Package ‘volcano3D’

October 12, 2022

Type Package

Title 3D Volcano Plots and Polar Plots for Three-Class Data

Version 2.0.8

Maintainer Katriona Goldmann <`k.goldmann@qmul.ac.uk`>

URL `https://katrionagoldmann.github.io/volcano3D/index.html`,
`https://github.com/KatrionaGoldmann/volcano3D`

BugReports `https://github.com/KatrionaGoldmann/volcano3D/issues`

Description Generates interactive plots for analysing and visualising three-class high dimensional data. It is particularly suited to visualising differences in continuous attributes such as gene/protein/biomarker expression levels between three groups. Differential gene/biomarker expression analysis between two classes is typically shown as a volcano plot. However, with three groups this type of visualisation is particularly difficult to interpret. This package generates 3D volcano plots and 3-way polar plots for easier interpretation of three-class data.

Language en-gb

License GPL-2

Encoding UTF-8

LazyData true

biocViews

Depends R (>= 3.5)

VignetteBuilder knitr

RoxygenNote 7.2.1

NeedsCompilation no

Imports plotly, ggplot2, ggpubr, htmlwidgets, magrittr, methods, Rfast, rlang, matrixTests

Suggests DESeq2, edgeR, limma, qvalue, SummarizedExperiment, knitr, rmarkdown, kableExtra, usethis, easylabel

Author Katriona Goldmann [aut, cre] (<`https://orcid.org/0000-0002-9073-6323`>),
Myles Lewis [aut] (<`https://orcid.org/0000-0001-9365-5345`>)

Repository CRAN
Date/Publication 2022-09-23 12:20:02 UTC

R topics documented:

add_animation 2
boxplot_trio 3
calc_pvals 5
calc_stats_2x3 6
deseq_2x3 7
deseq_2x3_polar 8
deseq_polar 10
forest_plot 11
grid-class 13
polar_coords 14
polar_coords_2x3 15
polar_grid 18
radial_ggplot 19
radial_plotly 21
show_grid 23
significance_subset 24
syn_example_meta 25
syn_example_rld 25
volc3d-class 26
volcano3D 26
voom_polar 28

Index **31**

`add_animation` *Add mode bar button to rotate the plot*

Description

Add mode bar button to rotate the plot

Usage

```
add_animation(
  p,
  rotate_icon_path = NULL,
  stop_icon_path = NULL,
  rotate_colour = "#c7c7c7",
  stop_colour = "#ff6347",
  scale = "scale(0.4) translate(-4, -4)",
  speed = 320,
  shiny_event_names = c()
)
```

Arguments

`p`	The `volcano3D` plot
`rotate_icon_path`	The svg icon path for rotation. If `NULL` a play button is used
`stop_icon_path`	The svg icon path for stop button. If `NULL` a pause button is used
`rotate_colour`	The colour for the rotate button (default="#c7c7c7")
`stop_colour`	The colour for the stop button (default='#ff6347', a.k.a 'tomato')
`scale`	Scaling for rotation button
`speed`	The rotation speed
`shiny_event_names`	If using shiny, pass in any shiny event names which should stop rotation when triggered (e.g. shiny_event_names = c('replot'))

Value

Returns a rotating cylindrical 3D plotly plot featuring variables on a tri-axis radial graph with the -log10(multi-group test p-value) on the z-axis

References

Lewis, Myles J., et al. (2019). Molecular portraits of early rheumatoid arthritis identify clinical and treatment response phenotypes. *Cell reports*, **28**:9

See Also

`volcano3D`

Examples

```
data(example_data)
syn_polar <- polar_coords(outcome = syn_example_meta$Pathotype,
                          data = t(syn_example_rld))

p <- volcano3D(syn_polar,
    label_rows = c("COBL", "TREX2"))
add_animation(p)
```

`boxplot_trio` *Boxplot to compare groups*

Description

Plots the expression of a specific row in expression to compare the three groups in a boxplot using either ggplot or plotly.

Usage

```
boxplot_trio(
  polar,
  value,
  box_colours = c("green3", "blue", "red"),
  test = "polar_pvalue",
  levels_order = NULL,
  my_comparisons = NULL,
  text_size = 10,
  stat_colour = "black",
  stat_size = 3,
  step_increase = 0.05,
  plot_method = "ggplot",
  ...
)
```

Arguments

`polar`	A 'volc3d' object including expression data from groups of interest. Created by `polar_coords`.
`value`	The column name or number in `polar@data` to be analysed
`box_colours`	The fill colours for each box assigned in order of levels_order. Default = c('green3', 'blue', 'red')).
`test`	The statistical test used to compare expression. Allowed values include: • `polar_pvalue` (default) and 'polar_padj' for the pvalues and adjusted pvalues in the polar object. • `polar_multi_pvalue` and `polar_multi_padj` for the pvalues and adjusted pvalues across all groups using the `polar@multi_group_test` columns. • `t.test` (parametric) and `wilcox.test` (non-parametric). Perform comparison between groups of samples. • `anova` (parametric) and `kruskal.test` (non-parametric). Perform one-way ANOVA test comparing multiple groups.
`levels_order`	A character vector stating the contrast groups to be plotted, in order. If `NULL` this defaults to the levels in `polar@outcome`.
`my_comparisons`	A list of contrasts to pass to `stat_compare_means`. If `NULL` (default) all contrast pvalues are calculated and plotted.
`text_size`	The font size of text (default = 10)
`stat_colour`	Colour to print statistics (default="black").
`stat_size`	The font size of statistical parameter (default = 3).
`step_increase`	The distance between statistics on the y-axis (default = 0.05).
`plot_method`	Whether to use 'plotly' or 'ggplot'. Default is 'ggplot'
`...`	Other parameters for `stat_compare_means`

Value

Returns a boxplot featuring the differential expression between groups in comparison with annotated pvalues.

