Package 'ideal'

October 18, 2017

Type Package

Title Interactive Differential Expression AnaLysis

Version 1.0.0

Date 2017-04-04

Maintainer Federico Marini <marinif@uni-mainz.de>

Description This package provides functions for an Interactive Differential Expression AnaLysis of RNA-sequencing datasets, to extract quickly and effectively information downstream the step of differential expression. A Shiny application encapsulates the whole package.

License MIT + file LICENSE

LazyData TRUE

Depends topGO

Imports DESeq2, SummarizedExperiment, GenomicRanges, IRanges, S4Vectors, ggplot2 (>= 2.0.0), d3heatmap, pheatmap, pcaExplorer, IHW, gplots, UpSetR, goseq, stringr, plyr, dplyr, limma, GOstats, GO.db, AnnotationDbi, shiny (>= 0.12.0), shinydashboard, shinyBS, DT, rentrez, rintrojs, knitr, rmarkdown, shinyAce, BiocParallel, grDevices, methods

Suggests testthat, BiocStyle, airway, org.Hs.eg.db, TxDb.Hsapiens.UCSC.hg38.knownGene, DEFormats, edgeR

URL https://github.com/federicomarini/ideal

BugReports https://github.com/federicomarini/ideal/issues

biocViews GeneExpression, DifferentialExpression, RNASeq, Sequencing, Visualization, QualityControl, GUI, GeneSetEnrichment, ReportWriting

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Federico Marini [aut, cre]

R topics documented:

deseqresult2DEgenes 2
deseqresult2tbl 3
ggplotCounts 3
goseqTable 4
ideal 5
ideal-pkg 6
plot_ma 7
plot_volcano 8
sepguesser 9

Index **11**

`deseqresult2DEgenes` *Generate a tidy table with the DE genes from the results of DESeq*

Description

Generate a tidy table with the DE genes from the results of DESeq

Usage

```
deseqresult2DEgenes(deseqresult, FDR = 0.05)
```

Arguments

`deseqresult`	A `DESeqResults` object
`FDR`	Numeric value, the significance level for thresholding adjusted p-values

Value

A "tidy" data.frame with only genes marked as differentially expressed

Examples

```
# with simulated data...
library(DESeq2)
dds <- DESeq2::makeExampleDESeqDataSet(n=100, m=8, betaSD = 2)
dds <- DESeq(dds)
res <- results(dds)
deseqresult2DEgenes(res)
```

deseqresult2tbl *Generate a tidy table with the results of DESeq*

Description

Generate a tidy table with the results of DESeq

Usage

```
deseqresult2tbl(deseqresult)
```

Arguments

deseqresult	A DESeqResults object

Value

A "tidy" data.frame with all genes

Examples

```
# with simulated data...
library(DESeq2)
dds <- DESeq2::makeExampleDESeqDataSet(n=100, m=8, betaSD = 1)
dds <- DESeq2::DESeq(dds)
res <- DESeq2::results(dds)
deseqresult2tbl(res)
```

ggplotCounts *Plot normalized counts for a gene*

Description

Plot for normalized counts of a single gene, with jittered points superimposed on the boxplot

Usage

```
ggplotCounts(dds, gene, intgroup = "condition", annotation_obj = NULL)
```

Arguments

dds	A DESeqDataSet object.
gene	A character, specifying the name of the gene to plot
intgroup	Interesting groups: a character vector of names in colData(dds) to use for grouping
annotation_obj	A data.frame object, with row.names as gene identifiers (e.g. ENSEMBL ids) and a column, gene_name, containing e.g. HGNC-based gene symbols. Optional.

Details

Note: this function relies on the plotCounts function of DESeq2, therefore pseudocounts of 0.5 are added to each point

Value

An object created by ggplot

Examples

```
library(airway)
data(airway)
airway
dds_airway <- DESeq2::DESeqDataSetFromMatrix(assay(airway),
                                             colData = colData(airway),
                                             design=~cell+dex)
ggplotCounts(dds_airway,
             gene = "ENSG00000103196", # CRISPLD2 in the original publication
             intgroup = "dex")
```

`goseqTable` *Extract functional terms enriched in the DE genes, based on goseq*

Description

A wrapper for extracting functional GO terms enriched in a list of (DE) genes, based on the algorithm and the implementation in the goseq package

Usage

```
goseqTable(de.genes, assayed.genes, genome = "hg38", id = "ensGene",
  testCats = c("GO:BP", "GO:MF", "GO:CC"), FDR_GO_cutoff = 1, nTop = 200,
  orgDbPkg = "org.Hs.eg.db", addGeneToTerms = TRUE)
```

