Package: omixVizR (via r-universe)

April 12, 2026

Title A Toolkit for Omics Data Visualization

Version 1.4.5

Description Provides a suite of tools for the comprehensive visualization of multi-omics data, including genomics, transcriptomics, and proteomics. Offers user-friendly functions to generate publication-quality plots, thereby facilitating the exploration and interpretation of complex biological datasets. Supports seamless integration with popular R visualization frameworks and is well-suited for both exploratory data analysis and the presentation of final results. Key formats and methods are presented in Huang, S., et al. (2024) ``The Born in Guangzhou Cohort Study enables generational genetic discoveries'' <doi:10.1038/s41586-023-06988-4>.

License MIT + file LICENSE

URL `https://github.com/Leslie-Lu/omixVizR`

BugReports `https://github.com/Leslie-Lu/omixVizR/issues`

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0), spelling

Config/testthat/edition 3

Imports corpcor, data.table, dplyr, genpwr, ggbreak, ggplot2, ggrepel, ggsci, ggtext, grid, gridExtra, gtable, lulab.utils, magrittr, Matrix, patchwork, pheatmap, purrr, rlang, scales, showtext, stringr, sysfonts, systemfonts, tibble

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev make libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev libssl-dev zlib1g-dev

Repository https://leslie-lu.r-universe.dev

Date/Publication 2025-11-13 11:19:59 UTC

RemoteUrl https://github.com/Leslie-Lu/omixVizR

RemoteRef HEAD

RemoteSha 92bf95f9a1513962851518fde48f5107325263ee

Contents

`MR_2SLS` *MR_2SLS*

Description

Perform Two-Stage Least Squares (2SLS) Mendelian Randomization analysis.

Usage

```
MR_2SLS(
  infile,
  proteins_data,
  proteins_measured,
  outcome,
  outcome_name,
  prs_cols_match,
  regexpr_pattern,
  standardise = TRUE,
  digits = 3,
  .progress = TRUE
)
```

Arguments

`infile`	The input file containing the data.
`proteins_data`	The file containing the proteins data actually measured for the samples.
`proteins_measured`	The name vector of proteins actually measured for the samples.
`outcome`	The outcome variable.
`outcome_name`	The name of the outcome variable.
`prs_cols_match`	The columns to match for PRS.

`regexpr_pattern`	The regular expression pattern to use for matching.
`standardise`	Whether to standardise the prs data, Default: TRUE
`digits`	The number of decimal places to round to, Default: 3
`.progress`	Whether to show progress, Default: TRUE

Details

This function performs a Two-Stage Least Squares (2SLS) Mendelian Randomization analysis.

Value

A list containing the results of the 2SLS analysis.

Author(s)

Zhen Lu luzh29@mail2.sysu.edu.cn

Siyang Liu liusy99@mail.sysu.edu.cn

References

Chen, Y., Lu, T., Pettersson-Kymmer, U., Stewart, I. D., Butler-Laporte, G., Nakanishi, T., Cerani, A., Liang, K. Y. H., Yoshiji, S., Willett, J. D. S., Su, C.-Y., Raina, P., Greenwood, C. M. T., Farjoun, Y., Forgetta, V., Langenberg, C., Zhou, S., Ohlsson, C., & Richards, J. B. (2023). Genomic atlas of the plasma metabolome prioritizes metabolites implicated in human diseases. Nature Genetics, 55(1), 44–53. https://doi.org/10.1038/s41588-022-01270-1

See Also

`fread`, `as.data.table`, `setattr`, `setDT` `select`, `reexports` `map` `str_extract` `nearPD` `cov.shrink`

`plot_forest` *plot_forest*

Description

Creates a forest plot overlaid on a table.

Usage

```
plot_forest(
  p_left_data,
  point_estimate,
  ci_lower_bound,
  ci_upper_bound,
  ci_sep = ", ",
  p_right_data = NULL,
```

```
  precision_digits = 3,
  p_mid_width = 30,
  null_line_at = 1,
  output_path = NULL,
  dpi = 600,
  display = TRUE,
  font_family = c("MetroSans", "mono"),
  p_left_data_name = NULL,
  p_right_data_name = "Effect size (95% CI)",
  stripe_colour = "#eff3f2",
  background_colour = "white",
  x_scale_linear = TRUE,
  xlim = NULL,
  xbreaks = NULL,
  nudge_y = 0,
  nudge_x = 1,
  nudge_height = 0,
  nudge_width = 0,
  justify = 0,
  arrows = FALSE,
  arrow_labels = c("Lower", "Higher"),
  risk_colors = (ggsci::pal_npg("nrc"))(2),
  arrow_nudge_y = 0,
  add_plot = NULL,
  add_plot_width = 1,
  add_plot_gap = FALSE,
  point_sizes = 3,
  point_shapes = 16,
  p_mid_forest = NULL,
  lower_header_row = FALSE,
  render_as = "png",
  table_theme = NULL
)
```

