode = "shared"` for a single faceted plot with one colour bar across all features.

Usage

```
feature_plot_sc(
  object,
  features,
  embedding,
  feature_labels = NULL,
  scale = FALSE,
  clip = NULL,
  expr_modality = c("rna", "adt"),
  embd_modality = c("rna", "adt", "wnn"),
  point_size = NULL,
  point_alpha = 0.5,
  raster = NULL,
  raster_dpi = c(512, 512),
  label_by = NULL,
```

```

    label_size = 3,
    label_color = "black",
    label_font = "bold",
    highlight_features = FALSE,
    highlight_quantile = 0.25,
    scale_mode = c("free", "shared"),
    palette = c("sequential", "spectral", "viridis", "diverging"),
    ncol = NULL,
    ...
)

```

Arguments

<code>object</code>	A single cell class.
<code>features</code>	Character vector. Gene/feature IDs to plot, taken from <code>expr_modality</code> .
<code>embedding</code>	String. Name of the embedding.
<code>feature_labels</code>	Optional named character vector mapping gene ids to display labels (default: <code>NULL</code>).
<code>scale</code>	Boolean. Whether to z-score the expression values.
<code>clip</code>	Optional numeric. Clip z-scores if <code>scale = TRUE</code> .
<code>expr_modality</code>	String. Modality the expression is pulled from. One of <code>c("rna", "adt")</code> .
<code>embd_modality</code>	String. Modality the embedding is pulled from. One of <code>c("rna", "adt", "wnn")</code> . Use "wnn" for WNN-derived embeddings.
<code>point_size</code>	Optional numeric. Defines the point size. If not provided, will be auto-determined.
<code>point_alpha</code>	Numeric. Defines the alpha.
<code>raster</code>	Optional boolean. Shall the plot be rasterised. If <code>NULL</code> and number of cells is larger than <code>1e5</code> , defaults to <code>TRUE</code> .
<code>raster_dpi</code>	Two numerics. Pixel resolution for rasterized plots, passed to <code>geom_scattermore()</code> . Default is <code>c(512, 512)</code> .
<code>label_by</code>	String. Optional obs column to label by. (default: <code>NULL</code>).
<code>label_size</code>	Numeric. Size of the labels
<code>label_color</code>	String. Color fo the labels.
<code>label_font</code>	String. Font of the labels.
<code>highlight_features</code>	Boolean. Shall the features be more strongly highlighted. Useful for sparsely expressed genes.
<code>highlight_quantile</code>	Numeric between <code>[0, 1]</code> . Defines the threshold.
<code>scale_mode</code>	String. One of <code>c("free", "shared")</code> . With "free" each feature is drawn as its own plot with its own colour bar and the panels are combined with <code>patchwork::wrap_plots()</code> . With "shared" all features go into one faceted plot with a single colour bar fitted to the pooled expression.
<code>palette</code>	String. Continuous palette for the expression values. One of <code>c("sequential", "spectral", "viridis", "diverging")</code> , see <code>bx_colors()</code> .

<code>ncol</code>	Optional integer. Number of columns of the panel grid. Only has an effect if <code>scale_mode = "free"</code> . If NULL, patchwork picks the layout.
<code>...</code>	Additional arguments forwarded to <code>bixverse::extract_feature_plot_data()</code> and onward to <code>get_embedding()</code> . Do not pass <code>modality</code> here; use <code>embd_modality</code> instead.

Value

A `patchwork` object if `scale_mode = "free"`, otherwise a `ggplot` object.

`feature_scatter_plot_sc`

Scatter / hex plot of two features against each other

Description

Plots two features against each other (typical use case: protein vs mRNA on ADT/RNA data). Each feature may carry a `_rna` or `_adt` suffix to select its modality independently; unsuffixed features fall back to modality. With `geom = "density"` points are coloured by 2D KDE, with `geom = "hex"` cells are binned into hexagons.

Usage

```
feature_scatter_plot_sc(
  object,
  feature_1,
  feature_2,
  geom = c("density", "hex"),
  remove_zeros = TRUE,
  smoothness = 10,
  bins = 60,
  modality = c("rna", "adt"),
  point_size = 2.5,
  point_alpha = 0.5,
  raster = NULL,
  raster_dpi = c(512, 512),
  palette = c("viridis", "sequential", "spectral", "diverging")
)
```

Arguments

<code>object</code>	A single cell class.
<code>feature_1</code>	String. First feature (x-axis), optionally <code>_rna</code> / <code>_adt</code> suffixed.
<code>feature_2</code>	String. Second feature (y-axis), optionally <code>_rna</code> / <code>_adt</code> suffixed.
<code>geom</code>	String. "density" or "hex" (default: "density").
<code>remove_zeros</code>	Boolean. Drop cells where both features are zero (default: TRUE).

smoothness	Numeric. Bandwidth multiplier for the KDE (default: 10). Only used for geom = "density".
bins	Numeric. Number of hex bins (default: 60). Only used for geom = "hex".
modality	String. Fallback modality for unsuffixed features. One of c("rna", "adt").
point_size	Numeric. Point size for density plots (default: 2.5).
point_alpha	Numeric. Alpha for density plots (default: 0.5).
raster	Optional boolean. Shall the plot be rasterised. If NULL and n_cells > 1e5, defaults to TRUE. Only applies to geom = "density".
raster_dpi	Two numerics. Pixel resolution for rasterised plots (default: c(512, 512)).
palette	String. Continuous palette for the density / count scale. One of c("viridis", "sequential", "spectral", "diverging"), see bx_colors() .

