Package: DEGreport (via r-universe)

November 24, 2026

Version 1.50.0

Date 2025-10-11

Type Package

Title Report of DEG analysis

Description Creation of ready-to-share figures of differential expression analyses of count data. It integrates some of the code mentioned in DESeq2 and edgeR vignettes, and report a ranked list of genes according to the fold changes mean and variability for each selected gene.

biocViews DifferentialExpression, Visualization, RNASeq, ReportWriting, GeneExpression, ImmunoOncology

URL http://lpantano.github.io/DEGreport/

BugReports https://github.com/lpantano/DEGreport/issues

Suggests BiocStyle, AnnotationDbi, limma, pheatmap, rmarkdown, statmod, testthat

Depends R (>= 4.0.0)

Imports utils, methods, Biobase, BiocGenerics, broom, circlize, ComplexHeatmap, cowplot, ConsensusClusterPlus, cluster, dendextend, DESeq2, dplyr, edgeR, ggplot2, ggdendro, grid, ggrepel, grDevices, knitr, logging, magrittr, psych, RColorBrewer, reshape, rlang, scales, stats, stringr, stringi, S4Vectors, SummarizedExperiment, tidyr, tibble

Maintainer Lorena Pantano <lorena.pantano@gmail.com>

License MIT + file LICENSE

VignetteBuilder knitr

RoxygenNote 7.3.1

Encoding UTF-8

Roxygen list(markdown = TRUE)

Config/pak/sysreqs libicu-dev libpng-dev perl zlib1g-dev

Repository Bioconductor 3.24

Date/Publication 2026-11-23 22:34:04 UTC

RemoteUrl https://github.com/bioc/DEGreport

RemoteRef RELEASE_3_24

RemoteSha 248b64329ad63dcb5fa71dfe2bbfc54ac3d237de

Contents

DEGreport-package . . . 3
createReport . . . 4
deg . . . 4
degCheckFactors . . . 5
degColors . . . 6
degComps . . . 7
degCorCov . . . 8
degCovariates . . . 9
degDefault . . . 11
degFilter . . . 12
degMA . . . 13
degMB . . . 14
degMDS . . . 15
degMean . . . 15
degMerge . . . 16
degMV . . . 17
degObj . . . 18
degPatterns . . . 19
degPCA . . . 21
degPlot . . . 22
degPlotCluster . . . 24
degPlotWide . . . 25
degQC . . . 26
degResults . . . 27
DEGSet . . . 28
degSignature . . . 30
degSummary . . . 31
degVar . . . 32
degVB . . . 33
degVolcano . . . 33
geneInfo . . . 35
geom_cor . . . 35
humanGender . . . 37
significants . . . 37

Index **40**

DEGreport-package *Deprecated functions in package DEGreport*

Description

These functions are provided for compatibility with older versions of DEGreport only and will be defunct at the next release.

Details

The following functions are deprecated and will be made defunct; use the replacement indicated below:

- degRank, degPR, degBIcmd, degBI, degFC, degComb, degNcomb: DESeq2::lcfShrink. This function was trying to avoid big FoldChange in variable genes. There are other methods nowadays like lcfShrink function. DEGreport

Author(s)

Maintainer: Lorena Pantano <lorena.pantano@gmail.com>

Other contributors:

- John Hutchinson [contributor]
- Victor Barrera [contributor]
- Mary Piper [contributor]
- Radhika Khetani [contributor]
- Kenneth Daily [contributor]
- Thanneer Malai Perumal [contributor]
- Rory Kirchner [contributor]
- Michael Steinbaugh [contributor]
- Ivo Zeller [contributor]

See Also

Useful links:

- http://lpantano.github.io/DEGreport/
- Report bugs at https://github.com/lpantano/DEGreport/issues

createReport *Create report of RNAseq DEG anlaysis*

Description

This function get the count matrix, pvalues, and FC of a DEG analysis and create a report to help to detect possible problems with the data.

Usage

```
createReport(g, counts, tags, pvalues, path, pop = 400, name = "DEGreport")
```

Arguments

g	Character vector with the group the samples belong to.
counts	Matrix with counts for each samples and each gene. Should be same length than pvalues vector.
tags	Genes of DEG analysis
pvalues	pvalues of DEG analysis
path	path to save the figure
pop	random genes for background
name	name of the html file

Value

A HTML file with all figures and tables

deg *Method to get all table stored for an specific comparison*

Description

Method to get all table stored for an specific comparison

Usage

```
deg(object, value = NULL, tidy = NULL, top = NULL, ...)

## S4 method for signature 'DEGSet'
deg(object, value = NULL, tidy = NULL, top = NULL, ...)
```

Arguments

object	DEGSet
value	Character to specify which table to use.
tidy	Return data.frame, tibble or original class.
top	Limit number of rows to return. Default: All.
...	Other parameters to pass for other methods.

Author(s)

Lorena Pantano

References

- Testing if top is whole number or not comes from: https://stackoverflow.com/a/3477158

---

degCheckFactors *Distribution of gene ratios used to calculate Size Factors.*

Description

This function check the median ratio normalization used by DESeq2 and similarly by edgeR to visualy check whether the median is the best size factor to represent depth.

Usage

```
degCheckFactors(counts, each = FALSE)
```

Arguments

counts	Matrix with counts for each samples and each gene. row number should be the same length than pvalues vector.
each	Plot each sample separately.

Details

This function will plot the gene ratios for each sample. To calculate the ratios, it follows the simliar logic than DESeq2/edgeR uses, where the expression of each gene is divided by the mean expression of that gene. The distribution of the ratios should approximate to a normal shape and the factors should be similar to the median of distributions. If some samples show different distribution, the factor may be bias due to some biological or technical factor.

Value

ggplot2 object

References

- Code to calculate size factors comes from DESeq2::estimateSizeFactorsForMatrix().

Examples

```
data(humanGender)
library(SummarizedExperiment)
degCheckFactors(assays(humanGender)[[1]][, 1:10])
```

---

degColors *Make nice colors for metadata*

Description

The function will take a metadata table and use Set2 palette when number of levels is > 3 or a set or orange/blue colors other wise.

