Package 'ggvolc'

November 1, 2026

Type Package

Title Create Volcano Plots for Differential Gene Expression Data

Version 0.4.0

Description Creates customizable volcano plots for differential gene expression results from RNA-seq analyses. Accepts result objects or tables from 'DESeq2', 'edgeR', and 'limma' directly, with automatic column detection. Labels the most significant or chosen genes, colors gene sets, compares several contrasts side by side, offers colorblind-safe color presets, and can produce interactive plots with 'ggiraph'. Gene tables are rendered with 'gt' and composed with 'patchwork'.

License MIT + file LICENSE

URL https://loukesio.github.io/ggvolc/, https://github.com/loukesio/ggvolc

BugReports https://github.com/loukesio/ggvolc/issues

Encoding UTF-8

LazyData true

Imports dplyr, ggplot2, ggrepel, ggtext, gt, patchwork, grid, grDevices, methods, stats

RoxygenNote 7.3.3

Suggests testthat (>= 3.0.0), ggiraph, DESeq2, edgeR, limma, knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

Config/Needs/website pkgdown

Depends R (>= 4.1.0)

NeedsCompilation no

Author Loukas Theodosiou [aut, cre] (ORCID: <https://orcid.org/0000-0001-6418-4652>)

Maintainer Loukas Theodosiou <theodosiou@evolbio.mpg.de>

Repository CRAN

Date/Publication 2026-11-01 08:20:02 UTC

Contents

all_genes 2
attention_genes 3
de_table 3
edger_genes 4
genes_table 5
ggvolc 7
ggvolc_palettes 11
stress_timecourse 12

Index **13**

`all_genes` *Gene Expression Analysis Dataset*

Description

A dataset containing gene expression analysis results. It has seven columns capturing various statistics related to gene expression.

Usage

```
all_genes
```

Format

A data frame with the following columns:

genes Character. Gene name or identifier.

baseMean Numeric. The base mean value for the gene across samples.

log2FoldChange Numeric. The log2 fold change of gene expression. Positive values indicate upregulation and negative values indicate downregulation.

lfcSE Numeric. Standard error of the log2 fold change.

stat Numeric. The Wald statistic for the gene's expression change.

pvalue Numeric. Raw p-value for the test of the gene's expression change.

padj Numeric. Adjusted p-value for multiple testing corrections.

attention_genes *Attention Genes Dataset*

Description

A dataset containing specific genes of interest referred to as "attention genes".

Usage

```
attention_genes
```

Format

A data frame with 10 rows and 7 variables:

genes Character. Gene name or identifier.

baseMean Numeric vector: Base mean expression level of genes

log2FoldChange Numeric vector: Log2 Fold Change of gene expression

lfcSE Numeric vector: Standard error for log2 fold change

stat Numeric vector: Wald statistic for the gene's expression change

pvalue Numeric vector: Raw p-value from Wald test

padj Numeric vector: Adjusted p-value for multiple testing using the Benjamini-Hochberg procedure

de_table *Turn a DE result object into a plain data frame*

Description

Converts the result objects returned by the three major differential expression pipelines into a plain `data.frame` that `ggvolc` and `genes_table` understand. You rarely need to call this yourself: both functions call it internally, so you can pass the objects straight in. It is exported for when you want to inspect or filter the table first.

Usage

```
de_table(x, coef = NULL)
```

Arguments

`x`	A DE result object or a data frame (see Details).
`coef`	For a multi-coefficient limma fit, the coefficient (name or index) to extract. For a `DESeqDataSet`, passed as `name` to `DESeq2::results()`. Ignored otherwise.

Details

Supported inputs:

DESeq2 A `DESeqResults` object from `DESeq2::results()` or `DESeq2::lfcShrink()`, or a `DESeqDataSet` after `DESeq2::DESeq()`, in which case `DESeq2::results()` is called for you (with `name = coef` when `coef` is given).

edgeR A `TopTags` object from `edgeR::topTags()`, or a `DGELRT` / `DGEExact` object from `edgeR::glmQLFTest()`, `edgeR::glmLRT()` or `edgeR::exactTest()`. All genes are kept (`edgeR::topTags(x, n = Inf)`).

limma An `MArrayLM` fit after `limma::eBayes()`. `limma::topTable(fit, coef = coef, number = Inf)` is called for you. If the fit has more than one coefficient, `coef` is required.

data frame Any `data.frame` (including tibbles) is returned unchanged.