Value

A [ggplot](#) object.

get_gsea_enrichment_data

Helper function to get the plot data for GSEA plots

Description

Helper function to get the plot data for GSEA plots

Usage

```
get_gsea_enrichment_data(
  stats,
  pathways,
  pathways_of_interest,
  gsea_results = NULL,
  gsea_param = 1
)
```

Arguments

stats	Named numeric vector. The gene level statistic.
pathways	List. A named list with each element containing the genes for this pathway.
pathways_of_interest	String vector. Names of the pathways to plot. These strings need to be represented in the names of pathways.
gsea_results	Optional data.table with the bixverse GSEA results. If provided, the FDR and NES for the given pathway of interest will be also added to the plot.
gsea_param	Numeric. Defaults to 1.

Value

A named list of gsea_par_plot_data objects, one per pathway that survived the size filters. Each one is a list with the following elements:

- curve_dt - data.table with the running enrichment score, columns rank and ES.
- ticks_dt - data.table with the positions of the pathway genes, columns rank and stat.
- stats_dt - data.table with the ranked statistic, columns rank and stat.
- key_points - Named numeric with pos_es, neg_es and spread_es, plus nes and fdr if gsea_results was provided.
- additional_label - String or NULL. The NES/FDR annotation.

heatmap_plot_sc

Heatmap of marker genes x cells across groups

Description

The classic marker heatmap to show which cluster is what: genes as rows, cells as columns, cells grouped by grouping_variable with white gaps and a colour strip per group. With feature_grouping the genes get grouped per cell type, too. Every block of cell group x gene group is drawn as one raster, not one tile per cell.

PDF viewers tend to smooth embedded rasters. If the blocks look blurry in a PDF, save as PNG instead.

Usage

```
heatmap_plot_sc(
  object,
  features,
  grouping_variable,
  feature_labels = NULL,
  feature_grouping = NULL,
  scale = TRUE,
  clip = 2.5,
  modality = c("rna", "adt"),
  max_cells_per_group = NULL,
  seed = 42L,
  group_gap = 0.002,
  show_feature_labels = NULL,
  palette = c("diverging", "spectral", "viridis", "sequential")
)
```

Arguments

object	A single cell class.
features	Character vector. Gene IDs to plot, one row each, in this order.
grouping_variable	String. Obs column to group the cells by. Factor levels set the group order.
feature_labels	Optional named character vector mapping gene ids to display labels (default: NULL).
feature_grouping	Optional named character vector mapping gene ids to grouping labels, e.g. cell type labels. If feature_labels is provided, the character vectors should contain the mapping of feature display labels to their respective groups (e.g. c(CD3E = "T cell", CD8A = "T cell", MS4A1 = "B cell", ...). Groups are ordered by first appearance. (default: NULL).
scale	Boolean. Whether to z-score the expression values per gene.
clip	Optional numeric. Clip z-scores if scale = TRUE. Without it a handful of extreme cells flatten the colour range.
modality	String. One of c("rna", "adt").
max_cells_per_group	Optional integer. Subsample every group to at most this many cells. NULL (default) draws all cells.
seed	Integer. Seed for the subsampling.
group_gap	Numeric between [0, 0.1]. Width of each gap between groups as fraction of the axis length. 0 removes the gaps.
show_feature_labels	Optional boolean. Shall the gene names be shown. NULL (default) shows them for up to 80 genes.
palette	String. Continuous palette for the expression values. One of c("diverging", "spectral", "viridis", "sequential"), see bx_colors() .

Value

A [ggplot](#) object.

joint_plot_sc.CellQc *Joint QC plot from a CellQc object*

Description

Joint QC plot from a CellQc object

Usage

```
## S3 method for class 'CellQc'
joint_plot_sc(
  x,
  library_size = "log10_lib_size",
  nb_features = "log10_nnz",
  palette = c("sequential", "spectral", "viridis", "diverging"),
  ...
)
```

Arguments

x	A CellQc object.
library_size	Character. Column with the library size per cell.
nb_features	Character. Column with the number of features per cell.
palette	String. Continuous palette for the hexbin fill. One of c("sequential", "spectral", "viridis", "diverging"), see bx_colors() .
...	Ignored.

Value

A ggExtraPlot object.

joint_plot_sc.data.table

Joint QC plot from a data.table

Description

Joint QC plot from a data.table

Usage

```
## S3 method for class 'data.table'
joint_plot_sc(
  x,
  library_size = "lib_size",
  nb_features = "nnz",
  log_scale = FALSE,
  palette = c("sequential", "spectral", "viridis", "diverging"),
  ...
)
```

Arguments

x	data.table. Input data containing the QC metrics.
library_size	Character. Column with the library size per cell.
nb_features	Character. Column with the number of features per cell.
log_scale	Logical. Log10-transform both axes (default: FALSE).
palette	String. Continuous palette for the hexbin fill. One of c("sequential", "spectral", "viridis", "diverging"), see bx_colors() .
...	Ignored.

Value

A ggExtraPlot object.

label_centroids	<i>Label Centroids in Scatter Plots</i>
-----------------	---

Description

Adds boxed text labels at the centroid position of each group in a scatter plot. Computes group centroids with data.table for efficient summarisation. Useful for labelling cluster centres in embedding or dimensionality reduction plots (UMAP, t-SNE, PCA).