Usage

```
degColors(
  ann,
  col_fun = FALSE,
  con_values = c("grey80", "black"),
  cat_values = c("orange", "steelblue"),
  palette = "Set2"
)
```

Arguments

ann	Data.frame with metadata information. Each column will be used to generate a palette suitable for the values in there.
col_fun	Whether to return a function for continuous variables (compatible with ComplexHeatmap::HeatmapAnnot or the colors themself (comparible with [pheatmap::pheatmap())]). [pheatmap::pheatmap())]: R:pheatmap::pheatmap())
con_values	Color to be used for continuous variables.
cat_values	Color to be used for 2-levels categorical variables.
palette	Palette to use from brewer.pal() for multi-levels categorical variables.

Examples

```
data(humanGender)
library(DESeq2)
library(ComplexHeatmap)
idx <- c(1:10, 75:85)
dse <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:10, idx],
  colData(humanGender)[idx,], design=~group)
```

```
th <- HeatmapAnnotation(df = colData(dse),
                        col = degColors(colData(dse), TRUE))
Heatmap(log2(counts(dse)+0.5), top_annotation = th)

custom <- degColors(colData(dse), TRUE,
          con_values = c("white", "red"),
          cat_values = c("white", "black"),
          palette = "Set1")
th <- HeatmapAnnotation(df = colData(dse),
                         col = custom)
Heatmap(log2(counts(dse)+0.5), top_annotation = th)
```

`degComps` *Automatize the use of* `results()` *for multiple comparisons*

Description

This function will extract the output of `DESeq2::results()` and `DESeq2::lfcShrink()` for multiple comparison using:

Usage

```
degComps(
  dds,
  combs = NULL,
  contrast = NULL,
  alpha = 0.05,
  skip = FALSE,
  type = "normal",
  pairs = FALSE,
  fdr = "default"
)
```

Arguments

`dds`	DESeq2::DESeqDataSet obcject.
`combs`	Optional vector indicating the coefficients or columns fom `colData(dds)` to create group comparisons.
`contrast`	Optional vector to specify contrast. See `DESeq2::results()`.
`alpha`	Numeric value used in independent filtering in `DESeq2::results()`.
`skip`	Boolean to indicate whether skip shrinkage. For instance when it comes from LRT method.
`type`	Type of shrinkage estimator. See `DESeq2::lfcShrink()`.
`pairs`	Boolean to indicate whether create all comparisons or only use the coefficient already created from `DESeq2::resultsNames()`.
`fdr`	type of fdr correction. `default` is FDR value, `lfdr-stat` is for local FDR using the statistics of the test, `lfdr-pvalue` is for local FDR using the p-value of the test. fdrtools needs to be installed and loaded by the user

Details

- coefficients
- contrast
- Multiple columns in `colData` that match coefficients
- Multiple columns in `colData` to create all possible contrasts

Value

DEGSet with unSrunken and Srunken results.

Author(s)

Lorena Pantano

Examples

```
library(DESeq2)
dds <- makeExampleDESeqDataSet(betaSD=1)
colData(dds)[["treatment"]] <- sample(colData(dds)[["condition"]], 12)
  design(dds) <-  ~ condition + treatment
dds <- DESeq(dds)
res <- degComps(dds, combs = c("condition", 2),
                contrast = list("treatment_B_vs_A", c("condition", "A", "B")))
# library(fdrtools)
#res <- degComps(dds,contrast = list("treatment_B_vs_A"),
#                fdr="lfdr-stat")
```

`degCorCov` *Calculate the correlation relationshipt among all covariates in the metadata table*

Description

This function will calculate the correlation among all columns in the metadata

Usage

```
degCorCov(metadata, fdr = 0.05, use_pval = FALSE, ...)
```

Arguments

`metadata`	data.frame with samples metadata.
`fdr`	numeric value to use as cutoff to determine the minimum fdr to consider significant correlations between pcs and covariates.
`use_pval`	boolean to indicate to use p-value instead of FDR to hide non-significant correlation.
`...`	Parameters to pass to `ComplexHeatmap::Heatmap()`.

Value

: list: a) cor, data.frame with pair-wise correlations, pvalues, FDR b) corMat, data.frame with correlation matrix c) fdrMat, data.frame with FDR matrix b) plot, Heatmap plot of correlation matrix

Author(s)

: Lorena Pantano, Kenneth Daily and Thanneer Malai Perumal

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dse <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
cor <- degCorCov(colData(dse))
```

`degCovariates` *Find correlation between pcs and covariates*

Description

This function will calculate the pcs using prcomp function, and correlate categorical and numerical variables from metadata. The size of the dots indicates the importance of the metadata, for instance, when the range of the values is pretty small (from 0.001 to 0.002 in ribosimal content), the correlation results is not important. If black stroke lines are shown, the correlation analysis has a FDR < 0.05 for that variable and PC. Only significant variables according the linear model are colored. See details to know how this is calculated.

Usage