Arguments

`de.genes`	A vector of (differentially expressed) genes
`assayed.genes`	A vector of background genes, e.g. all (expressed) genes in the assays
`genome`	A string identifying the genome that genes refer to, as in the goseq function
`id`	A string identifying the gene identifier used by genes, as in the goseq function
`testCats`	A vector specifying which categories to test for over representation amongst DE genes - can be any combination of "GO:CC", "GO:BP", "GO:MF" & "KEGG"
`FDR_GO_cutoff`	Numeric value for subsetting the results
`nTop`	Number of categories to extract, and optionally process for adding genes to the respective
`orgDbPkg`	Character string, named as the `org.XX.eg.db` package which should be available in Bioconductor
`addGeneToTerms`	Logical, whether to add a column with all genes annotated to each GO term

Details

Note: the feature length retrieval is based on the goseq function, and requires that the corresponding TxDb packages are installed and available

Value

A table containing the computed GO Terms and related enrichment scores

Examples

```
library(airway)
data(airway)
airway
dds_airway <- DESeq2::DESeqDataSetFromMatrix(assay(airway),
                                             colData = colData(airway),
                                             design=~cell+dex)
dds_airway <- DESeq2::DESeq(dds_airway)
res_airway <- DESeq2::results(dds_airway)

res_subset <- deseqresult2DEgenes(res_airway)[1:100,]
myde <- res_subset$id
myassayed <- rownames(res_airway)

## Not run:
mygo <- goseqTable(myde,
                   myassayed,
                   testCats = "GO:BP",
                   addGeneToTerms = FALSE)
head(mygo)

## End(Not run)
```

`ideal` *ideal: Interactive Differential Expression Analysis*

Description

ideal makes differential expression analysis interactive, easy and reproducible. This function launches the main application included in the package.

Usage

```
ideal(dds_obj = NULL, res_obj = NULL, annotation_obj = NULL,
  countmatrix = NULL, expdesign = NULL)
```

Arguments

`dds_obj`	A `DESeqDataSet` object. If not provided, then a `countmatrix` and a `expdesign` need to be provided. If none of the above is provided, it is possible to upload the data during the execution of the Shiny App

res_obj	A DESeqResults object. If not provided, it can be computed during the execution of the application
annotation_obj	A data.frame object, with row.names as gene identifiers (e.g. ENSEMBL ids) and a column, gene_name, containing e.g. HGNC-based gene symbols. If not provided, it can be constructed during the execution via the org.eg.XX.db packages - these need to be installed
countmatrix	A count matrix, with genes as rows and samples as columns. If not provided, it is possible to upload the data during the execution of the Shiny App
expdesign	A data.frame containing the info on the covariates of each sample. If not provided, it is possible to upload the data during the execution of the Shiny App

Value

A Shiny App is launched for interactive data exploration and differential expression analysis

Examples

```
# with simulated data...
library(DESeq2)
dds <- DESeq2::makeExampleDESeqDataSet(n=100, m=8)
cm <- counts(dds)
cd <- colData(dds)

# with the well known airway package...
library(airway)
data(airway)
airway
dds_airway <- DESeq2::DESeqDataSetFromMatrix(assay(airway),
                                             colData = colData(airway),
                                             design=~cell+dex)
## Not run:


ideal()
ideal(dds)
ideal(dds_airway)

dds_airway <- DESeq2::DESeq(dds_airway)
res_airway <- DESeq2::results(dds_airway)
ideal(dds_airway, res_airway)

## End(Not run)
```

---

ideal-pkg *ideal: Interactive Differential Expression Analysis*

Description

ideal makes differential expression analysis interactive, easy and reproducible. The analysis of RNA-seq datasets is guided by the Shiny app as main component of the package, which also provides a wide set of functions to efficiently extract information from the existing data. The app can be also deployed on a Shiny server, to allow its usage without any installation on the user's side.

Details

ideal makes differential expression analysis interactive, easy and reproducible. The analysis of RNA-seq datasets is guided by the Shiny app as main component of the package, which also provides a wide set of functions to efficiently extract information from the existing data. The app can be also deployed on a Shiny server, to allow its usage without any installation on the user's side.

Author(s)

Federico Marini <marinif@uni-mainz.de>, 2016-2017

Maintainer: Federico Marini <marinif@uni-mainz.de>

`plot_ma` *MA-plot from base means and log fold changes*

Description

MA-plot from base means and log fold changes, in the ggplot2 framework, with additional support to annotate genes if provided.