References

Lewis, Myles J., et al. (2019). Molecular portraits of early rheumatoid arthritis identify clinical and treatment response phenotypes. *Cell reports*, **28**:9

Examples

```
data(example_data)
syn_polar <- polar_coords(outcome = syn_example_meta$Pathotype,
                          data = t(syn_example_rld))

boxplot_trio(syn_polar, value = "COBL", plot_method="plotly")
boxplot_trio(syn_polar, value = "COBL")
```

`calc_pvals` *Calculate one-way test and pairwise tests*

Description

Internal function for calculating 3-class group test (either one-way ANOVA or Kruskal-Wallis test) and pairwise tests (either t-test or Wilcoxon test) on multi-column data against an outcome parameter with 3 levels.

Usage

```
calc_pvals(
  outcome,
  data,
  pcutoff = 0.05,
  padj.method = "BH",
  group_test = c("anova", "kruskal.test"),
  pairwise_test = c("t.test", "wilcoxon"),
  exact = FALSE,
  filter_pairwise = TRUE
)
```

Arguments

`outcome`	Outcome vector with 3 groups, ideally as a factor. If it is not a factor, this will be coerced to a factor. This must have exactly 3 levels.
`data`	Dataframe or matrix with variables in columns
`pcutoff`	Cut-off for p-value significance

`padj.method`	Can be any method available in `p.adjust` or `"qvalue"`. The option "none" is a pass-through.
`group_test`	Specifies statistical test for 3-class group comparison. "anova" means one-way ANOVA, "kruskal.test" means Kruskal-Wallis test.
`pairwise_test`	Specifies statistical test for pairwise comparisons
`exact`	Logical which is only used with `pairwise_test = "wilcoxon"`
`filter_pairwise`	Logical. If `TRUE` (the default) p-value adjustment on pairwise statistical tests is only conducted on attributes which reached the threshold for significance after p-value adjustment on the group statistical test.

`calc_stats_2x3` *Calculate p-values for 2x3-way analysis*

Description

Calculate p-values for 2x3-way analysis

Usage

```
calc_stats_2x3(
  data,
  outcome,
  group,
  padj.method,
  test = c("t.test", "wilcoxon"),
  exact = FALSE
)
```

Arguments

`data`	Dataframe or matrix with variables in columns and samples in rows
`outcome`	Factor with 2 levels
`group`	Factor with 3 levels
`padj.method`	Can be `"qvalue"` or any method available in `p.adjust`. The option `"none"` is a pass-through.
`test`	Character value specifying the statistical test between the 2 level response outcome. Current options are "t.test" or "wilcoxon".
`exact`	Logical for whether to use an exact test (Wilcoxon test only)

`deseq_2x3` *2 x 3 factor DESeq2 analysis*

Description

Experimental function for performing 2x3 factor DESeq2 analyses. Output can be passed to `deseq_2x3_polar()` and subsequently plotted. Example usage would include comparing gene expression against a binary outcome e.g. response vs non-response, across 3 drugs: the design would be ~ `response` and `group` would refer to the medication column in the metadata.

Usage

```
deseq_2x3(object, design, group, ...)
```

Arguments

`object`	An object of class 'DESeqDataSet' containing full dataset
`design`	Design formula. The main contrast is taken from the last term of the formula and must be a binary factor.
`group`	Character value for the column with the 3-way grouping factor within the sample information data `colData`
`...`	Optional arguments passed to `DESeq()`.

Value

Returns a list of 3 DESeq2 results objects which can be passed onto `deseq_2x3_polar()`

Examples

```
# Basic DESeq2 set up
library(DESeq2)

counts <- matrix(rnbinom(n=3000, mu=100, size=1/0.5), ncol=30)
rownames(counts) <- paste0("gene", 1:100)
cond <- rep(factor(rep(1:3, each=5), labels = c('A', 'B', 'C')), 2)
resp <- factor(rep(1:2, each=15), labels = c('non.responder', 'responder'))
metadata <- data.frame(drug = cond, response = resp)

# Full dataset object construction
dds <- DESeqDataSetFromMatrix(counts, metadata, ~response)

# Perform 3x DESeq2 analyses comparing binary response for each drug
res <- deseq_2x3(dds, ~response, "drug")

# Generate polar object
obj <- deseq_2x3_polar(res)
```

```
# 2d plot
radial_plotly(obj)

# 3d plot
volcano3D(obj)
```

`deseq_2x3_polar` *Convert 2x3 design DESeq2 objects to a volcano3d object*

Description

This function is used instead of `polar_coords` if you have raw RNA-Seq count data. It takes results from a 2x3 design DESeq2 analysis using `deseq_2x3()`. Alternatively it will accept a list of 3 `DESeqDataSet` or `DESeqResults` objects each with the same type of binary comparison across 3 groups. Statistical results are extracted and converted to a 'volc3d' object, which can be directly plotted. Example usage would include comparing gene expression against a binary outcome e.g. response vs non-response, across 3 drugs.

Usage

```
deseq_2x3_polar(
  object,
  pcutoff = 0.05,
  fc_cutoff = NULL,
  padj.method = "BH",
  process = c("positive", "negative", "two.sided"),
 scheme = c("grey60", "red", "gold2", "green3", "cyan", "blue", "purple", "black"),
  labs = NULL
)
```

Arguments

`object`	A named list of 3 objects of class 'DESeqDataSet', or a list of 3 DESeq2 results dataframes generated by calling `DESeq2::results()`. Each object should contain the binary outcome comparison for each of the 3 groups, e.g. group A response vs non-response, group B response vs non-response, group C etc. The names of the groups are taken from the list names.
`pcutoff`	Cut-off for p-value significance
`fc_cutoff`	Cut-off for fold change on radial axis
`padj.method`	Can be `"qvalue"` or any method available in `p.adjust`. The option `"none"` is a pass-through.
`process`	Character value specifying colour process for statistical significant genes: "positive" specifies genes are coloured if fold change is >0; "negative" for genes with fold change <0 (note that for clarity the polar position is altered so that

	genes along each axis have the most strongly negative fold change values); or "two.sided" which is a compromise in which positive genes are labelled as before but genes with negative fold changes and significant p-values have an inverted colour scheme.
`scheme`	Vector of colours starting with non-significant genes/variables
`labs`	Optional character vector for labelling groups. Default `NULL` leads to abbreviated labels based on levels in `outcome` using `abbreviate()`. A vector of length 3 with custom abbreviated names for the outcome levels can be supplied. Otherwise a vector length 8 is expected, of the form "ns", "A+", "A+B+", "B+", "B+C+", "C+", "A+C+", "A+B+C+", where "ns" means non-significant and A, B, C refer to levels 1, 2, 3 in `outcome`, and must be in the correct order.