Arguments

`p_left_data`	Data frame (required). The information to be displayed to the left of the forest plot.
`point_estimate`	Vector. The point estimates to be displayed in the forest plot.
`ci_lower_bound`	Vector. The lower confidence bounds.
`ci_upper_bound`	Vector. The upper confidence bounds.
`ci_sep`	String. What should separate the low and high confidence bounds? Default " to ".
`p_right_data`	Data frame (optional). Information to be displayed on the right side of the table. If not supplied, an Estimate column is generated automatically.
`precision_digits`	Integer. The number of decimal places on the point_estimate (default 3)

`p_mid_width`	Integer. The width of forest plot in characters (default 30)
`null_line_at`	Numeric. Default 1. Change to 0 if using absolute measures.
`output_path`	String. Where to save the image, default tempdir().
`dpi`	Numeric. The image resolution in dpi, default 600
`display`	Logical. Should the file be opened? Default TRUE.
`font_family`	String or character vector. The font to use for the ggplot and table. Default c("MetroSans", "mono"). The first available font is used.
`p_left_data_name`	String or String Vector. The name vector for the left side data. Default NULL, which uses the column names of p_left_data.
`p_right_data_name`	String. The name vector for the right side data. Default "Effect size (95% CI)"
`stripe_colour`	Hex String. Colour to use for the table stripes, default "#eff3f2".
`background_colour`	Hex String or Colour Name. The colour of the background, default "white".
`x_scale_linear`	Logical. Default TRUE, change to FALSE for a log scale.
`xlim`	Vector. Manually specify limits for the x axis as a vector length 2, i.e. c(low, high)
`xbreaks`	Vector. X axis breaks to label. Specify limits in xlim if using this option.
`nudge_y`	Numeric. Allows small changes to the vertical alignment of the forest plot points. 1 unit is approximately the height of 1 row.
`nudge_x`	Numeric. Nudge the alignment horizontally. Default 1. Higher values make the entire plot wider and consequently space out the elements of the figure.
`nudge_height`	Numeric. Adjust the overall height of the plot output. Default is 0.
`nudge_width`	Numeric. Adjust the overall width of the plot output. Default is 0.
`justify`	Numeric Vector. This should be a numeric vector either of length 1 (in which case it will apply to all columns) or of length equal to the number of columns in p_left_data + 1 (for the point_estimate column). Each number in the vector dictates the column justification, with 0 being left, 0.5 being center, and 1 being right.
`arrows`	Logical. Should there be arrows displayed below the ggplot? Default FALSE. Specify xlim if using arrows.
`arrow_labels`	String Vector, length 2. Labels for the arrows. Set arrows to TRUE or this will have no effect.
`risk_colors`	Vector. Length 2. Colors for the effect measure. Default is ggsci::pal_npg("nrc")(2).
`arrow_nudge_y`	Numeric. Nudge the vertical position of the arrows. Default 0.
`add_plot`	A ggplot object to add to the right side of the table. To align correctly with rows, 1 unit is the height of a row and y = 0 for the center of the bottom row.
`add_plot_width`	Numeric. Width to display add_plot. Relative to the width of the forest plot, where 1 (the default) is the same width.
`add_plot_gap`	Logical. Should there be space added between the plot and the main figure? Default FALSE.

`point_sizes`	Vector. Length should be equal to 1 or nrow(p_left_data). The sizes of the points in the center plot, where 3.25 is the default.
`point_shapes`	Vector. Length should be equal to 1 or nrow(p_left_data). The shapes of the points in the center plot, where 16 (a filled circle) is the default.
`p_mid_forest`	A ggplot object to use instead of the central plot.
`lower_header_row`	Logical. If TRUE, drops the header down one row (In the table rather than above it, like the default value (FALSE))
`render_as`	String or Function. What output format should be used? Option is passed to ggplot2::ggsave() as the argument "device". Either pass a device function (e.g. png) or one of "eps", "ps", "tex" (pictex), "pdf", "jpeg", "tiff", "png", "bmp", "svg" or "wmf" (windows only).
`table_theme`	A gridExtra table theme. If specified, overwrites all table theme customization in other options. The default is a modified version of ttheme_minimal.