Value

A `data.frame` with one row per gene. Gene identifiers are kept in the row names (or a `genes` column, if the input had one).

Examples

```
# A data frame passes straight through
head(de_table(all_genes))

## Not run:
# DESeq2
dds <- DESeq2::DESeq(dds)
ggvolc(DESeq2::results(dds))       # results object
ggvolc(dds)                        # or the dataset itself

# edgeR
fit <- edgeR::glmQLFit(y, design)
ggvolc(edgeR::glmQLFTest(fit, coef = 2))

# limma
fit <- limma::eBayes(limma::lmFit(v, design))
ggvolc(fit, coef = "treatment")

## End(Not run)
```

`edger_genes` *edgeR-style Differential Expression Dataset*

Description

The same experiment as `all_genes`, re-expressed with the column conventions of **edgeR**'s `topTags()` output. The gene identifiers are stored in the row names (as `topTags()` returns them), which lets it demonstrate that `ggvolc` accepts edgeR results directly and promotes rowname gene IDs to a `genes` column automatically.

Usage

```
edger_genes
```

Format

A data frame with gene identifiers in the row names and the following columns:

logFC Numeric. Log2 fold change of gene expression.

logCPM Numeric. Average log2 counts per million.

PValue Numeric. Raw p-value from the differential expression test.

FDR Numeric. Benjamini-Hochberg adjusted p-value (false discovery rate).

---

genes_table *Combine a Volcano Plot with a gt Table of Genes*

Description

Takes a ggplot object (typically from `ggvolc`) and a data frame of gene details and produces a combined layout using **patchwork**. The gene table is rendered as a polished **gt** table with color-coded significance and formatted numeric columns.

Usage

```
genes_table(
  plot_obj,
  data2,
  top_n = NULL,
  sig_col = c("padj", "pvalue"),
  dir = c("both", "up", "down", "each"),
  p_value = 0.05,
  table_height = NULL,
  palette = NULL,
  coef = NULL
)
```

Arguments

`plot_obj`	A ggplot object, typically the output of `ggvolc`.
`data2`	Gene details: a data frame, or a DESeq2 / edgeR / limma result object (see `de_table`). Columns are auto-detected in the same way as `ggvolc` (DESeq2, edgeR, or limma conventions). Required columns after mapping: `genes`, `log2FoldChange`, `pvalue`. Optional: `baseMean`, `padj`. When `top_n` is supplied you can pass the full DE table here (e.g. `all_genes`) and the most significant genes are selected automatically.

top_n	Optional integer. When supplied, the top_n most significant genes (smallest sig_col, among genes with sig_col < p_value) are selected from data2 for the table. When NULL (default) every row of data2 is shown, so passing a pre-filtered set such as attention_genes behaves as before.
sig_col	Column used to rank significance when top_n is set. Either "padj" (adjusted p-value / FDR, the default) or "pvalue". Falls back to "pvalue" when "padj" is the default but no adjusted-p column is present.
dir	Direction to draw the top_n genes from. One of "both" (top N over all significant genes, the default), "up" (top N upregulated), "down" (top N downregulated), or "each" (top N up *and* top N down, up to 2 * top_n rows). Ignored when top_n is NULL.
p_value	Significance threshold used for color-coding the p-value column, and for filtering eligible genes when top_n is set. Default is 0.05.
table_height	Optional relative height of the table panel (the plot panel is 3). By default (NULL) the table takes exactly the height its rows need and the plot fills the rest.
palette	Colors for the log2FC column. Default NULL takes the colors of plot_obj when it came from ggvolc; otherwise a preset name or named vector as in ggvolc.
coef	Coefficient to extract when data2 is a multi-coefficient limma fit (see de_table).

Value

A patchwork object that can be further composed or saved with ggplot2::ggsave().