Details

This function generates a 'volc3d' class object for visualising a 2x3 way analysis for RNA-Seq data. For usual workflow it is typically preceded by a call to `deseq_2x3()` which runs the 3x DESeq2 analyses required.

Scaled polar coordinates are based on the DESeq2 statistic for each group comparison. Unscaled polar coordinates are generated using the log2 fold change for each group comparison.

The z axis for 3d volcano plots does not have as clear a corollary in 2x3 analysis as for the standard 3-way analysis (which uses the likelihood ratio test for the 3 groups). For 2x3 polar analysis the smallest p-value from the 3 group pairwise comparisons for each gene is used to generate a z coordinate as -log10(p-value).

The colour scheme is not as straightforward as for the standard polar plot and volcano3D plot since genes (or attributes) can be significantly up or downregulated in the response comparison for each of the 3 groups. `process = "positive"` means that genes are labelled with colours if a gene is significantly upregulated in the response for that group. This uses the primary colours (RGB) so that if a gene is upregulated in both red and blue groups it becomes purple etc with secondary colours. If the gene is upregulated in all 3 groups it is labelled black. Non-significant genes are in grey.

With `process = "negative"` genes are coloured when they are significantly downregulated. With `process = "two.sided"` the colour scheme means that both significantly up- and down-regulated genes are coloured with downregulated genes labelled with inverted colours (i.e. cyan is the inverse of red etc). However, significant upregulation in a group takes precedence.

Value

Returns an S4 'volc3d' object containing:

- 'df' A list of 2 dataframes. Each dataframe contains both x,y,z coordinates as well as polar coordinates r, angle. The first dataframe has coordinates based on the DESeq2 statistic. The 2nd dataframe is unscaled and represents log2 fold change for gene expression. The `type` argument in `volcano3D`, `radial_plotly` and `radial_ggplot` corresponds to these dataframes.
- 'outcome' An empty factor whose levels are the three-group contrast factor for comparisons
- 'data' Empty dataframe for compatibility
- 'pvals' A dataframe containing p-values. Columns 1-3 are pairwise comparisons between the outcome factor for groups A, B, C respectively.

- ’padj’ A dataframe containing p-values adjusted for multiple testing
- ’pcutoff Numeric value for cut-off for p-value significance
- ’scheme’ Character vector with colour scheme for plotting
- ’labs’ Character vector with labels for colour groups

See Also

`deseq_2x3()`

`deseq_polar` *Convert DESeq2 objects to a volcano3d object*

Description

This function is used instead of `polar_coords` if you have raw RNA-Seq count data. It takes 2 `DESeqDataSet` objects, extracts statistical results and converts the results to a ’volc3d’ object, which can be directly plotted.

Usage

```
deseq_polar(
  object,
  objectLRT,
  contrast = NULL,
  data = NULL,
  pcutoff = 0.05,
  padj.method = "BH",
  filter_pairwise = TRUE,
  ...
)
```

Arguments

`object`	An object of class ’DESeqDataSet’ with the full design formula. The function `DESeq` needs to have been run.
`objectLRT`	An object of class ’DESeqDataSet’ with the reduced design formula. The function `DESeq` needs to have been run on this object with argument `test="LRT"`.
`contrast`	Character value specifying column within the metadata stored in the DESeq2 dataset objects is the outcome variable. This column must contain a factor with 3 levels. If not set, the function will select the last term in the design formula of `object` as per DESeq2 convention.
`data`	Optional matrix containing gene expression data. If not supplied, the function will pull the expression data from within the DESeq2 object using the DESeq2 function `assay()`. NOTE: for consistency with gene expression datasets, genes are in rows.

`pcutoff`	Cut-off for p-value significance
`padj.method`	Can be any method available in `p.adjust` or `"qvalue"`. The option `"none"` is a pass-through.
`filter_pairwise`	Logical whether adjusted p-value pairwise statistical tests are only conducted on genes which reach significant adjusted p-value cut-off on the group likelihood ratio test
`...`	Optional arguments passed to `polar_coords`

Value

Calls `polar_coords` to return an S4 'volc3d' object

See Also

`polar_coords`, `voom_polar`, `DESeq` in the DESeq2 package

Examples

```
library(DESeq2)

counts <- matrix(rnbinom(n=1500, mu=100, size=1/0.5), ncol=15)
cond <- factor(rep(1:3, each=5), labels = c('A', 'B', 'C'))

# object construction
dds <- DESeqDataSetFromMatrix(counts, DataFrame(cond), ~ cond)

# standard analysis
dds <- DESeq(dds)

# Likelihood ratio test
ddsLRT <- DESeq(dds, test="LRT", reduced= ~ 1)

polar <- deseq_polar(dds, ddsLRT, "cond")
volcano3D(polar)
radial_ggplot(polar)
```

`forest_plot` *Forest plot individual gene from 2x3 factor analysis*

Description

Forest plot individual gene from 2x3 factor analysis using either base graphics, plotly or ggplot2.