```
degCovariates(
  counts,
  metadata,
  fdr = 0.1,
  scale = FALSE,
  minPC = 5,
  correlation = "kendall",
  addCovDen = TRUE,
  legacy = FALSE,
  smart = TRUE,
  method = "lm",
  plot = TRUE
)
```

Arguments

counts	normalized counts matrix
metadata	data.frame with samples metadata.
fdr	numeric value to use as cutoff to determine the minimum fdr to consider significant correlations between pcs and covariates.
scale	boolean to determine wether counts matrix should be scaled for pca. default FALSE.
minPC	numeric value that will be used as cutoff to select only pcs that explain more variability than this.
correlation	character determining the method for the correlation between pcs and covariates.
addCovDen	boolean. Whether to add the covariates dendograme to the plot to see covariates relationship. It will show `degCorCov()` dendograme on top of the columns of the heatmap.
legacy	boolean. Whether to plot the legacy version.
smart	boolean. Whether to avoid normalization of the numeric covariates when calculating importance. This is not used if `legacy = TRUE`. See @details for more information.
method	character. Whether to use `lm` to calculate the significance of the variable during reduction step. See @details for more information.
plot	Whether to plot or not the correlation matrix.

Details

This method is adapeted from Daily et al 2017 article. Principal components from PCA analysis are correlated with covariates metadata. Factors are transformed to numeric variables. Correlation is measured by `cor.test` function with Kendall method by default.

The size of the dot, or importance, indicates the importance of the covariate based on the range of the values. Covariates where the range is very small (like a % of mapped reads that varies between 0.001 to 0.002) will have a very small size (0.1*max_size). The maximum value is set to 5 units. To get to importance, each covariate is normalized using this equation: `1 - min(v/max(v))`, and the minimum and maximum values are set to 0.01 and 1 respectively. For instance, 0.5 would mean there is at least 50% of difference between the minimum value and the maximum value. Categorical variables are plot using the maximum size always, since it is not possible to estimate the variability. By default, it won't do `v/max(v)` if the values are already between 0-1 or 0-100 (already normalized values as rates and percentages). If you want to ignore the importance, use `legacy = TRUE`.

Finally, a linear model is used to calculate the significance of the covariates effect on the PCs. For that, this function uses `lm` to regress the data and uses the p-value calculated by each variable in the model to define significance (pvalue < 0.05). Variables with a black stroke are significant after this step. Variables with grey stroke are significant at the first pass considering p.value < 0.05 for the correlation analysis.

Value

: list:

- plot, heatmap showing the signifcance of the variables.
- corMatrix, correlation, p-value, FDR values for each covariate and PCA pais
- pcsMatrix: PCs loading for each sample
- scatterPlot: plot for each significant covariate and the PC values.
- significants: contains the significant covariates using a linear model to predict the coefficient of covariates that have some color in the plot. All the significant covariates from the liner model analysis are returned.

Author(s)

: Lorena Pantano, Victor Barrera, Kenneth Daily and Thanneer Malai Perumal

References

Daily, K. et al. Molecular, phenotypic, and sample-associated data to describe pluripotent stem cell lines and derivatives. Sci Data 4, 170030 (2017).

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dse <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
res <- degCovariates(log2(counts(dse)+0.5), colData(dse))
res <- degCovariates(log2(counts(dse)+0.5),
  colData(dse), legacy = TRUE)
res$plot
res$scatterPlot[[1]]
```

---

degDefault *Method to get the default table to use.*

Description

It can accept a list of new padj values matching the same dimmensions than the current vector.

Usage

```
degDefault(object)

degCorrect(object, fdr)

## S4 method for signature 'DEGSet'
degDefault(object)

## S4 method for signature 'DEGSet'
degCorrect(object, fdr)
```

Arguments

object	DEGSet
fdr	It can be fdr-stat, fdr-pvalue, vector of new padj

Author(s)

Lorena Pantano

Examples

```
library(DESeq2)
library(dplyr)
dds <- makeExampleDESeqDataSet(betaSD=1)
colData(dds)[["treatment"]] <- sample(colData(dds)[["condition"]], 12)
design(dds) <-  ~ condition + treatment
dds <- DESeq(dds)
res <- degComps(dds, contrast = list("treatment_B_vs_A"))
```

---

degFilter *Filter genes by group*

Description

This function will keep only rows that have a minimum counts of 1 at least in a min number of samples (default 80%).

Usage

```
degFilter(counts, metadata, group, min = 0.8, minreads = 0)
```

Arguments

counts	Matrix with expression data, columns are samples and rows are genes or other feature.
metadata	Data.frame with information about each column in counts matrix. Rownames should match colnames(counts).
group	Character column in metadata used to group samples and applied the cutoff.
min	Percentage value indicating the minimum number of samples in each group that should have more than 0 in count matrix.
minreads	Integer minimum number of reads to consider a feature expressed.

Value

count matrix after filtering genes (features) with not enough expression in any group.

Examples

```
data(humanGender)
library(SummarizedExperiment)
idx <- c(1:10, 75:85)
c <- degFilter(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], "group", min=1)
```

---

degMA *MA-plot from base means and log fold changes*

Description

MA-plot addaptation to show the shrinking effect.

Usage

```
degMA(
  results,
  title = NULL,
  label_points = NULL,
  label_column = "symbol",
  limit = NULL,
  diff = 5,
  raw = FALSE,
  correlation = FALSE
)
```

Arguments

results	DEGSet class.
title	*Optional*. Plot title.
label_points	Optionally label these particular points.
label_column	Match label_points to this column in the results.
limit	Absolute maximum to plot on the log2FoldChange.
diff	Minimum difference between logFoldChange before and after shrinking.
raw	Whether to plot just the unshrunken log2FC.
correlation	Whether to plot the correlation of the two logFCs.

Value

MA-plot ggplot.

Author(s)

Victor Barrera

Rory Kirchner

Lorena Pantano

Examples

```
library(DESeq2)
dds <- makeExampleDESeqDataSet(betaSD=1)
dds <- DESeq(dds)
res <- degComps(dds, contrast = list("condition_B_vs_A"))
degMA(res)
```

---

degMB *Distribution of expression of DE genes compared to the background*

Description

Distribution of expression of DE genes compared to the background

Usage

```
degMB(tags, group, counts, pop = 400)
```

Arguments

tags	List of genes that are DE.
group	Character vector with group name for each sample in the same order than counts column names.
counts	Matrix with counts for each samples and each gene Should be same length than pvalues vector.
pop	number of random samples taken for background comparison

Value

ggplot2 object

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dds <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
dds <- DESeq(dds)
res <- results(dds)
degMB(row.names(res)[1:20], colData(dds)[["group"]],
  counts(dds, normalized = TRUE))
```

degMDS *Plot MDS from normalized count data*

Description

Uses cmdscale to get multidimensional scaling of data matrix, and plot the samples with ggplot2.

Usage

```
degMDS(counts, condition = NULL, k = 2, d = "euclidian", xi = 1, yi = 2)
```

Arguments

counts	matrix samples in columns, features in rows
condition	vector define groups of samples in counts. It has to be same order than the count matrix for columns.
k	integer number of dimensions to get
d	type of distance to use, c("euclidian", "cor").
xi	number of component to plot in x-axis
yi	number of component to plot in y-axis

Value

ggplot2 object

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dse <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
degMDS(counts(dse), condition = colData(dse)[["group"]])
```

degMean *Distribution of pvalues by expression range*

Description

This function plot the p-values distribution colored by the quantiles of the average count data.