Usage

```
plot_ma(res_obj, FDR = 0.05, point_alpha = 0.2, sig_color = "red",
  annotation_obj = NULL, hlines = NULL, title = NULL,
  xlab = "mean of normalized counts - log10 scale", ylim = NULL,
  add_rug = TRUE, intgenes = NULL, intgenes_color = "steelblue",
  labels_intgenes = TRUE)
```

Arguments

`res_obj`	A `DESeqResults` object
`FDR`	Numeric value, the significance level for thresholding adjusted p-values
`point_alpha`	Alpha transparency value for the points (0 = transparent, 1 = opaque)
`sig_color`	Color to use to mark differentially expressed genes. Defaults to red
`annotation_obj`	A `data.frame` object, with row.names as gene identifiers (e.g. ENSEMBL ids) and a column, `gene_name`, containing e.g. HGNC-based gene symbols. Optional
`hlines`	The y coordinate (in absolute value) where to draw horizontal lines, optional
`title`	A title for the plot, optional
`xlab`	X axis label, defaults to "mean of normalized counts - log10 scale"
`ylim`	Vector of two numeric values, Y axis limits to restrict the view
`add_rug`	Logical, whether to add rug plots in the margins
`intgenes`	Vector of genes of interest. Gene symbols if a `symbol` column is provided in `res_obj`, or else the identifiers specified in the row names
`intgenes_color`	The color to use to mark the genes on the main plot.
`labels_intgenes`	Logical, whether to add the gene identifiers/names close to the marked plots

Details

The genes of interest are to be provided as gene symbols if a `symbol` column is provided in `res_obj`, or else b< using the identifiers specified in the row names

Value

An object created by `ggplot`

Examples

```
library(airway)
data(airway)
airway
dds_airway <- DESeq2::DESeqDataSetFromMatrix(assay(airway),
                                             colData = colData(airway),
                                             design=~cell+dex)
dds_airway <- DESeq2::DESeq(dds_airway)
res_airway <- DESeq2::results(dds_airway)

plot_ma(res_airway, FDR = 0.05, hlines = 1)

plot_ma(res_airway, FDR = 0.1,
        intgenes = c("ENSG00000103196",  # CRISPLD2
                     "ENSG00000120129",  # DUSP1
                     "ENSG00000163884",  # KLF15
                     "ENSG00000179094")  # PER1
       )
```

`plot_volcano` *Volcano plot for log fold changes and log p-values*

Description

Volcano plot for log fold changes and log p-values in the ggplot2 framework, with additional support to annotate genes if provided.

Usage

```
plot_volcano(res_obj, FDR = 0.05, ylim_up = NULL, vlines = NULL,
  title = NULL, intgenes = NULL, intgenes_color = "steelblue",
  labels_intgenes = TRUE)
```

Arguments

`res_obj`	A `DESeqResults` object
`FDR`	Numeric value, the significance level for thresholding adjusted p-values
`ylim_up`	Numeric value, Y axis upper limits to restrict the view
`vlines`	The x coordinate (in absolute value) where to draw vertical lines, optional
`title`	A title for the plot, optional

intgenes	Vector of genes of interest. Gene symbols if a symbol column is provided in res_obj, or else the identifiers specified in the row names
intgenes_color	The color to use to mark the genes on the main plot.
labels_intgenes	Logical, whether to add the gene identifiers/names close to the marked plots

Details

The genes of interest are to be provided as gene symbols if a symbol column is provided in res_obj, or else b< using the identifiers specified in the row names

Value

An object created by ggplot

Examples

```
library(airway)
data(airway)
airway
dds_airway <- DESeq2::DESeqDataSetFromMatrix(assay(airway),
                                             colData = colData(airway),
                                             design=~cell+dex)
dds_airway <- DESeq2::DESeq(dds_airway)
res_airway <- DESeq2::results(dds_airway)

plot_volcano(res_airway)
```

---

sepguesser *Make an educated guess on the separator character*

Description

This function tries to guess which separator was used in a text delimited file

Usage

```
sepguesser(file, sep_list = c(",", "\t", ";", " "))
```

Arguments

file	The name of the file which the data are to be read from
sep_list	A vector containing the candidates for being identified as separators. Defaults to c(",", "\t", ";"," ")

Value

A character value, corresponding to the guessed separator. One of "," (comma), "\t" (tab), ";" (semicolon)," " (whitespace)

Examples

```
sepguesser(system.file("extdata/design_commas.txt",package = "ideal"))
sepguesser(system.file("extdata/design_semicolons.txt",package = "ideal"))
sepguesser(system.file("extdata/design_spaces.txt",package = "ideal"))
mysep <- sepguesser(system.file("extdata/design_tabs.txt",package = "ideal"))

# to be used for reading in the same file, without having to specify the sep
```

Index

DESeqDataSet, *3*, *5*
deseqresult2DEgenes, 2
deseqresult2tbl, 3
DESeqResults, *2, 3*, *6–8*

ggplotCounts, 3
goseq, *4, 5*
goseqTable, 4

ideal, 5
ideal-pkg, 6
ideal-pkg-package *(*ideal-pkg*)*, 6

plot_ma, 7
plot_volcano, 8
plotCounts, *4*

sepguesser, 9