Usage

```
forest_plot(
  object,
  genes,
  scheme = c("red", "green3", "blue"),
  labs = NULL,
  error_type = c("ci", "se"),
  error_width = 0.05,
  gap = 1,
  transpose = FALSE,
  mar = if (transpose) c(5, 7, 5, 4) else c(5, 5, 5, 3),
  ...
)

forest_plotly(
  object,
  genes,
  scheme = c("red", "green3", "blue"),
  labs = NULL,
  error_type = c("ci", "se"),
  error_width = 4,
  gap = 1,
  transpose = FALSE,
  ...
)

forest_ggplot(
  object,
  genes,
  scheme = c("red", "green3", "blue"),
  labs = NULL,
  error_type = c("ci", "se"),
  error_width = 0.3,
  facet = TRUE,
  gap = 1,
  transpose = FALSE,
  ...
)
```

Arguments

`object`	A 'volc3d' class object from a 2x3 analysis generated by `deseq_2x3_polar()`
`genes`	Vector of genes to plot
`scheme`	Vector of 3 colours for plotting
`labs`	Optional character vector of labels for the groups
`error_type`	Either "ci" or "se" to specify whether error bars use 95% confidence intervals or standard error

`error_width`	Width of error bars
`gap`	Size of gap between groupings for each gene
`transpose`	Logical whether to transpose the plot
`mar`	Vector of margins on four sides. See `par()`
`...`	Optional arguments
`facet`	Logical whether to use facets for individual genes (ggplot2 only)

Value

Returns a plot using either base graphics (`forest_plot`), plotly (`forest_plotly`) or ggplot2 (`forest_ggplot`). `forest_plot` also invisibly returns the dataframe used for plotting.

See Also

`deseq_2x3_polar()`

`grid-class` *An S4 class to define the polar grid coordinates system.*

Description

An S4 class to define the polar grid coordinates system.

Slots

`polar_grid`	The coordinates for the cylindrical grid segments with x,y,z coordinates
`axes`	The axes features for 'plotly'
`axis_labs`	The axis labels
`r`	The grid radius
`z`	The grid height
`text_coords`	data frame for axis label cartesian coordinates (x, y, z)
`n_r_breaks`	The number of ticks on the r axis
`n_z_breaks`	The number of ticks on the z axis
`r_breaks`	The r axis ticks as a numeric
`z_breaks`	The z axis ticks as a numeric

`polar_coords` *Coordinates for Three Way Polar Plot*

Description

This function creates a 'volc3d' object of S4 class for downstream plots containing the p-values from a three-way group comparison, expression data sample data and polar coordinates. For RNA-Seq count data, two functions `deseq_polar` or `voom_polar` can be used instead.

Usage

```
polar_coords(
  outcome,
  data,
  pvals = NULL,
  padj = pvals,
  pcutoff = 0.05,
  fc_cutoff = NULL,
  scheme = c("grey60", "red", "gold2", "green3", "cyan", "blue", "purple"),
  labs = NULL,
  ...
)
```

Arguments

`outcome`	Outcome vector with 3 groups, ideally as a factor. If it is not a factor, this will be coerced to a factor. This must have exactly 3 levels. NOTE: if `pvals` is given, the order of the levels in `outcome` must correspond to the order of columns in `pvals`.
`data`	Dataframe or matrix with variables in columns
`pvals`	Matrix or dataframe with p-values. The first column represents a test across all 3 categories such as one-way ANOVA or likelihood ratio test. Columns 2-4 represent pairwise tests comparing groups A vs B, A vs C and B vs C, where A, B, C represent levels 1, 2, 3 in `outcome`. Columns 2-4 must be provided in the correct order. If `pvals` is not given, it is calculated using the function `calc_pvals`.
`padj`	Matrix or dataframe with adjusted p-values. If not supplied, defaults to use nominal p-values from `pvals`.
`pcutoff`	Cut-off for p-value significance
`fc_cutoff`	Cut-off for fold change on radial axis
`scheme`	Vector of colours starting with non-significant variables
`labs`	Optional character vector for labelling groups. Default `NULL` leads to abbreviated labels based on levels in `outcome` using `abbreviate()`. A vector of length 3 with custom abbreviated names for the outcome levels can be supplied. Otherwise a vector length 7 is expected, of the form "ns", "B+", "B+C+", "C+",

	"A+C+", "A+", "A+B+", where "ns" means non-significant and A, B, C refer to levels 1, 2, 3 in `outcome`, and must be in the correct order.
...	Optional arguments passed to `calc_pvals`

Value

Returns an S4 'volc3d' object containing:

- 'df' A list of 2 dataframes. Each dataframe contains both x,y,z coordinates as well as polar coordinates r, angle. The first dataframe has coordinates on scaled data. The 2nd dataframe has unscaled data (e.g. log2 fold change for gene expression). The `type` argument in `volcano3D`, `radial_plotly` and `radial_ggplot` corresponds to these dataframes.
- 'outcome' The three-group contrast factor used for comparisons
- 'data' Dataframe or matrix containing the expression data
- 'pvals' A dataframe containing p-values. First column is the 3-way comparison (LRT or ANOVA). Columns 2-4 are pairwise comparisons between groups A vs B, A vs C and B vs C, where A, B, C are the 3 levels in the outcome factor.
- 'padj' A dataframe containing p-values adjusted for multiple testing
- 'pcutoff Numeric value for cut-off for p-value significance
- 'scheme' Character vector with colour scheme for plotting
- 'labs' Character vector with labels for colour groups

See Also

`deseq_polar`, `voom_polar`, `calc_pvals`

Examples

```
data(example_data)
syn_polar <- polar_coords(outcome = syn_example_meta$Pathotype,
                          data = t(syn_example_rld))
```

`polar_coords_2x3` *Coordinates for three way polar plot from 2x3 factor analysis*

Description

This function creates a 'volc3d' object of S4 class for downstream plots containing the p-values from a 2x3 factor analysis, expression data sample data and polar coordinates. For RNA-Seq count data, two functions `deseq_2x3` followed by `deseq_2x3_polar()` can be used instead.