Usage

```
degMean(pvalues, counts)
```

Arguments

pvalues	pvalues of DEG analysis.
counts	Matrix with counts for each samples and each gene. row number should be the same length than pvalues vector.

Value

ggplot2 object

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dds <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
dds <- DESeq(dds)
res <- results(dds)
degMean(res[, 4], counts(dds))
```

---

degMerge *Integrate data comming from degPattern into one data object*

Description

The simplest case is if you want to conbine the pattern profile for gene expression data and proteomic data. It will use the first element as the base for the integration. Then, it will loop through clusters and run degPatterns in the second data set to detect patterns that match this one.

Usage

```
degMerge(
  matrix_list,
  cluster_list,
  metadata_list,
  summarize = "group",
  time = "time",
  col = "condition",
  scale = TRUE,
  mapping = NULL
)
```

Arguments

matrix_list	list expression data for each element
cluster_list	list df item from degPattern output

metadata_list	list data.frames from each element with design experiment. Normally colData output
summarize	character column to use to group samples
time	character column to use as x-axes in figures
col	character column to color samples in figures
scale	boolean scale by row expression matrix
mapping	data.frame mapping table in case elements use different ID in the row.names of expression matrix. For instance, when integrating miRNA/mRNA.

Value

A data.frame with information on what genes are in each cluster in all data set, and the correlation value for each pair cluster comparison.

---

degMV *Correlation of the standard desviation and the mean of the abundance of a set of genes.*

Description

Correlation of the standard desviation and the mean of the abundance of a set of genes.

Usage

```
degMV(group, pvalues, counts, sign = 0.01)
```

Arguments

group	Character vector with group name for each sample in the same order than counts column names.
pvalues	pvalues of DEG analysis.
counts	Matrix with counts for each samples and each gene.
sign	Defining the cutoff to label significant features. row number should be the same length than pvalues vector.

Value

ggplot2 object

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dds <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
dds <- DESeq(dds)
res <- results(dds)
degMV(colData(dds)[["group"]],
      res[, 4],
      counts(dds, normalized = TRUE))
```

---

degObj *Create a deg object that can be used to plot expression values at shiny server:runGist(9930881)*

Description

Create a deg object that can be used to plot expression values at shiny server:runGist(9930881)

Usage

```
degObj(counts, design, outfile)
```

Arguments

counts	Output from get_rank function.
design	Colour used for each gene.
outfile	File that will contain the object.

Value

R object to be load into vizExp.

Examples

```
data(humanGender)
library(SummarizedExperiment)
degObj(assays(humanGender)[[1]], colData(humanGender), NULL)
```

degPatterns *Make groups of genes using expression profile.*

Description

Note that this function doesn't calculate significant difference between groups, so the matrix used as input should be already filtered to contain only genes that are significantly different or the most interesting genes to study.

Usage

```
degPatterns(
  ma,
  metadata,
  minc = 15,
  summarize = "merge",
  time = "time",
  col = NULL,
  consensusCluster = FALSE,
  reduce = FALSE,
  cutoff = 0.7,
  scale = TRUE,
  pattern = NULL,
  groupDifference = NULL,
  eachStep = FALSE,
  plot = TRUE,
  fixy = NULL,
  nClusters = NULL,
  skipDendrogram = TRUE
)
```

Arguments

ma	log2 normalized count matrix
metadata	data frame with sample information. Rownames should match ma column names row number should be the same length than p-values vector.
minc	integer minimum number of genes in a group that will be return
summarize	character column name in metadata that will be used to group replicates. If the column doesn't exist it'll merge the time and the col columns, if col doesn't exist it'll use time only. For instance, a merge between summarize and time parameters: control_point0 ... etc
time	character column name in metadata that will be used as variable that changes, normally a time variable.
col	character column name in metadata to separate samples. Normally control/mutant
consensusCluster	Indicates whether using ConsensusClusterPlus or cluster::diana()

reduce	boolean remove genes that are outliers of the cluster distribution. boxplot function is used to flag a gene in any group defined by time and col as outlier and it is removed from the cluster. Not used if consensusCluster is TRUE.
cutoff	This is deprecated.
scale	boolean scale the ma values by row
pattern	numeric vector to be used to find patterns like this from the count matrix. As well, it can be a character indicating the genes inside the count matrix to be used as reference.
groupDifference	Minimum abundance difference between the maximum value and minimum value for each feature. Please, provide the value in the same range than the ma value (if ma is in log2, groupDifference should be inside that range).
eachStep	Whether apply groupDifference at each stem over time variable. **This only work properly for one group with multiple time points**.
plot	boolean plot the clusters found
fixy	vector integers used as ylim in plot
nClusters	an integer scalar or vector with the desired number of groups
skipDendrogram	a boolean to run or not dendextend. Temporary fix to memory issue in linux.

Details

It can work with one or more groups with 2 or more several time points. Before calculating the genes similarity among samples, all samples inside the same time point (time parameter) and group (col parameter) are collapsed together, and the mean value is the representation of the group for the gene abundance. Then, all pair-wise gene expression is calculated using cor.test R function using kendall as the statistical method. A distance matrix is created from those values. After that, cluster::diana() is used for the clustering of gene-gene distance matrix and cut the tree using the divisive coefficient of the clustering, giving as well by diana. Alternatively, if consensusCluster is on, it would use ConsensusClusterPlus to cut the tree in stable clusters. Finally, for each group of genes, only the ones that have genes higher than minc parameter will be added to the figure. The y-axis in the figure is the results of applying scale() R function, what is similar to creating a Z-score where values are centered to the mean and scaled to the standard desviation by each gene.