Usage

```
label_centroids(
  data = NULL,
  label_by,
  colour = "black",
  fill = "white",
  alpha = 0.5,
  label.size = 0,
  size = 4,
  fontface = "bold",
  ...
)
```

Arguments

data	A data.frame or data.table containing the scatter plot points. If NULL, the data from the parent ggplot is used. Must contain the columns referenced by the parent plot's x/y aesthetics and by label_by.
label_by	Character. Name of the (discrete) column to group and label by.
colour	Text colour. Default: "black".
fill	Box fill colour. Default: "white".

<code>alpha</code>	Box fill transparency in $[0, 1]$. Default: <code>0.5</code> .
<code>label.size</code>	Box border line width in mm. Set to <code>0</code> to hide the border. Default: <code>0</code> .
<code>size</code>	Text size in mm. Default: <code>4</code> .
<code>fontface</code>	Font face. Default: <code>"bold"</code> .
<code>...</code>	Additional arguments passed to geom_label .

Value

A ggplot layer.

Examples

```
## Not run:
embedding_plot_sc(
  object = sc_object,
  embedding = "umap",
  colour_by = "donor_id"
) +
  label_centroids(label_by = "donor_id")

## End(Not run)
```

<code>milo_nhood_plot_sc</code>	<i>miloR neighbourhood graph over an embedding</i>
---------------------------------	--

Description

The Milo neighbourhood graph figure: every tested neighbourhood sits at its index cell in an embedding, sized by its cell count and connected to the neighbourhoods it shares cells with. With `colour_by = "logFC"` the significant neighbourhoods are filled by their logFC and drawn on top, strongest last, while the rest stay white underneath.

Usage

```
milo_nhood_plot_sc(
  object,
  milo_res,
  embedding = "umap",
  alpha = 0.1,
  overlap = 1L,
  colour_by = c("logFC", "majority_celltype"),
  show_cells = TRUE,
  embd_modality = c("rna", "adt", "wnn"),
  point_size = NULL,
  point_alpha = 0.4,
  raster = NULL,
```

```

    raster_dpi = c(512, 512),
    cell_colour = "grey90",
    edge_colour = "grey50",
    edge_width = c(0.1, 1.5),
    size_range = c(0.5, 4),
    palette = NULL
  )

```

Arguments

<code>object</code>	A single cell class. The one the <code>mihoR</code> result was built on.
<code>miho_res</code>	<code>mihoR</code> class, run through <code>bixverse::test_nhoods()</code> .
<code>embedding</code>	String. Name of the embedding to position the nodes in.
<code>alpha</code>	Numeric. Spatial FDR threshold for a neighbourhood to count as significant (default: 0.1).
<code>overlap</code>	Integer. Minimum number of shared cells for an edge (default: 1L).
<code>colour_by</code>	String. One of <code>c("logFC", "majority_celltype")</code> . The latter needs <code>bixverse::add_nhoods_info()</code> to have been run.
<code>show_cells</code>	Boolean. Draw the cells underneath the graph (default: TRUE).
<code>embd_modality</code>	String. One of <code>c("rna", "adt", "wnn")</code> . Modality the embedding is pulled from.
<code>point_size</code>	Optional numeric. Size of the cells. If not provided, will be auto-determined.
<code>point_alpha</code>	Numeric. Alpha of the cells (default: 0.4).
<code>raster</code>	Optional boolean. Shall the cell layer be rasterised. If NULL and the number of cells is larger than 1e5, defaults to TRUE.
<code>raster_dpi</code>	Two numerics. Pixel resolution for rasterized plots, passed to <code>geom_scattermore()</code> . Default is <code>c(512, 512)</code> .
<code>cell_colour</code>	String. Colour of the cell layer (default: "grey90").
<code>edge_colour</code>	String. Colour of the edges (default: "grey50").
<code>edge_width</code>	Two numerics. Range the edge widths are scaled into (default: <code>c(0.1, 1.5)</code>).
<code>size_range</code>	Two numerics. Range the node sizes are scaled into (default: <code>c(0.5, 4)</code>).
<code>palette</code>	Optional string. Palette for the nodes, see <code>bx_colors()</code> . NULL (default) resolves to "diverging" for logFC, centred at zero, and "main" for the cell types.

Value

A `ggplot` object.

References

Dann, et al., Nat Biotechnol, 2022.

paga_plot_sc

PAGA abstracted graph over an embedding

Description

Draws the PAGA graph with every cluster sitting at the centroid of its cells in an embedding, over a faint scatter of the cells themselves. A free layout of the abstracted graph puts the nodes somewhere arbitrary, which the reader then has to relate back to the embedding by hand. This does not.

The abstracted graph is close to complete on real data, so `threshold` is doing real work and dropping it to zero gives a hairball. `tree_only` is the shortcut to the backbone.