Usage

```
polar_coords_2x3(
  data,
  metadata = NULL,
  outcome,
  group,
  pvals = NULL,
  padj = pvals,
  pcutoff = 0.05,
  fc_cutoff = NULL,
  padj.method = "BH",
  process = c("positive", "negative", "two.sided"),
 scheme = c("grey60", "red", "gold2", "green3", "cyan", "blue", "purple", "black"),
  labs = NULL,
  ...
)
```

Arguments

`data`	Dataframe or matrix with variables in columns and samples in rows
`metadata`	Dataframe of sample information with samples in rows
`outcome`	Either the name of column in `metadata` containing the binary outcome data. Or a vector with 2 groups, ideally a factor. If it is not a factor, this will be coerced to a factor. This must have exactly 2 levels.
`group`	Either the name of column in `metadata` containing the 3-way grouping data. Or a vector with 3 groups, ideally a factor. If it is not a factor, this will be coerced to a factor. This must have exactly 3 levels. NOTE: if `pvals` is given, the order of the levels in `group` must correspond to the order of columns in `pvals`.
`pvals`	Optional matrix or dataframe with p-values in 3 columns. If `pvals` is not given, it is calculated using the function `calc_stats_2x3`. The p-values in 3 columns represent the comparison between the binary outcome with each column for the 3 groups as specified in `group`.
`padj`	Matrix or dataframe with adjusted p-values. If not supplied, defaults to use nominal p-values from `pvals`.
`pcutoff`	Cut-off for p-value significance
`fc_cutoff`	Cut-off for fold change on radial axis
`padj.method`	Can be `"qvalue"` or any method available in `p.adjust`. The option `"none"` is a pass-through.
`process`	Character value specifying colour process for statistical significant genes: "positive" specifies genes are coloured if fold change is >0; "negative" for genes with fold change <0 (note that for clarity the polar position is altered so that genes along each axis have the most strongly negative fold change values); or "two.sided" which is a compromise in which positive genes are labelled as before but genes with negative fold changes and significant p-values have an inverted colour scheme.

`scheme`	Vector of colours starting with non-significant variables
`labs`	Optional character vector for labelling groups. Default `NULL` leads to abbreviated labels based on levels in `outcome` using `abbreviate()`. A vector of length 3 with custom abbreviated names for the outcome levels can be supplied. Otherwise a vector length 8 is expected, of the form "ns", "A+", "A+B+", "B+", "B+C+", "C+", "A+C+", "A+B+C+", where "ns" means non-significant and A, B, C refer to levels 1, 2, 3 in `outcome`, and must be in the correct order.
`...`	Optional arguments passed to `calc_stats_2x3`

Details

This function is designed for manually generating a ’volc3d’ class object for visualising a 2x3 way analysis comparing a large number of attributes such as genes. For RNA-Seq data we suggest using `deseq_2x3()` and `deseq_2x3_polar()` functions in sequence instead.

Scaled polar coordinates are generated using the t-score for each group comparison. Unscaled polar coordinates are generated as difference between means for each group comparison. If p-values are not supplied they are calculated by `calc_stats_2x3()` using either t-tests or wilcoxon tests.

The z axis for 3d volcano plots does not have as clear a corollary in 2x3 analysis as for the standard 3-way analysis (which uses the likelihood ratio test for the 3 groups). For 2x3 polar analysis the smallest p-value from the 3 group pairwise comparisons for each gene is used to generate a z coordinate as -log10(p-value).

The colour scheme is not as straightforward as for the standard polar plot and volcano3D plot since genes (or attributes) can be significantly up or downregulated in the response comparison for each of the 3 groups. `process = "positive"` means that genes are labelled with colours if a gene is significantly upregulated in the response for that group. This uses the primary colours (RGB) so that if a gene is upregulated in both red and blue group it becomes purple etc with secondary colours. If the gene is upregulated in all 3 groups it is labelled black. Non-significant genes are in grey.

With `process = "negative"` genes are coloured when they are significantly downregulated. With `process = "two.sided"` the colour scheme means that both significantly up- and down-regulated genes are coloured with downregulated genes labelled with inverted colours (i.e. cyan is the inverse of red etc). However, significant upregulation in a group takes precedence.

Value

Returns an S4 ’volc3d’ object containing:

- ’df’ A list of 2 dataframes. Each dataframe contains both x,y,z coordinates as well as polar coordinates r, angle. The first dataframe has coordinates on scaled data. The 2nd dataframe has unscaled data (e.g. log2 fold change for gene expression). The `type` argument in `volcano3D`, `radial_plotly` and `radial_ggplot` corresponds to these dataframes.
- ’outcome’ The three-group contrast factor used for comparisons, linked to the `group` column
- ’data’ Dataframe or matrix containing the expression data
- ’pvals’ A dataframe containing p-values in 3 columns representing the binary comparison for the outcome for each of the 3 groups.
- ’padj’ A dataframe containing p-values adjusted for multiple testing
- ’pcutoff Numeric value for cut-off for p-value significance

- 'scheme' Character vector with colour scheme for plotting
- 'labs' Character vector with labels for colour groups

See Also

`deseq_2x3`, `deseq_2x3_polar`, `calc_stats_2x3`

`polar_grid` *Grid required for 3D volcano plot and 2D radial plots*

Description

Generates a cylindrical grid of the appropriate dimensions for a 3D volcano plot

Usage

```
polar_grid(
  r_vector = NULL,
  z_vector = NULL,
  r_axis_ticks = NULL,
  z_axis_ticks = NULL,
  axis_angle = 5/6,
  n_spokes = 12,
  axes_from_origin = TRUE,
  ...
)
```

Arguments

`r_vector`	An optional numerical vector for the radial coordinates. This is used to calculate breaks on the r axis using `pretty`. If this is NULL the r_axis_ticks are used as breaks.
`z_vector`	An optional numerical vector for the z coordinates. This is used to calculate breaks on the z axis using `pretty`. If this is NULL the z_axis_ticks are used as breaks.
`r_axis_ticks`	A numerical vector of breaks for the radial axis (used if r_vector is NULL).
`z_axis_ticks`	A numerical vector of breaks for the z axis (used if z_vector is NULL).
`axis_angle`	angle in radians to position the radial axis (default = 5/6)
`n_spokes`	the number of outward spokes to be plotted (default = 12)
`axes_from_origin`	Whether the axis should start at 0 or the first break (default = TRUE)
`...`	optional parameters for `pretty` on the r axis