The different patterns can be merged to get similar ones into only one pattern. The expression correlation of the patterns will be used to decide whether some need to be merged or not.

Value

list wiht two items:

- df is a data.frame with two columns. The first one with genes, the second with the clusters they belong.
- pass is a vector of the clusters that pass the minc cutoff.
- plot ggplot figure.
- hr clustering of the genes in hclust format.

- profile normalized count data used in the plot.
- raw data.frame with gene values summarized by biological replicates and with metadata information attached.
- summarise data.frame with clusters values summarized by group and with the metadata information attached.
- normalized data.frame with the clusters values as used in the plot.
- benchmarking plot showing the different patterns at different values for clustering cuttree function.
- benchmarking_curve plot showing how the numbers of clusters and genes changed at different values for clustering cuttree function.

Examples

```
data(humanGender)
library(SummarizedExperiment)
library(ggplot2)
ma <- assays(humanGender)[[1]][1:100,]
des <- colData(humanGender)
des[["other"]] <- sample(c("a", "b"), 85, replace = TRUE)
res <- degPatterns(ma, des, time="group", col = "other")
# Use the data yourself for custom figures
 ggplot(res[["normalized"]],
        aes(group, value, color = other, fill = other)) +
  geom_boxplot() +
   geom_point(position = position_jitterdodge(dodge.width = 0.9)) +
   # change the method to make it smoother
   geom_smooth(aes(group=other), method = "lm")
```

---

degPCA *smart PCA from count matrix data*

Description

nice plot using ggplot2 from prcomp function

Usage

```
degPCA(
  counts,
  metadata = NULL,
  condition = NULL,
  pc1 = "PC1",
  pc2 = "PC2",
  name = NULL,
  shape = NULL,
  data = FALSE
)
```

Arguments

counts	matrix with count data
metadata	dara.frame with sample information
condition	character column in metadata to use to color samples
pc1	character PC to plot on x-axis
pc2	character PC to plot on y-axis
name	character if given, column in metadata to print label
shape	character if given, column in metadata to shape points
data	Whether return PCA data or just plot the PCA.

Value

if `results <-` used, the function return the output of `prcomp()`.

Author(s)

Lorena Pantano, Rory Kirchner, Michael Steinbaugh

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dse <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
colData(humanGender)[idx,], design=~group)
degPCA(log2(counts(dse)+0.5), colData(dse),
  condition="group", name="group", shape="group")
```

---

degPlot *Plot top genes allowing more variables to color and shape points*

Description

Plot top genes allowing more variables to color and shape points

Usage

```
degPlot(
  dds,
  xs,
  res = NULL,
  n = 9,
  genes = NULL,
  group = NULL,
  batch = NULL,
```

```
  metadata = NULL,
  ann = c("geneID", "symbol"),
  slot = 1L,
  log2 = TRUE,
  xsLab = xs,
  ysLab = "abundance",
  color = "black",
  groupLab = group,
  batchLab = batch,
  sizePoint = 1
)
```

Arguments

dds	DESeq2::DESeqDataSet object or SummarizedExperiment or Matrix or data.frame. In case of a DESeqDataSet object, always the normalized expression will be used from `counts(dds, normalized = TRUE)`.
xs	Character, colname in colData that will be used as X-axes.
res	DESeq2::DESeqResults object.
n	Integer number of genes to plot from the `res` object. It will take the top N using padj values to order the table.
genes	Character of gene names matching rownames of count data.
group	Character, colname in colData to color points and add different lines for each level.
batch	Character, colname in colData to shape points, normally used by batch effect visualization.
metadata	Metadata in case dds is a matrix.
ann	Columns in rowData (if available) used to print gene names. First element in the vector is the column name in rowData that matches the row.names of the `dds` or `count` object. Second element in the vector is the column name in rowData that it will be used as the title for each gene or feature figure.
slot	Name of the slot to use to get count data.
log2	Whether to apply or not log2 transformation.
xsLab	Character, alternative label for x-axis (default: same as xs)
ysLab	Character, alternative label for y-axis..
color	Color to use to plot groups. It can be one color, or a palette compatible with `ggplot2::scale_color_brewer()`.
groupLab	Character, alternative label for group (default: same as group).
batchLab	Character, alternative label for batch (default: same as batch).
sizePoint	Integer, indicates the size of the plotted points (default 1).

Value

ggplot showing the expresison of the genes

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dse <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
dse <- DESeq(dse)
degPlot(dse, genes = rownames(dse)[1:10], xs = "group")
degPlot(dse, genes = rownames(dse)[1:10], xs = "group", color = "orange")
degPlot(dse, genes = rownames(dse)[1:10], xs = "group", group = "group",
        color = "Accent")
```

---

degPlotCluster *Plot clusters from degPattern function output*

Description

This function helps to format the cluster plots from degPatterns(). It allows to control the layers and it returns a ggplot object that can accept more ggplot functions to allow customization.

Usage

```
degPlotCluster(
  table,
  time,
  color = NULL,
  min_genes = 10,
  process = FALSE,
  points = TRUE,
  boxes = TRUE,
  smooth = TRUE,
  lines = TRUE,
  facet = TRUE,
  cluster_column = "cluster",
  prefix_title = "Group: "
)
```

Arguments

table	normalized element from degPatterns() output. It can be a data.frame with the following columns in there: genes, sample, expression, cluster, xaxis_column, color_colu
time	column name to use in the x-axis.
color	column name to use to color and divide the samples.
min_genes	minimum number of genes to be added to the plot.
process	whether to process the table if it is not ready for plotting.
points	Add points to the plot.

boxes	Add boxplot to the plot.
smooth	Add regression line to the plot.
lines	Add gene lines to the plot.
facet	Split figures based on cluster ID.
cluster_column	column name if cluster is in a column with a different name. Usefull, to plot cluster with different cutoffs used when grouping genes from the clustering step.
prefix_title	text to add before the cluster ID in the figure title.