Examples

```
# Load example datasets
data(all_genes)
data(attention_genes)

# Create a volcano plot and combine it with a gt gene table
p <- ggvolc(all_genes, attention_genes, add_seg = TRUE)
genes_table(p, attention_genes)

# Auto-select the 10 most significant genes from the full table
p2 <- ggvolc(all_genes, label_top = 10)
genes_table(p2, all_genes, top_n = 10)

# Top 8 of each direction (up to 16 rows)
genes_table(p2, all_genes, top_n = 8, dir = "each")
```

ggvolc *Create a Volcano Plot*

Description

This function creates a volcano plot using ggplot2 based on provided datasets. It is particularly useful for visualizing differential gene expression data.

Usage

```
ggvolc(
  data1,
  data2 = NULL,
  size_var = NULL,
  p_value = 0.05,
  fc = 1,
  sig_col = c("padj", "pvalue"),
  label_top = NULL,
  label_dir = c("both", "up", "down", "each"),
  title = NULL,
  interactive = FALSE,
  palette = "default",
  not_sig_color = NULL,
  down_reg_color = NULL,
  up_reg_color = NULL,
  add_seg = FALSE,
  show_counts = FALSE,
  highlight = NULL,
  highlight_colors = NULL,
  highlight_alpha = 0.5,
  flip = FALSE,
  facet_scales = NULL,
  coef = NULL
)
```

Arguments

data1	The DE results to plot: a data frame, or a DESeq2 (DESeqResults, DESeqDataSet), edgeR (TopTags, DGELRT, DGEExact) or limma (MArrayLM) object. Gene identifiers can live in a genes column or in the row names. To compare contrasts, pass a *named list* of results, e.g. list("Day 1" = res1, "Day 7" = res7): each becomes a panel on shared axes, and label_top and show_counts work per panel.
data2	Genes to highlight and label: a data frame or DE object (same auto-detection as data1), or simply a character vector of gene names. With a list in data1, the genes are highlighted in every panel. Default is NULL.

`size_var` Variable for determining the size of points. Options are `"log2FoldChange"` and `"pvalue"`. Default is NULL (fixed size).

`p_value` Threshold for statistical significance. Default is 0.05.

`fc` Fold change threshold for determining upregulated or downregulated genes. Default is 1.

`sig_col` Column used to call significance and to build the y-axis. Either `"padj"` (the adjusted p-value / FDR, the default and the recommended cutoff for calling hits in most DE workflows) or `"pvalue"` (the raw p-value). The y-axis and the significance segment follow this choice, so the plot stays internally consistent. When the default `"padj"` is used but the data has no adjusted-p column, ggvolc falls back to the raw p-value.

`label_top` Optional integer. When supplied, the `label_top` most significant genes (smallest `sig_col`, among significant genes) are highlighted and labelled automatically, without building a separate `data2`. Combined with `data2` if both are given. Default NULL.

`label_dir` Direction to draw the `label_top` genes from. One of `"both"` (top N over all significant genes, the default), `"up"` (top N upregulated), `"down"` (top N downregulated), or `"each"` (top N upregulated *and* top N downregulated, i.e. up to 2N labels). Ignored when `label_top` is NULL.

`title` Plot title. Default NULL (no title).

`interactive` Logical. If TRUE, returns an interactive **ggiraph** `girafe` widget where hovering a point reveals the gene name and its statistics. Requires the optional **ggiraph** package. Default FALSE.

`palette` Colors for the plot: a preset name (see `ggvolc_palettes`), such as `"default"`, `"okabe_ito"` (colorblind-safe), or one of the ltc-derived presets, or a named vector such as `c(down = "navy", up = "firebrick")`.

`not_sig_color`, `down_reg_color`, `up_reg_color`

Colors for non-significant, downregulated, and upregulated genes. When given, they override the matching color from `palette`. Default NULL.

`add_seg` Logical. If TRUE, dashed lines will be added to the plot indicating the p-value and fold change thresholds. Default is FALSE.

`show_counts` Logical. If TRUE, the number of significantly down- and upregulated genes is printed in the top corners of the plot (of each panel when comparing contrasts). Default FALSE.

`highlight` Optional named list of gene sets to color, e.g. `list(Ribosome = c("RPL3", "RPS2"), Glycolysis = c("PFK1", "PYK1"))`. Genes in a set are drawn in the set's color, whatever their significance, and each set gets a legend entry. A gene in several sets takes the first.