Usage

```
paga_plot_sc(
  object,
  paga_res,
  embedding = "umap",
  threshold = 0.01,
  tree_only = FALSE,
  node_colour_by = NULL,
  show_cells = TRUE,
  label = TRUE,
  embd_modality = c("rna", "adt", "wnn"),
  centroid = c("median", "mean"),
  point_size = NULL,
  point_alpha = 0.4,
  raster = NULL,
  raster_dpi = c(512, 512),
  cell_colour = "grey80",
  edge_colour = "grey40",
  edge_width = c(0.2, 3),
  max_node_size = 12,
  palette = NULL,
  label_size = 3,
  label_color = "black",
  label_font = "bold"
)
```

Arguments

<code>object</code>	A single cell class.
<code>paga_res</code>	PagaRes class. The output of <code>bixverse::run_paga_sc()</code> , run on this object.
<code>embedding</code>	String. Name of the embedding to position the nodes in.
<code>threshold</code>	Numeric. Edges below this connectivity are dropped (default: 0.01).
<code>tree_only</code>	Boolean. Draw the maximum spanning forest rather than the full abstracted graph (default: FALSE).

node_colour_by	Optional string. A numeric obs column summarised per cluster, e.g. "palantir_pseudotime", giving continuously coloured nodes. NULL (default) colours the nodes discretely by cluster.
show_cells	Boolean. Draw the cells underneath the graph (default: TRUE).
label	Boolean. Label the nodes with their cluster (default: TRUE).
embd_modality	String. One of c("rna", "adt", "wnn"). Modality the embedding is pulled from.
centroid	String. One of c("median", "mean"). How a cluster's position is summarised. Median by default, since embeddings throw stragglers that drag a mean off its cluster.
point_size	Optional numeric. Size of the cells. If not provided, will be auto-determined.
point_alpha	Numeric. Alpha of the cells (default: 0.4).
raster	Optional boolean. Shall the cell layer be rasterised. If NULL and the number of cells is larger than 1e5, defaults to TRUE.
raster_dpi	Two numerics. Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).
cell_colour	String. Colour of the cell layer (default: "grey80").
edge_colour	String. Colour of the edges (default: "grey40").
edge_width	Two numerics. Range the edge widths are scaled into (default: c(0.2, 3)).
max_node_size	Numeric. Size of the largest node. Node <i>area</i> tracks the cell count, which is what people read off it (default: 12).
palette	Optional string. Palette for the nodes, see bx_colors() . NULL (default) resolves to "main" for the discrete case and "sequential" when node_colour_by is given.
label_size	Numeric. Size of the labels.
label_color	String. Colour of the labels.
label_font	String. Font of the labels.

Value

A [ggplot](#) object.

References

Wolf, et al., Genome Biol., 2019.

params_plots

Wrapper function for standard plot parameters

Description

Wrapper function for standard plot parameters

Usage

```
params_plots(
  width = 5,
  height = 5,
  file_type = c(".png", ".pdf"),
  unit = c("in", "px", "cm"),
  res = 450L,
  create_dir = TRUE
)
```

Arguments

width	Numeric. Width of the plot. Defaults to 5.0.
height	Numeric. Height of the plot. Defaults to 5.0.
file_type	String. Plot type to save. Might be expanded to other file types. One of c(".png", ".pdf"). Defaults to ".png".
unit	String. Unit type for width and height. One of c("in", "px", "cm"). Defaults to "in".
res	Integer. Resolution for PNGs. Defaults to 450L.
create_dir	Boolean. Shall the plot directory be generated recursively. Defaults to TRUE.

Value

A named list with the following elements:

- width - Numeric. Width of the plot. Defaults to 5.0.
- height - Numeric. Height of the plot. Defaults to 5.0.
- file_type - String. Plot type to save. Might be expanded to other file types. One of c(".png", ".pdf"). Defaults to ".png".
- unit - String. Unit type for width and height. One of c("in", "px", "cm"). Defaults to "in".
- res - Integer. Resolution for PNGs. Defaults to 450L.
- create_dir - Boolean. Shall the plot directory be generated recursively. Defaults to TRUE.

params_volcano

*Wrapper function for volcano plot parameters***Description**

Wrapper function for volcano plot parameters

Usage

```
params_volcano(
  x_axis = "log2FC",
  y_axis = "FDR",
  colour = NULL,
  label_column = NULL,
  top_features_to_label = NULL
)
```

Arguments

x_axis	String. Column holding the effect size (e.g. "log2FC"). Defaults to "log2FC".
y_axis	String. Column holding the raw significance values (e.g. "FDR", "fdr", "q_value"). The function applies $-\log_{10}()$ internally. Defaults to "FDR".
colour	String or NULL. Column to colour points by (continuous gradient). If NULL, points are coloured by x_axis. Defaults to NULL.
label_column	String or NULL. Column holding feature labels. Required if top_features_to_label is set. Defaults to NULL.
top_features_to_label	Integer or NULL. Number of features to label, ranked by y_axis ascending (most significant first). Defaults to NULL.

Value

A named list with the following elements:

- x_axis - String. Column holding the effect size (e.g. "log2FC"). Defaults to "log2FC".
- y_axis - String. Column holding the raw significance values (e.g. "FDR", "fdr", "q_value"). The function applies $-\log_{10}()$ internally. Defaults to "FDR".
- colour - String or NULL. Column to colour points by (continuous gradient). If NULL, points are coloured by x_axis. Defaults to NULL.
- label_column - String or NULL. Column holding feature labels. Required if top_features_to_label is set. Defaults to NULL.
- top_features_to_label - Integer or NULL. Number of features to label, ranked by y_axis ascending (most significant first). Defaults to NULL.

plot_binary_heatmap *Binary heatmap, e.g. SCENIC regulon on/off calls*

Description

Draws a samples x features on/off matrix in the style of the SCENIC paper: features as rows, samples as columns, black for on and white for off. Every block of sample group x feature group is drawn as one raster, so the plot stays light no matter how many samples there are, and the blocks are separated by white gaps. Groups get a thin colour strip with their labels written next to it. Sample and feature groups share one colour mapping, so a feature group with the same name as a sample group gets the same colour. `group_annotation` adds a second strip with a legend between the matrix and the sample group strip, e.g. the response group of each cell line.