Value

ggplot2 object.

Examples

```
data(humanGender)
library(SummarizedExperiment)
library(ggplot2)
ma <- assays(humanGender)[[1]][1:100,]
des <- colData(humanGender)
des[["other"]] <- sample(c("a", "b"), 85, replace = TRUE)
res <- degPatterns(ma, des, time="group", col = "other", plot = FALSE)
degPlotCluster(res$normalized, "group", "other")
degPlotCluster(res$normalized, "group", "other", lines = FALSE)

library(dplyr)
library(tidyr)
library(tibble)
table <- rownames_to_column(as.data.frame(ma), "genes") %>%
    gather("sample", "expression", -genes) %>%
    right_join(distinct(res$df[,c("genes", "cluster")]),
               by = "genes") %>%
    left_join(rownames_to_column(as.data.frame(des), "sample"),
              by = "sample") %>%
              as.data.frame()
degPlotCluster(table, "group", "other", process = TRUE)
```

---

degPlotWide *Plot selected genes on a wide format*

Description

Plot selected genes on a wide format

Usage

```
degPlotWide(counts, genes, group, metadata = NULL, batch = NULL)
```

Arguments

counts	DESeq2::DESeqDataSet object or expression matrix
genes	character genes to plot.
group	character, colname in colData to color points and add different lines for each level
metadata	data.frame, information for each sample. Not needed if DESeq2::DESeqDataSet given as counts.
batch	character, colname in colData to shape points, normally used by batch effect visualization

Value

ggplot showing the expresison of the genes on the x axis

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dse <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
dse <- DESeq(dse)
degPlotWide(dse, rownames(dse)[1:10], group = "group")
```

---

degQC *Plot main figures showing p-values distribution and mean-variance correlation*

Description

This function joins the output of degMean, degVar and degMV in a single plot. See these functions for further information.

Usage

```
degQC(counts, groups, object = NULL, pvalue = NULL)
```

Arguments

counts	Matrix with counts for each samples and each gene.
groups	Character vector with group name for each sample in the same order than counts column names.
object	DEGSet oobject.
pvalue	pvalues of DEG analysis.

Value

ggplot2 object

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dds <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
dds <- DESeq(dds)
res <- results(dds)
degQC(counts(dds, normalized=TRUE), colData(dds)[["group"]],
  pvalue = res[["pvalue"]])
```

---

degResults *Complete report from DESeq2 analysis*

Description

Complete report from DESeq2 analysis

Usage

```
degResults(
  res = NULL,
  dds,
  rlogMat = NULL,
  name,
  org = NULL,
  FDR = 0.05,
  do_go = FALSE,
  FC = 0.1,
  group = "condition",
  xs = "time",
  path_results = ".",
  contrast = NULL
)
```

Arguments

res	output from DESeq2::results() function.
dds	DESeq2::DESeqDataSet() object.
rlogMat	matrix from DESeq2::rlog() function.
name	string to identify results
org	an organism annotation object, like org.Mm.eg.db. NULL if you want to skip this step.

FDR	int cutoff for false discovery rate.
do_go	boolean if GO enrichment is done.
FC	int cutoff for log2 fold change.
group	string column name in colData(dds) that separates samples in meaninful groups.
xs	string column name in colData(dss) that will be used as X axes in plots (i.e time)
path_results	character path where files are stored. NULL if you don't want to save any file.
contrast	list with character vector indicating the fold change values from different comparisons to add to the output table.

Value

ggplot2 object

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dse <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
dse <- DESeq(dse)
res <- degResults(dds = dse, name = "test", org = NULL,
  do_go = FALSE, group = "group", xs = "group", path_results = NULL)
```

DEGSet *DEGSet*

Description

S4 class to store data from differentially expression analysis. It should be compatible with different package and stores the information in a way the methods will work with all of them.

Usage

```
DEGSet(resList, default)

DEGSet(resList, default)

as.DEGSet(object, ...)

## S4 method for signature 'TopTags'
as.DEGSet(object, default = "raw", extras = NULL)

## S4 method for signature 'data.frame'
as.DEGSet(object, contrast, default = "raw", extras = NULL)

## S4 method for signature 'DESeqResults'
as.DEGSet(object, default = "shrunken", extras = NULL)
```

Arguments

resList	List with results as elements containing log2FoldChange, pvalues and padj as column. Rownames should be feature names. Elements should have names.
default	The name of the element to use by default.
object	Different objects to be transformed to DEGSet when using as.DEGSet.
...	Optional parameters of the generic.
extras	List of extra tables related to the same comparison when using as.DEGSet.
contrast	To name the comparison when using as.DEGSet.

Details

For now supporting only DESeq2::results() output. Use constructor degComps() to create the object.

The list will contain one element for each comparison done. Each element has the following structure:

- DEG table
- Optional table with shrunk Fold Change when it has been done.

To access the raw table use deg(dgs, "raw"), to access the shrunken table use deg(dgs, "shrunken") or just deg(dgs).

Author(s)

Lorena Pantano

Examples

```
library(DESeq2)
library(edgeR)
library(limma)
dds <- makeExampleDESeqDataSet(betaSD = 1)
colData(dds)[["treatment"]] <- sample(colData(dds)[["condition"]], 12)
design(dds) <-  ~ condition + treatment
dds <- DESeq(dds)
res <- degComps(dds, combs = c("condition"))
deg(res)
deg(res, tidy = "tibble")
# From edgeR
dge <- DGEList(counts=counts(dds), group=colData(dds)[["treatment"]])
dge <- estimateCommonDisp(dge)
res <- as.DEGSet(topTags(exactTest(dge)))
# From limma
v <- voom(counts(dds), model.matrix(~treatment, colData(dds)), plot=FALSE)
fit <- lmFit(v)
fit <- eBayes(fit, robust=TRUE)
res <- as.DEGSet(topTable(fit, n = "Inf"), "A_vs_B")
```

degSignature *Plot gene signature for each group and signature*

Description

Given a list of genes beloging to a different classes, like markers, plot for each group, the expression values for all the samples.

Usage