Details

This function creates a forest plot overlaid on a table. It is highly customizable, allowing for adjustments to fonts, colors, and layout.

Value

image

Author(s)

Zhen Lu luzh29@mail2.sysu.edu.cn

`plot_gwas_power` *plot_gwas_power*

Description

Generates a statistical power analysis plot for GWAS studies. Supports binary (case-control) traits over a range of odds ratios and minor allele frequencies, and quantitative traits over a range of effect sizes and minor allele frequencies. This function uses the ’genpwr’ package for calculations and creates a highly customized ggplot.

Usage

```
plot_gwas_power(
  trait_type = "bt",
  n_cases = NULL,
  n_controls = NULL,
  sd_trait = NULL,
  N = NULL,
```

```
  maf_levels = c(0.01, 0.02, 0.05, 0.1, 0.2, 0.5),
  or_range = seq(1.01, 2, 0.001),
  effect_size = seq(0.01, 0.3, 0.001),
  alpha = 5e-08,
  plot_title = NULL,
  save_plot = TRUE,
  output_graphics = "png",
  width = 17,
  height = 9,
  dpi = 600
)
```

Arguments

`trait_type`	Character string specifying trait type: "bt" for binary (case-control) or "qt" for quantitative traits. Default: `"bt"`.
`n_cases`	Number of cases in the study (required if `trait_type = "bt"`).
`n_controls`	Number of controls in the study (required if `trait_type = "bt"`).
`sd_trait`	Numeric, standard deviation of the quantitative trait (required if `trait_type = "qt"`).
`N`	Numeric, total sample size for quantitative traits (required if `trait_type = "qt"`).
`maf_levels`	A numeric vector of Minor Allele Frequencies (MAFs) to test. Default: `c(0.01, 0.02, 0.05, 0.10, 0.20, 0.50)`.
`or_range`	A numeric vector specifying the sequence of Odds Ratios (ORs) to test. Default: `seq(1.01, 2.00, 0.001)`. Used when `trait_type = "bt"`.
`effect_size`	A numeric vector specifying the sequence of effect sizes (beta) to test for quantitative traits. Default: `seq(0.01, 0.30, 0.001)`. Used when `trait_type = "qt"`.
`alpha`	The significance level (alpha) for the power calculation. Default: `5e-8`.
`plot_title`	A string for the plot title. Default: `Cases / Controls` for binary traits or `Sample Size` for quantitative traits.
`save_plot`	Logical, whether to save the plot to a file. If `FALSE`, the plot object is only returned. Default: `TRUE`.
`output_graphics`	The file format for saving the plot. Currently supports "png" and "pdf". Default: "png".
`width`	The width of the saved plot in inches. Default: 17.
`height`	The height of the saved plot in inches. Default: 9.
`dpi`	The resolution of the saved plot in dots per inch. Default: 600.

Details

This function automates the process of calculating and visualizing GWAS power for both binary (case-control) and quantitative traits. For binary traits, it analyzes power across odds ratios, while for quantitative traits, it analyzes power across effect sizes. It highlights the minimum OR/effect size required to achieve 80% power for the lowest and third-lowest MAF levels, adding dashed lines and color-coded labels for clarity.

Value

A list containing two elements:

`plot`	The ggplot object for the power plot.
`power_data`	A data.table containing the full results from the power analysis.

Font Information

The MetroSans font included in this package is intended for academic research and non-commercial use only. For commercial use, please contact the font copyright holder.

The font files are included in the package's inst/extdata directory and are automatically loaded for plotting.

Author(s)

Zhen Lu luzh29@mail2.sysu.edu.cn

Examples

```
# Binary trait example (case-control)
power_results_bt <- plot_gwas_power(
  trait_type = "bt",
  n_cases = 4324,
  n_controls = 93945,
  save_plot = FALSE
)

# Quantitative trait example
power_results_qt <- plot_gwas_power(
  trait_type = "qt",
  sd_trait = 0.09365788681305078,
  N = 10000,
  maf_levels = c(0.01, 0.02, 0.05, 0.10, 0.20, 0.50),
  effect_size = seq(0.01, 0.10, 0.001),
  save_plot = FALSE
)

# Access the ggplot object and data
# print(power_results_bt$plot)
# print(power_results_bt$power_data)
```

`plot_heatmap` *plot_heatmap*

Description

Generate a heatmap for a given correlation matrix.