Filtering, ordering and binning happen in `bixverse::extract_binary_heatmap_data()`. If the samples were binned the cells show the fraction of samples on as a grey ramp.

PDF viewers tend to smooth embedded rasters. If the blocks look blurry in a PDF, save as PNG instead.

Usage

```
plot_binary_heatmap(
  x,
  sample_groups = NULL,
  feature_groups = NULL,
  heatmap_params = bixverse::params_binary_heatmap(),
  show_feature_labels = NULL,
  group_gap = 0.001,
  group_annotation = NULL,
  annotation_colours = NULL,
  palette = c("main", "sequential", "diverging", "viridis", "spectral"),
  .verbose = FALSE
)
```

Arguments

<code>x</code>	Either a <code>BinaryHeatmapData</code> from <code>bixverse::extract_binary_heatmap_data()</code> or a logical matrix of samples x features, which then gets passed through it.
<code>sample_groups</code>	Optional named character vector or factor mapping samples to groups. Only used if <code>x</code> is a matrix.
<code>feature_groups</code>	Optional named character vector or factor mapping features to groups. Only used if <code>x</code> is a matrix.
<code>heatmap_params</code>	List. Output of <code>bixverse::params_binary_heatmap()</code> . Only used if <code>x</code> is a matrix.
<code>show_feature_labels</code>	Optional boolean. Shall the feature names be shown. <code>NULL</code> (default) shows them for up to 80 features.

group_gap	Numeric between $[0, 0.1]$. Width of each gap between groups as fraction of the axis length. Applied per boundary, so with many groups keep it small. 0 removes the gaps.
group_annotation	Optional named character vector or factor mapping every sample group to a coarser label. Factor levels set the legend order. Ignored if there is only one sample group.
annotation_colours	Optional named character vector of colours for the group_annotation labels. NULL (default) uses viridis.
palette	String. Discrete palette for the group strips, see bx_colors() .
.verbose	Boolean. Controls verbosity of the extraction.

Value

A [ggplot](#) object.

References

Aibar, et al., Nat Methods, 2017

plot_blitzgsea_es_null

Plot observed enrichment scores against the blitzGSEA null

Description

For each pathway of interest, draws the fitted null density at that pathway's size and drops the observed enrichment score on top, with the tail beyond it shaded. This is where the p-value comes from: blitzGSEA never permutes per pathway, it reads the score off the gamma tail interpolated to the set size.

The raw permutation scores never cross the Rust boundary, so this is the fitted density, not a histogram of null draws.

Usage

```
plot_blitzgsea_es_null(
  null_model,
  res,
  pathways_of_interest,
  n_points = 512L,
  text_size = 3
)
```

Arguments

null_model	BlitzGseaNull object from <code>bixverse::blitzgsea_calibrate()</code> . Has to be the null the results were scored against.
res	data.table. Output of <code>bixverse::calc_blitzgsea()</code> . Needs the columns <code>c("pathway_name", "es", "size", "pvals", "fdr")</code> .
pathways_of_interest	String vector. Names of the pathways to plot. These need to be represented in <code>res\$pathway_name</code> .
n_points	Integer. Resolution of the density curve. Defaults to 512L.
text_size	Numeric. Size of the annotation text. Defaults to 3.

Value

A named list of ggplot objects, one per element of `pathways_of_interest`.

References

Lachmann, et al., Bioinformatics, 2022

plot_blitzgsea_null	<i>Plot the blitzGSEA calibration</i>
---------------------	---------------------------------------

Description

Diagnostic for a BlitzGseaNull. Shows the smoothed gamma parameters against anchor set size for both tails, plus the fraction of positive null scores. A fit that sags at one end of the size range is obvious here and invisible in the results table.

The KS goodness-of-fit p-values are in the subtitle. Low values mean the gamma tails do not describe the permutation scores well and the p-values are optimistic. More permutations is the usual fix.

Usage

```
plot_blitzgsea_null(null_model, line_size = 0.8, point_size = 1.2)
```

Arguments

null_model	BlitzGseaNull object from <code>bixverse::blitzgsea_calibrate()</code> .
line_size	Numeric. Line width. Defaults to 0.8.
point_size	Numeric. Point size for the anchor knots. Defaults to 1.2.

Value

A ggplot object, faceted over the parameter type.

References

Lachmann, et al., Bioinformatics, 2022

plot_enrichment_map_ggraph

igraph enrichment map to ggraph plot

Description

Takes in the output from `enrichment_map_oae()` or `enrichment_map_gsea()` and generates a ggraph object for subsequent saving, etc.