`highlight_colors`

Optional colors for the `highlight` sets, in order or named by set. Defaults to the remaining colors of the ltc *minou* palette that the default up/down colors come from (mustard, navy, sage, ...), so they match the plot.

`highlight_alpha`

Transparency of the `highlight` points, from 0 (invisible) to 1 (solid): one value for all sets, or one per set (in order or named by set). Default 0.5, the same as all other points.

flip	Logical. If TRUE, swap the axes: log2 fold change on the y-axis and significance on the x-axis. Handy when comparing contrasts side by side, because the fold-change axis then lines up across the panels. Default FALSE.
facet_scales	When comparing contrasts, which axes each panel may set for itself: "fixed", "free_y", "free_x" or "free". The default (NULL) frees only the significance axis ("free_y", or "free_x" when flip = TRUE), so the fold-change axis stays shared and comparable while a weaker contrast is not squashed by a stronger one. Use "fixed" to share both axes.
coef	For a multi-coefficient limma fit, the coefficient to plot; for a DESeqDataSet, the result name. See de_table.

Details

You can pass the result objects of DESeq2, edgeR, and limma directly (see de_table), or a data frame. Column names from all three pipelines are detected and mapped automatically.

Value

A ggplot2 object displaying the volcano plot, or, when interactive = TRUE, a **ggiraph** girafe widget.

Examples

```
# Load example datasets included in the package
data(all_genes)
data(attention_genes)

# Create a basic volcano plot with default settings
# Points are colored by significance (p < 0.05, |log2FC| > 1)
ggvolc(all_genes)

# Highlight specific genes of interest with labels
# These genes are shown with black borders and gene names
ggvolc(all_genes, attention_genes)


# Add dashed lines to indicate significance thresholds
ggvolc(all_genes, attention_genes, add_seg = TRUE)

# Significance is called on the adjusted p-value (FDR) by default;
# use the raw p-value instead with:
ggvolc(all_genes, sig_col = "pvalue")

# Automatically label the 10 most significant genes
ggvolc(all_genes, label_top = 10)

# Label the top 10 upregulated genes only
ggvolc(all_genes, label_top = 10, label_dir = "up")

# Label the top 8 of each direction (up to 16 labels)
ggvolc(all_genes, label_top = 8, label_dir = "each")
```

```
# Give the plot a title
ggvolc(all_genes, title = "Treated vs. control")

# Show how many genes go up and down
ggvolc(all_genes, show_counts = TRUE)

# Color gene sets, e.g. a pathway
ggvolc(all_genes, highlight = list(Set_A = all_genes$genes[1:15],
                                   Set_B = all_genes$genes[100:130]))

# ... each set with its own transparency
ggvolc(all_genes, highlight = list(Set_A = all_genes$genes[1:15],
                                   Set_B = all_genes$genes[100:130]),
       highlight_alpha = c(Set_A = 1, Set_B = 0.5))

# Compare contrasts side by side: a real stress time course
ggvolc(stress_timecourse, label_top = 5, show_counts = TRUE)

# Share the significance axis too
ggvolc(stress_timecourse, label_top = 5, facet_scales = "fixed")

# Fold change on the y-axis, significance on the x-axis
ggvolc(stress_timecourse, label_top = 5, flip = TRUE)

# Colorblind-safe colors, or any preset from ggvolc_palettes()
ggvolc(all_genes, label_top = 10, palette = "okabe_ito")

# Customize colors for up- and down-regulated genes
ggvolc(all_genes, attention_genes,
       up_reg_color = "#E63946",
       down_reg_color = "#457B9D")

# Scale point size by p-value instead of default
ggvolc(all_genes, attention_genes, size_var = "pvalue")

# Adjust significance thresholds (p-value and fold change)
ggvolc(all_genes, p_value = 0.01, fc = 2)

# edgeR-style input works directly
edger_df <- data.frame(
  genes = paste0("gene", 1:50),
  logFC = rnorm(50, 0, 2),
  logCPM = runif(50, 5, 15),
  PValue = 10^-runif(50, 0, 6),
  FDR = 10^-runif(50, 0, 5)
)
ggvolc(edger_df)

# limma-style input works too
limma_df <- data.frame(
  genes = paste0("gene", 1:50),
  logFC = rnorm(50, 0, 2),
```

```
  AveExpr = runif(50, 5, 15),
  t = rnorm(50),
  P.Value = 10^-runif(50, 0, 6),
  adj.P.Val = 10^-runif(50, 0, 5)
)
ggvolc(limma_df)



## Not run:
# DE objects work directly, no conversion needed
ggvolc(DESeq2::results(dds))
ggvolc(edgeR::glmQLFTest(qlf_fit, coef = 2))
ggvolc(limma::eBayes(limma_fit), coef = "treatment")

# Interactive volcano (requires the ggiraph package): hover a point to
# see the gene name and its statistics
ggvolc(all_genes, attention_genes, interactive = TRUE)

## End(Not run)
```