Value

Returns an S4 grid object containing:

- 'polar_grid' The coordinates for a radial grid
- 'axes' The axes features for 'plotly'
- 'axis_labels' The axis labels
- 'r' The grid radius
- 'z' The grid height
- 'text_coords' The coordinates for text labels
- 'n_r_breaks' The number of ticks on the r axis
- 'n_r_breaks' The number of ticks on the z axis

References

Lewis, Myles J., et al. (2019). Molecular portraits of early rheumatoid arthritis identify clinical and treatment response phenotypes. *Cell reports*, **28**:9

Examples

```
data(example_data)
syn_polar <- polar_coords(outcome = syn_example_meta$Pathotype,
                          data = t(syn_example_rld))

grid <- polar_grid(r_vector=syn_polar@df[[1]]$r,
           z_vector=NULL,
           r_axis_ticks = NULL,
           z_axis_ticks = c(0, 8, 16, 32),
           n_spokes = 4)
```

`radial_ggplot` *'Ggplot' for Three Way Polar Plot*

Description

This function creates a 3-way polar plot using 'ggplot' for a three-class comparison.

Usage

```
radial_ggplot(
  polar,
  type = 1,
  colours = NULL,
  label_rows = NULL,
  arrow_length = 1,
  label_size = 5,
  colour_code_labels = FALSE,
```

```
  label_colour = "black",
  grid_colour = "grey80",
  grid_width = 0.7,
  axis_colour = "black",
  axis_width = 1,
  axis_title_size = 5,
  axis_label_size = 3,
  marker_alpha = 0.7,
  marker_size = 3,
  marker_outline_colour = "white",
  marker_outline_width = 0.5,
  legend_size = 20,
  ...
)
```

Arguments

`polar`	A 'volc3d' object with the p-values between groups of interest and polar coordinates created by `polar_coords`, `deseq_polar` or `voom_polar`.
`type`	Numeric value whether to use scaled (z-score) or unscaled (fold change) as magnitude. Options are 1 = z-score (default) or 2 = unscaled/fold change.
`colours`	A vector of colours for the non-significant points and each of the six groups.
`label_rows`	A vector of row names or indices to label
`arrow_length`	The length of label arrows
`label_size`	Font size of labels/annotations (default = 5).
`colour_code_labels`	Logical whether label annotations should be colour coded. If FALSE `label_colour` is used.
`label_colour`	Colour of annotation labels if not colour coded
`grid_colour`	The colour of the grid (default="grey80")
`grid_width`	The width of the axis lines (default=0.6)
`axis_colour`	The colour of the grid axes and labels (default="black")
`axis_width`	The width of the axis lines (default=1)
`axis_title_size`	Font size for axis titles (default = 5)
`axis_label_size`	Font size for axis labels (default = 3)
`marker_alpha`	The alpha parameter for markers (default = 0.7)
`marker_size`	Size of the markers (default = 3)
`marker_outline_colour`	Colour for marker outline (default = white)
`marker_outline_width`	Width for marker outline (default = 0.5)
`legend_size`	Size for the legend text (default = 20).
`...`	Optional parameters passed to `polar_grid` e.g. `r_axis_ticks` or `axis_angle`

Value

Returns a polar 'ggplot' object featuring variables on a tri-axis radial graph

References

Lewis, Myles J., et al. (2019). Molecular portraits of early rheumatoid arthritis identify clinical and treatment response phenotypes. *Cell reports*, **28**:9

See Also

`polar_coords`

Examples

```
data(example_data)
syn_polar <- polar_coords(outcome = syn_example_meta$Pathotype,
                          data = t(syn_example_rld))

radial_ggplot(polar = syn_polar, label_rows = c("COBL"))
```

`radial_plotly` *Three-way radial comparison Polar Plot (using plotly)*

Description

This function creates an interactive plotly object which maps differential expression onto a polar coordinates.

Usage

```
radial_plotly(
  polar,
  type = 1,
  colours = polar@scheme,
  label_rows = NULL,
  arrow_length = 80,
  label_size = 14,
  colour_code_labels = FALSE,
  label_colour = "black",
  grid_colour = "grey80",
  grid_width = 1,
  marker_size = 7,
  marker_alpha = 0.8,
  marker_outline_colour = "white",
  marker_outline_width = 0.5,
  axis_title_size = 16,
  axis_label_size = 10,
```

```
  axis_colour = "black",
  axis_width = 2,
  ...
)
```

Arguments

`polar`	A 'volc3d' object with the p-values between groups of interest and polar coordinates created by `polar_coords`, `deseq_polar` or `voom_polar`.
`type`	Numeric value whether to use scaled (Z-score) or unscaled (fold change) as magnitude. Options are 1 = Z-score (default) or 2 = unscaled/fold change.
`colours`	A vector of colour names or hex triplets for the non-significant points and each of the six groups.
`label_rows`	A vector of row names or numbers to label.
`arrow_length`	The length of label arrows (default = 80).
`label_size`	Font size of labels/annotations (default = 14)
`colour_code_labels`	Logical whether label annotations should be colour coded. If FALSE label_colour is used.
`label_colour`	HTML colour of annotation labels if not colour coded.
`grid_colour`	The colour of the grid (default="grey80")
`grid_width`	The width of the grid lines (default=1)
`marker_size`	Size of the markers (default = 6)
`marker_alpha`	Opacity for the markers (default = 0.7)
`marker_outline_colour`	Colour for marker outline (default = white)
`marker_outline_width`	Width for marker outline (default = 0.5)
`axis_title_size`	Font size for axis titles (default = 16)
`axis_label_size`	Font size for axis labels (default = 10)
`axis_colour`	The colour of the grid axes and labels (default="black")
`axis_width`	The width of the axis lines (default=2)
`...`	Optional parameters passed to `polar_grid` e.g. `r_axis_ticks` or `axis_angle`

Details

This function builds a layered plotly object. By default this produces an SVG output, but this can be slow with 1000s of points. For large number of points we recommend switching to webGL by piping to `toWebGL()` as shown in the examples.