```
degSignature(
  counts,
  signature,
  group = NULL,
  metadata = NULL,
  slot = 1,
  scale = FALSE
)
```

Arguments

counts	expression data. It accepts bcbioRNASeq, DESeqDataSet and SummarizedExperiment. As well, data.frame or matrix is supported, but it requires metadata in that case.
signature	data.frame with two columns: a) genes that match row.names of counts, b) label to classify the gene inside a group. Normally, cell tissue name.
group	character in metadata used to split data into different groups.
metadata	data frame with sample information. Rownames should match ma column names row number should be the same length than p-values vector.
slot	slotName in the case of SummarizedExperiment objects.
scale	Whether to scale or not the expression.

Value

ggplot plot.

Examples

```
data(humanGender)
data(geneInfo)
degSignature(humanGender, geneInfo, group = "group")
```

degSummary *Print Summary Statistics of Alpha Level Cutoffs*

Description

Print Summary Statistics of Alpha Level Cutoffs

Usage

```
degSummary(
  object,
  alpha = c(0.1, 0.05, 0.01),
  contrast = NULL,
  caption = "",
  kable = FALSE
)
```

Arguments

object	Can be DEGSet or DESeqDataSet or DESeqResults.
alpha	Numeric vector of desired alpha cutoffs.
contrast	Character vector to use with `results()` function.
caption	Character vector to add as caption to the table.
kable	Whether return a `knitr::kable()` output. Default is data.frame.

Value

data.frame or `knitr::kable()`.

Author(s)

Lorena Pantano

References

- original idea of multiple alpha values and code syntax from Michael Steinbaugh.

Examples

```
library(DESeq2)
data(humanGender)
idx <- c(1:5, 75:80)
counts <- assays(humanGender)[[1]]
dse <- DESeqDataSetFromMatrix(counts[1:1000, idx],
                              colData(humanGender)[idx,],
                              design = ~group)
dse <- DESeq(dse)
```

```
res1 <- results(dse)
res2 <- degComps(dse, contrast = c("group_Male_vs_Female"))
degSummary(dse, contrast = "group_Male_vs_Female")
degSummary(res1)
degSummary(res1, kable = TRUE)
degSummary(res2[[1]])
```

---

degVar *Distribution of pvalues by standard desviation range*

Description

This function pot the p-valyes distribution colored by the quantiles of the standard desviation of count data.

Usage

```
degVar(pvalues, counts)
```

Arguments

pvalues	pvalues of DEG analysis
counts	Matrix with counts for each samples and each gene. row number should be the same length than pvalues vector.

Value

ggplot2 object

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dds <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
dds <- DESeq(dds)
res <- results(dds)
degVar(res[, 4], counts(dds))
```

degVB — *Distribution of the standard desviation of DE genes compared to the background*

Description

Distribution of the standard desviation of DE genes compared to the background

Usage

```
degVB(tags, group, counts, pop = 400)
```

Arguments

tags	List of genes that are DE.
group	Character vector with group name for each sample in the same order than counts column names.
counts	matrix with counts for each samples and each gene. Should be same length than pvalues vector.
pop	Number of random samples taken for background comparison.

Value

ggplot2 object

Examples

```
data(humanGender)
library(DESeq2)
idx <- c(1:10, 75:85)
dds <- DESeqDataSetFromMatrix(assays(humanGender)[[1]][1:1000, idx],
  colData(humanGender)[idx,], design=~group)
dds <- DESeq(dds)
res <- results(dds)
degVB(row.names(res)[1:20], colData(dds)[["group"]],
  counts(dds, normalized = TRUE))
```

degVolcano — *Create volcano plot from log2FC and adjusted pvalues data frame*

Description

Create volcano plot from log2FC and adjusted pvalues data frame

Usage

```
degVolcano(
  stats,
  side = "both",
  title = "Volcano Plot with Marginal Distributions",
  pval.cutoff = 0.05,
  lfc.cutoff = 1,
  shade.colour = "orange",
  shade.alpha = 0.25,
  point.colour = "gray",
  point.alpha = 0.75,
  point.outline.colour = "darkgray",
  line.colour = "gray",
  plot_text = NULL
)
```

Arguments

stats	data.frame with two columns: logFC and Adjusted.Pvalue
side	plot UP, DOWN or BOTH de-regulated points
title	title for the figure
pval.cutoff	cutoff for the adjusted pvalue. Default 0.05
lfc.cutoff	cutoff for the log2FC. Default 1
shade.colour	background color. Default orange.
shade.alpha	transparency value. Default 0.25
point.colour	colours for points. Default gray
point.alpha	transparency for points. Default 0.75
point.outline.colour	Default darkgray
line.colour	Defaul gray
plot_text	data.frame with three columns: logFC, Pvalue, Gene name

Details

This function was mainly developed by @jnhutchinson.

Value

The function will plot volcano plot together with density of the fold change and p-values on the top and the right side of the volcano plot.

Author(s)

Lorena Pantano, John Hutchinson

Examples

```
library(DESeq2)
dds <- makeExampleDESeqDataSet(betaSD = 1)
dds <- DESeq(dds)
stats <- results(dds)[,c("log2FoldChange", "padj")]
stats[["name"]] <- row.names(stats)
degVolcano(stats, plot_text = stats[1:10,])
```

---

geneInfo *data.frame with chromose information for each gene*

Description

data.frame with chromose information for each gene

Usage

```
data(geneInfo)
```

Format

data.frame

Author(s)

Lorena Pantano, 2014-08-14

Source

biomart

---

geom_cor *Add correlation and p-value to a ggplot2 plot*

Description

`geom_cor` will add the correlatin, method and p-value to the plot automatically guessing the position if nothing else specidfied. family font, size and colour can be used to change the format.