`ggvolc_palettes` *Color presets for volcano plots*

Description

Lists the color presets available through the `palette` argument of `ggvolc` and `genes_table`. Each preset sets the color of downregulated, non-significant, and upregulated genes.

Usage

```
ggvolc_palettes()
```

Details

Presets marked `colorblind_safe` keep the up and down colors clearly apart under simulated deuteranopia, protanopia, and tritanopia (checked with **colorspace**). Several presets come from the ltc palettes; ltc does not need to be installed to use them.

You are not limited to these presets: `palette` also accepts a named color vector such as `c(down = "navy", up = "firebrick")`. That makes any palette usable, for example one from ltc: `pal <- ltc::ltc("maya"); ggvolc(all_genes, palette = c(down = pal[1], up = pal[4]))`.

Value

A data frame with one row per preset and columns `palette`, `down`, `not_sig`, `up`, `colorblind_safe`, and `source`.

Examples

```
ggvolc_palettes()

ggvolc(all_genes, palette = "okabe_ito")
ggvolc(all_genes, palette = c(down = "navy", up = "firebrick"))
```

`stress_timecourse` *Fission Yeast Stress Time Course (three contrasts)*

Description

Real DESeq2 results from an RNA-seq time course of wild-type fission yeast (*Schizosaccharomyces pombe*) exposed to osmotic stress (1 M sorbitol). Each element of the list compares one time point after stress with time 0, so the list can be passed straight to `ggvolc` to compare the contrasts side by side. The response is strongest at 30 minutes and fades by 180 minutes.

Usage

```
stress_timecourse
```

Format

A named list of three data frames, `"30 min vs 0 min"`, `"60 min vs 0 min"`, and `"180 min vs 0 min"`, each with 6535 rows and the columns:

genes Character. Gene symbol, or the systematic ID when no symbol exists.

baseMean Numeric. Mean of normalized counts across all samples.

log2FoldChange Numeric. log2 fold change versus time 0.

pvalue Numeric. Wald test p-value.

padj Numeric. Benjamini-Hochberg adjusted p-value.

Details

The counts come from the Bioconductor data package **fission** (wild-type samples only, 3 replicates per time point; genes with fewer than 10 reads in total removed). Results were computed with `DESeq2::DESeq()` using `design = ~ minute`; see `data-raw/stress_timecourse.R` in the package sources. As in any DESeq2 output, `padj` is `NA` for genes removed by independent filtering.

Source

Leong HS, Dawson K, Wirth C, et al. (2014). A global non-coding RNA system modulates fission yeast protein levels in response to stress. *Nature Communications* 5:3947. doi:10.1038/ncomms4947. GEO accession GSE56761. Data via the Bioconductor package **fission**.

Examples

```
names(stress_timecourse)
ggvolc(stress_timecourse, label_top = 5, show_counts = TRUE)
```

Index

∗ **datasets**
all_genes, 2
attention_genes, 3
edger_genes, 4
stress_timecourse, 12

all_genes, 2, *4*
attention_genes, 3

de_table, 3, *5, 6*, *9*

edger_genes, 4

genes_table, *3*, 5, *11*
ggvolc, *3–6*, 7, *11, 12*
ggvolc_palettes, *8*, 11

stress_timecourse, 12