Value

Returns a plotly plot featuring variables on a tri-axis radial graph

References

Lewis, Myles J., et al. (2019). Molecular portraits of early rheumatoid arthritis identify clinical and treatment response phenotypes. *Cell reports*, **28**:9

See Also

`polar_coords`

Examples

```
data(example_data)
syn_polar <- polar_coords(outcome = syn_example_meta$Pathotype,
                          data = t(syn_example_rld))

radial_plotly(polar = syn_polar, label_rows = c("COBL"))

## Faster webGL version for large numbers of points
library(plotly)
radial_plotly(polar = syn_polar, label_rows = c("COBL")) %>%
  toWebGL()
```

`show_grid` *Plots grid objects for inspection using plotly*

Description

This function creates an interactive grids in polar and cylindrical coordinates

Usage

```
show_grid(grid, plot_height = 700, axis_angle = 0, z_axis_title_offset = 1.2)
```

Arguments

`grid`	A grid object produced by `polar_grid`.
`plot_height`	The plot height in px (default=700),
`axis_angle`	The angle in radians at which to add axis (default=0).
`z_axis_title_offset`	Offset for z axis title (default=1.2).

Value

Returns a list containing a polar and cylindrical coordinate system.

References

Lewis, Myles J., et al. (2019). Molecular portraits of early rheumatoid arthritis identify clinical and treatment response phenotypes. *Cell reports*, **28**:9

Examples

```
data(example_data)
syn_polar <- polar_coords(outcome = syn_example_meta$Pathotype,
                          data = t(syn_example_rld))

grid <- polar_grid(r_vector=syn_polar@df[[1]]$r,
        z_vector=syn_polar@df[[1]]$z,
        r_axis_ticks = NULL,
        z_axis_ticks = NULL)
p <- show_grid(grid)
p$polar
p$cyl
```

`significance_subset` *Extract a subset population*

Description

Subsets data according to the significance groups.

Usage

```
significance_subset(polar, significance = NULL, output = "pvals")
```

Arguments

`polar`	A polar object including expression data from groups of interest. Created by `polar_coords`.
`significance`	Which significance factors to subset to. If `NULL` all levels except 'ns' (non-significant) are selected.
`output`	What object to return. Options are "pvals", "padj", "data", "df" for subset dataframes, or "polar" to subset the entire 'volc3d' class object.

References

Lewis, Myles J., et al. (2019). Molecular portraits of early rheumatoid arthritis identify clinical and treatment response phenotypes. *Cell reports*, **28**:9

Examples

```
data(example_data)
syn_polar <- polar_coords(outcome = syn_example_meta$Pathotype,
                          data = t(syn_example_rld))

subset <- significance_subset(syn_polar, "L+", "df")
```

`syn_example_meta` *PEAC synovial sample data*

Description

A dataset containing sample data for 81 synovial biopsies from the PEAC cohort

Usage

```
syn_example_meta
```

Format

A data frame with 81 rows and 1 variables:

Pathotype The synovial biopsy histological pathotype

Source

https://pubmed.ncbi.nlm.nih.gov/31461658/

`syn_example_rld` *PEAC synovial gene expression data*

Description

A dataset containing the gene expression data for 81 synovial biopsies from the PEAC cohort

Usage

```
syn_example_rld
```

Format

A data frame with 500 rows representing the most significant genes/probes and 81 columns representing samples.

Source

https://pubmed.ncbi.nlm.nih.gov/31461658/

`volc3d-class` *An S4 class to define the polar coordinates.*

Description

An S4 class to define the polar coordinates.

Slots

`df` List of coordinate data frames for scaled and unscaled expression

`outcome` Outcome vector

`data` Expression data

`pvals` Matrix or dataframe with p-values

`padj` Matrix adjusted p-values

`pcutoff` Cut-off for p-value significance

`scheme` Vector for colour scheme

`labs` Character vector for labelling groups

`volcano3D` *Three-Dimensional Volcano Plot*

Description

Plots the three-way comparisons of variables such as gene expression data in 3D space using plotly. x, y position represents polar position on 3 axes representing the amount each variable or gene tends to each of the 3 categories. The z axis represents -log10 P value for the one-way test comparing each variable across the 3 groups.

Usage

```
volcano3D(
  polar,
  type = 1,
  label_rows = c(),
  label_size = 14,
  arrow_length = 100,
  colour_code_labels = FALSE,
  label_colour = "black",
  grid_colour = "grey80",
  grid_width = 2,
  grid_options = NULL,
  axis_colour = "black",
  axis_width = 2,
```

```
  marker_size = 3,
  marker_outline_width = 0,
  marker_outline_colour = "white",
  z_axis_title_offset = 1.2,
  z_axis_title_size = 12,
  z_axis_angle = 0.5,
  radial_axis_title_size = 14,
  radial_axis_title_offset = 1.2,
  xy_aspectratio = 1,
  z_aspectratio = 0.8,
  camera_eye = list(x = 0.9, y = 0.9, z = 0.9),
  ...
)
```

Arguments

`polar`	Object of S4 class 'volc3d' following call to either `polar_coords`, `deseq_polar` or `voom_polar`
`type`	Either `1` or `2` specifying type of polar coordinates: `1` = Z-scaled, `2` = unscaled (equivalent to log2 fold change for gene expression).
`label_rows`	A vector of row names or numbers to label
`label_size`	font size for labels (default 14).
`arrow_length`	The length of label arrows (default 100)
`colour_code_labels`	Logical whether label annotations should be colour coded. If `FALSE` `label_colour` is used.
`label_colour`	HTML colour of annotation labels if not colour coded.
`grid_colour`	The colour of the cylindrical grid (default "grey80")
`grid_width`	The width of the grid lines (default 2)
`grid_options`	Optional list of additional arguments to pass to `polar_grid`, eg. `z_axis_ticks` and `r_axis_ticks`
`axis_colour`	The colour of the grid axes and labels (default "black")
`axis_width`	The width of axis lines (default 2)
`marker_size`	Size of the markers (default 3)
`marker_outline_width`	Width for marker outline (default 0 means no outline)
`marker_outline_colour`	Colour for marker outline (default white)
`z_axis_title_offset`	The position scaling between grid and z axis title (default=1.2)
`z_axis_title_size`	The font size for the z axis title (default=12)
`z_axis_angle`	Angle in radians for the position of z axis (default 0.5)