Usage

```
plot_enrichment_map_ggraph(
  g,
  label_nodes = "adaptive",
  labels_to_include = NULL,
  adaptive_thresholds = c(`15` = 3, `5` = 2, `2` = 1, `1` = 0),
  font_size = 4,
  ...
)
```

Arguments

<code>g</code>	igraph. Output from enrichment_map functions.
<code>label_nodes</code>	String. Controls which nodes to label. Options: • "all": Label all nodes • "adaptive": Adaptive labelling based on community size (default) • NULL: No labels • Integer: Label top N nodes by size
<code>labels_to_include</code>	Optional string. These are labels you want to include no matter what.
<code>adaptive_thresholds</code>	Named numeric. The names indicate the community size and the values how many pathways per community to show. An example would be <code>c("15" = 3, "5" = 2, "2" = 1, "1" = 0)</code> , which is also the default.
<code>font_size</code>	Numeric. Font size of the labels on top of the enrichment map.
<code>...</code>	Other parameters you wish to forward to <code>ggraph::geom_node_text()</code> .

Value

A ggplot2 object with the enrichment map

plot_enrichment_map_visnetwork	<i>igraph enrichment map to VisNetwork interactive network</i>
--------------------------------	--

Description

Takes in the output from [enrichment_map_oae\(\)](#) or [enrichment_map_gsea\(\)](#) and generates an interactive VisNetwork widget.

Usage

```
plot_enrichment_map_visnetwork(g)
```

Arguments

`g` igraph. Output from `enrichment_map` functions.

Value

The interactive visnetwork

plot_gse_dotplot	<i>Generate GSE dotplots</i>
------------------	------------------------------

Description

This function can take in the output of [bixverse::gse_hypergeometric\(\)](#) or [bixverse::gse_hypergeometric_list\(\)](#) and generates in the former case a single plot and in the latter case a list of plots per target set.

Usage

```
plot_gse_dotplot(
  res,
  size_range = c(2, 5),
  viridis_option = "D",
  direction = -1,
  max_terms = NULL,
  .verbose = TRUE,
  ...
)
```

Arguments

res	data.table with the enrichment results. Needs to have the columns c("hits", "target_set_lengths", "gene_set_name", "gene_set_lengths", "fdr").
size_range	Numerical vector of size 2. Defines the size range for the dots in the plot.
viridis_option	String. The option to forward to ggplot2::scale_fill_viridis_c() .
direction	1 or -1. The direction in the colour palette.
max_terms	Optional integer. Show only the this many most significant gene sets. Applied per target set when several were tested. Defaults to NULL, i.e. everything that passed the enrichment threshold.
.verbose	Boolean. Controls verbosity of the function.
...	Further parameters to forward to wrap_and_truncate() , which shortens the gene set labels.

Value

If the output of [bixverse::gse_hypergeometric_list\(\)](#) was provided, a list of dotplots per target gene set. Otherwise, a single GSE OAE dot plot.

plot_gsea_enrichment *Plot GSEA enrichment results*

Description

Helper function to create the classical GSEA plots for a set of pathways of interest. You can also provide the underlying GSEA results to add the FDR and NES information to the plot.

Usage

```
plot_gsea_enrichment(
  stats,
  pathways,
  pathways_of_interest,
  gsea_results = NULL,
  gsea_param = 1,
  tick_size = 0.2,
  text_size = 5
)
```

Arguments

stats	Named numeric vector. The gene level statistic.
pathways	List. A named list with each element containing the genes for this pathway.
pathways_of_interest	String vector. Names of the pathways to plot. These strings need to be represented in the names of pathways.

gsea_results	Optional data.table with the bixverse GSEA results. If provided, the FDR and NES for the given pathway of interest will be also added to the plot.
gsea_param	Numeric. Defaults to 1.
tick_size	Numeric. The tick size. Defaults to 0.2.
text_size	Numeric. The text size. Defaults to 5. Only relevant when gsea_results is provided.

Value

A named list of ggplot objects, one per element of pathways_of_interest that survived the size filters.

prepare_mitch_scores	<i>Generate adjusted scores for plotting mitch results</i>
----------------------	--

Description

Function that takes results from `bixverse::calc_mitch()`, extracts the individual scores and p-values for each contrast, runs an FDR correction on top of the p-values and sets scores above the provided FDR threshold to 0.

Usage

```
prepare_mitch_scores(res, fdr_threshold = 0.05)
```

Arguments

`res` data.table. Output of `bixverse::calc_mitch()`.

`fdr_threshold` Numeric. The FDR threshold you want to apply.

Value

A list with the following two elements

- `fdr_corrections` - FDR-corrected scores as a matrix of the individual contrast p-values.
- `adj_scores` - The extract scores per contrast with scores above the individual FDR threshold set to 0.

save_plot	<i>General helper to save all types of plots to file</i>
-----------	--

Description

General helper to save all types of plots to file

Usage

```
save_plot(plot, file_name = NULL, path, plot_params = params_plots())
```

Arguments

plot	The plot you want to save to disk.
file_name	Optional String. If not provided, the plot will be named after the R variable.
path	Directory. Where to save the plot to.
plot_params	List. Output of params_plots() . A list with the following elements: • width - Width of the plot. • height - Height of the plot. • file_type - Which file type. • unit - Which unit do width and height describe. • res - Resolution for PNG plots

Value

Saves the plot to disk and returns invisible.

save_plot_ls	<i>Save a list of plots</i>
--------------	-----------------------------