Usage

```
plot_heatmap(
  cor_mat,
  scale = ”none”,
  show_rownames = FALSE,
  show_colnames = FALSE,
  labels_row = NULL,
  labels_col = NULL,
  breaks = seq(-1, 1, length.out = 101),
  legend_breaks = c(-1, -0.5, 0, 0.5, 1),
  legend_labels = c(”-1.0”, ”-0.5”, ”0”, ”0.5”, ”1.0”),
  legend_name = ”Correlation”,
  output_path = NULL,
  display = TRUE,
  render_as = ”png”,
  width = 18,
  height = 16,
  dpi = 600
)
```

Arguments

`cor_mat`	A numeric matrix representing the correlation values.
`scale`	Character string indicating how to scale the data. Default is ’none’.
`show_rownames`	Logical indicating whether to show row names. Default is FALSE.
`show_colnames`	Logical indicating whether to show column names. Default is FALSE.
`labels_row`	Character vector of labels for the rows. Default is NULL.
`labels_col`	Character vector of labels for the columns. Default is NULL.
`breaks`	Numeric vector of breakpoints for the heatmap colors. Default is seq(-1, 1, length.out = 101).
`legend_breaks`	Numeric vector of breakpoints for the legend. Default is c(-1, -0.5, 0, 0.5, 1).
`legend_labels`	Character vector of labels for the legend. Default is c("-1.0", "-0.5", "0", "0.5", "1.0").
`legend_name`	Character string for the legend title. Default is ’Correlation’.
`output_path`	Character string for the output file path. Default is NULL.
`display`	Logical indicating whether to display the plot. Default is TRUE.
`render_as`	Character string indicating the file format to save the plot. Default is ’png’.
`width`	Numeric indicating the width of the plot. Default is 18.
`height`	Numeric indicating the height of the plot. Default is 16.
`dpi`	Numeric indicating the resolution of the plot. Default is 600.

Details

This function generates a heatmap for a given correlation matrix.

Value

A heatmap plot.

Author(s)

Zhen Lu luzh29@mail2.sysu.edu.cn

`plot_qqman` *plot_qqman*

Description

Create GWAS QQ & Manhattan Plots.

Usage

```
plot_qqman(
  plink_assoc_file,
  pheno_name,
  maf_filter = NULL,
  gwas_threshold = 5e-08,
  label_col = "SNP",
  output_graphics = "png",
  save_plot = TRUE,
  lambda1_qq_pos = c(2.1, -5.5),
  lambda2_qq_pos = c(1.565, -4)
)
```

Arguments

`plink_assoc_file`	Path to the PLINK association file.
`pheno_name`	Phenotype name.
`maf_filter`	Minor allele frequency filter, Default: NULL
`gwas_threshold`	The significance threshold for highlighting SNPs. Default: `5e-8`.
`label_col`	The name of the column to use for labeling significant SNPs. Default: `"SNP"`.
`output_graphics`	Output graphics format, Default: 'png'
`save_plot`	Logical, whether to save plots to files. If FALSE, plots are only displayed. Default: TRUE
`lambda1_qq_pos`	A numeric vector of length 2 specifying the `c(hjust, vjust)` for the lambda text in the QQ plot. Default: `c(2.1, -5.5)`.
`lambda2_qq_pos`	A numeric vector of length 2 specifying the `c(hjust, vjust)` for the SNP count (N) text in the QQ plot. Default: `c(1.565, -4.0)`.

Details

This function reads a PLINK association file and generates Manhattan and QQ plots for the GWAS results.

Value

A list containing the ggplot objects for the Manhattan and QQ plots.

Font Information

The MetroSans font included in this package is intended for academic research and non-commercial use only. For commercial use, please contact the font copyright holder.

The font files are included in the package's inst/extdata directory and are automatically loaded for plotting.

Author(s)

Zhen Lu luzh29@mail2.sysu.edu.cn

Yanhong Liu liuyh275@mail2.sysu.edu.cn

Siyang Liu liusy99@mail.sysu.edu.cn

Examples

```
sample_file <- system.file("extdata", "sample_gwas.assoc.linear", package = "omixVizR")

# Check if the file exists before running the example
if (file.exists(sample_file)) {
  # Run the function with the sample data
  plots <- plot_qqman(
    plink_assoc_file = sample_file,
    pheno_name = "SamplePheno",
    save_plot = FALSE
  )
  # You can then access the plots like this:
  # print(plots$manhattan_plot)
  # print(plots$qq_plot)
} else {
  message("Sample file not found, skipping example.")
}
```

Index