`radial_axis_title_size`	The font size for the radial (default=15)
`radial_axis_title_offset`	The position scaling between grid and radial axis title (default=1.2)
`xy_aspectratio`	The aspect ratio for the xy axis compared to z (default 1). Increasing this makes the grid wider in the plot window.
`z_aspectratio`	The aspect ratio for the z axis compared to x and y (default 0.8). Decreasing this makes the plot appear more squat.
`camera_eye`	The (x,y,z) components of the start ’eye’ camera vector. This vector determines the view point about the origin of this scene.
`...`	Optional arguments passed to `plot_ly`

Value

Returns a cylindrical 3D plotly plot featuring variables on a tri-axis radial graph with the -log10(multi-group test p-value) on the z-axis

References

Lewis, Myles J., et al. (2019). Molecular portraits of early rheumatoid arthritis identify clinical and treatment response phenotypes. *Cell reports*, **28**:9

See Also

`polar_coords`

Examples

```
data(example_data)
syn_polar <- polar_coords(outcome = syn_example_meta$Pathotype,
                          data = t(syn_example_rld))
volcano3D(syn_polar)
```

`voom_polar` *Convert RNA-Seq count data to a volcano3d object using ’limma voom’*

Description

This function is used instead of `polar_coords` if you have raw RNA-Seq count data. The function takes a design formula, metadata and raw RNA-Seq count data and uses ’limma voom’ to analyse the data. The results are converted to a ’volc3d’ object ready for plotting a 3d volcano plot or polar plot.

Usage

```
voom_polar(
  formula,
  metadata,
  counts,
  pcutoff = 0.05,
  padj.method = "BH",
  filter_pairwise = TRUE,
  ...
)
```

Arguments

`formula`	Design formula which must be of the form `~ 0 + outcome + ...`. The 3-way outcome variable must be the first variable after the '0', and this variable must be a factor with exactly 3 levels.
`metadata`	Matrix or dataframe containing metadata as referenced by `formula`
`counts`	Matrix containing raw gene expression count data
`pcutoff`	Cut-off for p-value significance
`padj.method`	Can be any method available in `p.adjust` or `"qvalue"`. The option "none" is a pass-through.
`filter_pairwise`	Logical whether adjusted p-value pairwise statistical tests are only conducted on genes which reach significant adjusted p-value cut-off on the group likelihood ratio test
`...`	Optional arguments passed to `polar_coords`

Details

Statistical results for the group and pairwise comparisons are calculated using the 'limma voom' pipeline and the results passed to `polar_coords` to generate a 'volc3d' object ready for plotting a 3d volcano plot or polar plot.

Value

Calls `polar_coords` to return an S4 'volc3d' object

See Also

`polar_coords`, `deseq_polar`, `voom` in the limma package

Examples

```
if (requireNamespace("limma", quietly = TRUE) &
    requireNamespace("edgeR", quietly = TRUE)) {
  library(limma)
  library(edgeR)
```

```
  counts <- matrix(rnbinom(n=1500, mu=100, size=1/0.5), ncol=15)
  cond <- factor(rep(1:3, each=5), labels = c('A', 'B', 'C'))
  cond <- data.frame(cond)

  polar <- voom_polar(~0 + cond, cond, counts)

  volcano3D(polar)
  radial_ggplot(polar)
}
```

Index

∗ **datasets**
    syn_example_meta, 25
    syn_example_rld, 25
∗ **dplot**
    polar_grid, 18
∗ **hplot**
    add_animation, 2
    boxplot_trio, 3
    radial_ggplot, 19
    radial_plotly, 21
    show_grid, 23
∗ **htest**
    polar_grid, 18
∗ **iplot**
    add_animation, 2
    radial_plotly, 21
    show_grid, 23
∗ **manip**
    polar_grid, 18
∗ **volcanoplot**
    add_animation, 2

abbreviate(), *9*, *14*, *17*
add_animation, 2
anova, *4*

boxplot_trio, 3

calc_pvals, 5, *14*, *15*
calc_stats_2x3, 6, *16–18*
calc_stats_2x3(), *17*

DESeq, *11*
deseq_2x3, 7, *15*, *18*
deseq_2x3(), *8–10*, *17*
deseq_2x3_polar, 8, *18*
deseq_2x3_polar(), *7*, *12*, *13*, *15*, *17*
deseq_polar, 10, *14*, *15*, *20*, *22*, *27*, *29*

forest_ggplot *(*forest_plot*)*, 11
forest_plot, 11
forest_plotly *(*forest_plot*)*, 11

grid-class, 13

kruskal.test, *4*

par(), *13*
plot_ly, *28*
polar_coords, *4*, *8*, *10*, *11*, 14, *20–24*, *27–29*
polar_coords_2x3, 15
polar_grid, 18, *20*, *22*, *23*, *27*
pretty, *18*

radial_ggplot, *9*, *15*, *17*, 19
radial_plotly, *9*, *15*, *17*, 21

show_grid, 23
significance_subset, 24
stat_compare_means, *4*
syn_example_meta, 25
syn_example_rld, 25

t.test, *4*

volc3d-class, 26
volcano3D, *3*, *9*, *15*, *17*, 26
voom, *29*
voom_polar, *11*, *14*, *15*, *20*, *22*, *27*, 28

wilcox.test, *4*