Usage

```
geom_cor(
  mapping = NULL,
  data = NULL,
  method = "spearman",
  xpos = NULL,
  ypos = NULL,
  inherit.aes = TRUE,
  ...
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame., and will be used as the layer data.
method	Method to calculate the correlation. Values are passed to cor.test(). (Spearman, Pearson, Kendall).
xpos	Locate text at that position on the x axis.
ypos	Locate text at that position on the y axis.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders().
...	other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or size = 3. They may also be parameters to the paired geom/stat.

Details

It was integrated after reading this tutorial to extend ggplot2 layers

See Also

ggplot2::layer()

Examples

```
data(humanGender)
library(SummarizedExperiment)
library(ggplot2)
ggplot(as.data.frame(assay(humanGender)[1:1000,]),
       aes(x = NA20502, y = NA20504)) +
  geom_point() +
  ylim(0,1.1e5) +
  geom_cor(method = "kendall", ypos = 1e5)
```

---

humanGender *DGEList object for DE genes betwen Male and Females*

Description

DGEList object for DE genes betwen Male and Females

Usage

```
data(humanGender)
```

Format

DGEList

Author(s)

Lorena Pantano, 2017-08-37

Source

gEUvadis

---

significants *Method to get the significant genes*

Description

Function to get the features that are significant according to some thresholds from a DEGSet, DESeq2::DESeqResults and edgeR::topTags.

Usage

```
significants(object, padj = 0.05, fc = 0, direction = NULL, full = FALSE, ...)

## S4 method for signature 'DEGSet'
significants(object, padj = 0.05, fc = 0, direction = NULL, full = FALSE, ...)

## S4 method for signature 'DESeqResults'
significants(object, padj = 0.05, fc = 0, direction = NULL, full = FALSE, ...)

## S4 method for signature 'TopTags'
significants(object, padj = 0.05, fc = 0, direction = NULL, full = FALSE, ...)

## S4 method for signature 'list'
significants(
  object,
  padj = 0.05,
  fc = 0,
  direction = NULL,
  full = FALSE,
  newFDR = FALSE,
  ...
)
```

Arguments

object	DEGSet
padj	Cutoff for the FDR column.
fc	Cutoff for the log2FC column.
direction	Whether to take down/up/ignore. Valid arguments are down, up and NULL.
full	Whether to return full table or not.
...	Passed to deg. Default: value = NULL. Value can be 'raw', 'shrunken'.
newFDR	Whether to recalculate the FDR or not. See https://support.bioconductor.org/p/104059/#104072. Only used when a list is giving to the method.

Value

a dplyr::tbl_df data frame. gene column has the feature name. In the case of using this method with the results from degComps, log2FoldChange has the higher foldChange from the comparisons, and padj has the padj associated to the previous column. Then, there is two columns for each comparison, one for the log2FoldChange and another for the padj.

Author(s)

Lorena Pantano

Examples

```
library(DESeq2)
dds <- makeExampleDESeqDataSet(betaSD=1)
colData(dds)[["treatment"]] <- sample(colData(dds)[["condition"]], 12)
  design(dds) <-  ~ condition + treatment
dds <- DESeq(dds)
res <- degComps(dds, contrast = list("treatment_B_vs_A",
                                     c("condition", "A", "B")))
significants(res, full = TRUE)
# significants(res, full = TRUE, padj = 1) # all genes
```

Index

aes(), *36*
aes_(), *36*
as.DEGSet *(*DEGSet*)*, 28
as.DEGSet,data.frame-method *(*DEGSet*)*, 28
as.DEGSet,DESeqResults-method *(*DEGSet*)*, 28
as.DEGSet,TopTags-method *(*DEGSet*)*, 28

borders(), *36*

cluster::diana(), *19, 20*
ComplexHeatmap::Heatmap(), *8*
ComplexHeatmap::HeatmapAnnotation(), *6*
ConsensusClusterPlus, *19, 20*
cor.test(), *36*
createReport, 4

data.frame, *31*
deg, 4, *38*
deg,DEGSet-method *(*deg*)*, 4
degCheckFactors, 5
degColors, 6
degComps, 7, *38*
degComps(), *29*
degCorCov, 8
degCorCov(), *10*
degCorrect *(*degDefault*)*, 11
degCorrect,DEGSet-method *(*degDefault*)*, 11
degCovariates, 9
degDefault, 11
degDefault,DEGSet-method *(*degDefault*)*, 11
degFilter, 12
degMA, 13
degMB, 14
degMDS, 15
degMean, 15, *26*
degMerge, 16
degMV, 17, *26*
degObj, 18
degPatterns, *16*, 19
degPatterns(), *24*
degPCA, 21
degPlot, 22
degPlotCluster, 24
degPlotWide, 25
degQC, 26
DEGreport *(*DEGreport-package*)*, 3
DEGreport-package, 3
degResults, 27
DEGSet, *5*, *8*, *12, 13*, *26*, 28, *31*, *37, 38*
DEGSet-class *(*DEGSet*)*, 28
degSignature, 30
degSummary, 31
degVar, *26*, 32
degVB, 33
degVolcano, 33
DESeq2::DESeqDataSet, *7*, *23*, *26*
DESeq2::DESeqDataSet(), *27*
DESeq2::DESeqResults, *23*, *37*
DESeq2::estimateSizeFactorsForMatrix(), *6*
DESeq2::lfcShrink(), *7*
DESeq2::results(), *7*, *27*, *29*
DESeq2::rlog(), *27*
DESeqDataSet, *31*
DESeqResults, *31*
dplyr::tbl_df, *38*

edgeR::topTags, *37*

fortify(), *36*

geneInfo, 35
geom_cor, 35
ggplot, *13*
ggplot(), *36*
ggplot2, *25*, *35*
ggplot2::layer(), *36*

humanGender, 37

knitr::kable(), *31*

layer(), *36*

prcomp(), *22*

results(), *31*

significants, 37
significants,DEGSet-method
(significants), 37
significants,DESeqResults-method
(significants), 37
significants,list-method
(significants), 37
significants,TopTags-method
(significants), 37