Description

Save a list of plots

Usage

```
save_plot_ls(plot_ls, path, plot_params = params_plots())
```

Arguments

plot_ls	Named list of plots. The names will serve as file names (the extension will be added by the function)
path	Directory. Where to save the plot to.
plot_params	List. Output of <code>params_plots()</code> . A list with the following elements: • width - Width of the plot. • height - Height of the plot. • file_type - Which file type. • unit - Which unit do width and height describe. • res - Resolution for PNG plots

Value

Saves the plots to disk and returns invisible.

scale_color_bx	<i>Bixverse Color Scale (Discrete)</i>
----------------	--

Description

Bixverse Color Scale (Discrete)

Usage

```
scale_color_bx(palette = "main", reverse = FALSE, ...)
```

Arguments

palette	Palette name (default: "main")
reverse	Reverse colors? (default: FALSE)
...	Additional arguments passed to scale_color_manual

Value

A ggplot2 scale object

scale_color_bx_c	<i>Bixverse Color Scale (Continuous)</i>
------------------	--

Description

Bixverse Color Scale (Continuous)

Usage

```
scale_color_bx_c(palette = "sequential", reverse = FALSE, n = 20, ...)
```

Arguments

palette	Palette name (default: "sequential")
reverse	Reverse colors? (default: FALSE)
n	Integer. Number of colours to interpolate over (default: 20), see bx_colors() .
...	Additional arguments passed to <code>scale_color_gradientn</code>

Value

A ggplot2 scale object

scale_fill_bx	<i>Bixverse Fill Scale (Discrete)</i>
---------------	---------------------------------------

Description

Bixverse Fill Scale (Discrete)

Usage

```
scale_fill_bx(palette = "main", reverse = FALSE, ...)
```

Arguments

palette	Palette name (default: "main")
reverse	Reverse colors? (default: FALSE)
...	Additional arguments passed to <code>scale_fill_manual</code>

Value

A ggplot2 scale object

scale_fill_bx_c	<i>Bixverse Fill Scale (Continuous)</i>
-----------------	---

Description

Bixverse Fill Scale (Continuous)

Usage

```
scale_fill_bx_c(palette = "sequential", reverse = FALSE, n = 20, ...)
```

Arguments

palette	Palette name (default: "sequential")
reverse	Reverse colors? (default: FALSE)
n	Integer. Number of colours to interpolate over (default: 20), see bx_colors() .
...	Additional arguments passed to <code>scale_fill_gradientn</code>

Value

A ggplot2 scale object

stacked_violin_plot_sc	<i>Stacked violin plot of gene expression across groups</i>
------------------------	---

Description

Stacked violin plot of gene expression across groups

Usage

```
stacked_violin_plot_sc(  
  object,  
  features,  
  grouping_variable,  
  feature_labels = NULL,  
  scale = FALSE,  
  clip = NULL,  
  modality = c("rna", "adt"),  
  palette = BX_PALETTES  
)
```

Arguments

object	A single cell class.
features	Character vector. Gene IDs to plot, one row each.
grouping_variable	String. Obs column to group by.
feature_labels	Optional named character vector mapping gene ids to display labels (default: NULL).
scale	Boolean. Whether to z-score the expression values.
clip	Optional numeric. Clip z-scores if scale = TRUE.
modality	String. One of c("rna", "adt").
palette	String. Discrete palette for the group fills. One of c("main", "sequential", "diverging", "viridis", "spectral"), see bx_colors() .

Value

A [ggplot](#) object.

theme_bx	<i>Bixverse ggplot2 Theme</i>
----------	-------------------------------

Description

A custom theme for the bixverse ecosystem based on the [ggplot2::theme_minimal\(\)](#)

Usage

```
theme_bx(
  base_size = 12,
  base_family = "Helvetica",
  base_line_size = 0.5,
  base_rect_size = 0.5
)
```

Arguments

base_size	Base font size (default: 12)
base_family	Base font family (default: "Helvetica")
base_line_size	Base line size (default: 0.5)
base_rect_size	Base rectangle size (default: 0.5)

Value

A [ggplot2](#) theme object

Examples

```
## Not run:
library(ggplot2)
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point() +
  theme_bx()

## End(Not run)
```

violin_plot_sc.CellQc *Per-metric violin plots from a CellQc object*

Description

Per-metric violin plots from a CellQc object

Usage

```
## S3 method for class 'CellQc'
violin_plot_sc(
  x,
  log_scale = FALSE,
  show_outlier = TRUE,
  raster = NULL,
  palette = BX_PALETTES,
  ...
)
```

Arguments

x	A CellQc object.
log_scale	Logical. Apply a log10 y-axis (default: FALSE).
show_outlier	Logical. Overlay outlier points (default: TRUE).
raster	Optional boolean. Shall the plot be rasterised. If NULL and number of cells is larger than 1e5, defaults to TRUE.
palette	String. Discrete palette for the group colours. One of c("main", "sequential", "diverging", "viridis", "spectral"), see bx_colors() . Ignored when show_outlier = TRUE.
...	Ignored.

Value

A named list of ggplot objects, one per metric.

violin_plot_sc.data.table

Violin plot from a data.table

Description

Per-cell outliers are recomputed within each grouping_column group.

Usage

```
## S3 method for class 'data.table'
violin_plot_sc(
  x,
  grouping_column,
  variable,
  direction = c("twosided", "below", "above"),
  threshold = 3,
  group_name = NULL,
  var_name = NULL,
  log_scale = TRUE,
  show_outlier = TRUE,
  raster = NULL,
  palette = BX_PALETTES,
  ...
)
```

Arguments

x	data.table. Input data containing the QC metric.
grouping_column	Character. Column used for the x-axis groups.
variable	Character. Numeric column to plot on the y-axis.
direction	Character. One of "twosided", "below", "above".
threshold	Numeric. Number of MADs for outlier detection (default: 3).
group_name	Character. x-axis label (default: NULL).
var_name	Character. y-axis label (default: NULL).
log_scale	Logical. Apply a log10 y-axis (default: TRUE).
show_outlier	Logical. Overlay outlier points (default: TRUE).
raster	Optional boolean. Shall the plot be rasterised. If NULL and number of cells is larger than 1e5, defaults to TRUE.
palette	String. Discrete palette for the group colours. One of c("main", "sequential", "diverging", "viridis", "spectral"), see bx_colors() . Ignored when show_outlier = TRUE.
...	Ignored.

Value

A `ggplot` object.

volcano_plot	<i>Create a volcano plot</i>
--------------	------------------------------

Description

Helper function to generate volcano plots for differential gene/transcript/protein expression analysis. The function applies under the hood a $-\log_{10}(\text{stat})$ transformation to the respective statistic you want to display on the y axis. If you provide labels, the top features to label will be chosen by ranking the $-\log_{10}(\text{stat})$.

Usage

```
volcano_plot(
  dt,
  volcano_plot_params = params_volcano(),
  x_lab = NULL,
  y_lab = NULL,
  plot_title = NULL,
  plot_sub_title = NULL
)
```

Arguments

<code>dt</code>	A data.table/data.frame holding the differential expression results.
<code>volcano_plot_params</code>	A list as returned by <code>params_volcano()</code> .
<code>x_lab</code>	Optional string. Overwrite of the x label of the plot. If not provided, will default to <code>volcano_plot_params\$x_axis</code> .
<code>y_lab</code>	Optional string. Overwrite of the x label of the plot. If not provided, will default to $-\log_{10}(\text{volcano_plot_params}\$y_axis)$.
<code>plot_title</code>	Optional string. If provided, the plot will get a title.
<code>plot_sub_title</code>	Optional string. If provided, adds this as a sub title to the plot.

Value

A `ggplot` object.

wrap_and_truncate	<i>Helper function to wrap and truncate text</i>
-------------------	--

Description

Helper function to wrap and truncate text

Usage

```
wrap_and_truncate(text, width = 40L, max_lines = 2L, whitespace_only = TRUE)
```

Arguments

text	String. The string to truncate.
width	Integer. Maximum width of a given line.
max_lines	Integer. Maximum lines before truncating.
whitespace_only	Boolean. Shall the string wrapping happen only around whitespaces.

Value

The string wrapped and/or truncated.

Index

bixverse::add_nhoods_info(), 17
bixverse::blitzgsea_calibrate(), 24
bixverse::calc_blitzgsea(), 7, 24
bixverse::calc_fgsea(), 7
bixverse::calc_mitch(), 28
bixverse::extract_binary_heatmap_data(), 22
bixverse::extract_embedding_data(), 6
bixverse::extract_feature_plot_data(), 10
bixverse::gse_hypergeometric(), 26
bixverse::gse_hypergeometric_list(), 26, 27
bixverse::params_binary_heatmap(), 22
bixverse::run_paga_sc(), 18
bixverse::test_nhoods(), 17
bx_colors(), 3–6, 9, 11, 13–15, 17, 19, 23, 31–35

density_plot_sc.CellQc, 2
density_plot_sc.data.table, 3
dot_plot_sc, 4

embedding_plot_sc, 5
enrichment_map_gsea, 6
enrichment_map_gsea(), 25, 26
enrichment_map_oae, 7
enrichment_map_oae(), 25, 26

feature_plot_sc, 8
feature_scatter_plot_sc, 10

geom_label, 16
get_embedding(), 10
get_gsea_enrichment_data, 11
ggplot, 4–6, 10, 11, 13, 17, 19, 23, 33, 36
ggplot2::scale_fill_viridis_c(), 27
ggplot2::theme_minimal(), 33
ggraph::geom_node_text(), 25
heatmap_plot_sc, 12

igraph::add_layout_(), 7, 8

joint_plot_sc.CellQc, 13
joint_plot_sc.data.table, 14

label_centroids, 15

milo_nhood_plot_sc, 16

paga_plot_sc, 18
params_plots, 20
params_plots(), 29, 30
params_volcano, 21
params_volcano(), 36
patchwork, 10
patchwork::wrap_plots(), 9
plot_binary_heatmap, 22
plot_blitzgsea_es_null, 23
plot_blitzgsea_null, 24
plot_enrichment_map_ggraph, 25
plot_enrichment_map_visnetwork, 26
plot_gse_dotplot, 26
plot_gsea_enrichment, 27
prepare_mitch_scores, 28

save_plot, 29
save_plot_ls, 29
scale_color_bx, 30
scale_color_bx_c, 31
scale_fill_bx, 31
scale_fill_bx_c, 32
stacked_violin_plot_sc, 32

theme_bx, 33

violin_plot_sc.CellQc, 34
violin_plot_sc.data.table, 35
volcano_plot, 36

wrap_and_truncate, 37
wrap_and_truncate(), 7, 8, 27