Package: terapadog (via r-universe)

March 12, 2025

Type Package

Title Translational Efficiency Regulation Analysis using the PADOG Method

Version 0.99.6

Description This package performs a Gene Set Analysis with the approach adopted by PADOG on the genes that are reported as translationally regulated (ie. exhibit a significant change in TE) by the DeltaTE package. It can be used on its own to see the impact of translation regulation on gene sets, but it is also integrated as an additional analysis method within ReactomeGSA, where results are further contextualised in terms of pathways and directionality of the change.

License GPL-2

Encoding UTF-8

LazyData FALSE

Imports DESeq2, KEGGREST, stats, utils, dplyr, plotly, htmlwidgets, biomaRt, methods

Suggests apeglm, BiocStyle, knitr, rmarkdown, testthat

Collate preprocessing_helpers.R id_converter.R prepareTerapadogData.R terapadogBricks.R terapadog.R get_FCs.R assign_Regmode.R plotDTA.R

biocViews RiboSeq, Transcriptomics, GeneSetEnrichment, GeneRegulation, Reactome, Software

RoxygenNote 7.3.2

VignetteBuilder knitr

URL https://github.com/Gionmattia/terapadog

BugReports https://github.com/Gionmattia/terapadog/issues

Config/pak/sysreqs make libicu-dev libpng-dev libxml2-dev libssl-dev

Repository https://bioc.r-universe.dev

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

RemoteRef HEAD

RemoteSha edf72cebdb37c76e881ef605276e8a4da7f29c9a

Contents

`get_FCs` *This function execute the Differential Translation Analysis on its own using DeltaTE. The output is a dataframe with the FC in mRNA counts, RIBO counts or TE between the conditions in exam.*

Description

This function execute the Differential Translation Analysis on its own using DeltaTE. The output is a dataframe with the FC in mRNA counts, RIBO counts or TE between the conditions in exam.

Usage

```
get_FCs(expression.data, exp_de, paired = FALSE)
```

Arguments

`expression.data`	A matrix containing the counts from RNA and RIBO samples.
`exp_de`	A dataframe containing information regarding the samples. It has number of rows equal to the columns of esetm.
`paired`	Logical. Default is false. Set to TRUE if the experiment has paired samples in its design.

Value

A dataframe with the results of a Differential Translation Analysis. Each gene's change in RNA counts, RFP(/RIBO) counts and TE are reported, along with the relative adjusted p-values. The RegModes are also reported.

Examples

```
# The execution of a DTA can take some time and computational resources.
# Henceforth, the following code is not supposed to be run from the man page.
# Load the data
rna_file <- system.file("extdata", "rna_counts.tsv",
package = "terapadog")
ribo_file <- system.file("extdata", "ribo_counts.tsv",
package = "terapadog")
sample_file <- system.file("extdata", "sample_info.tsv",
package = "terapadog")
 # Use the paths to load the files.
prepared_data <- prepareTerapadogData(rna_file, ribo_file,
sample_file, "1", "2")
# Unpacks the expression.data and exp_de from the output
expression.data <- prepared_data$expression.data
exp_de <- prepared_data$exp_de
result <- get_FCs(expression.data, exp_de)
```

---

id_converter *Convert the human gene identifier (hgnc_symbol or ensembl_gene_id) to entrezgene_id format for the analysis.*

Description

Convert the human gene identifier (hgnc_symbol or ensembl_gene_id) to entrezgene_id format for the analysis.

Usage

```
id_converter(esetm, id_type, save_report = FALSE, outdir = tempdir())
```

Arguments

esetm	A matrix with the gene count values and whose rownames are the gene Ids (gene symbol or ensembl gene ID).
id_type	A string representing the type of ID given as input. Must be either hgnc_symbol or ensembl_gene_id.
save_report	A boolean. By default, the duplicates report is not saved locally.
outdir	Path to a directory where to save the report. If none is given, a temporary directory will be chosen. No report will be creatted if save_report is set to FALSE.

Value

A matrix with gene IDs in the entrezgene_id format. Also provides a report on the duplicated mappings (conversion_report.txt) in the working dir.

Examples

```
# To showcase thisl internal function, a small example is made.
gene_ids <- c('ENSG00000103197', 'ENSG00000008710', 'ENSG00000167964'
, 'ENSG00000167964')
esetm <- matrix(c(
2.5, 3.1, 5.2, 0.1,
4.1, 2.9, 6.3, 0.5,
1.5, 3.7, 4.8, 0.1), nrow = 4, byrow = FALSE)
rownames(esetm) <- gene_ids
colnames(esetm) <- c("Sample 1", "Sample 2", "Sample 3")
# Call the function
esetm <- id_converter(esetm, "ensembl_gene_id")
print(head(esetm))
```

`plotDTA` *This function will plot an interactive html plot of the results of get_FCs.R That is to say, a plot of the genes undergoing translational regulation, coloured by RegMode. Genes whose RegMode was Undeterminable or Undetermined are omitted.*

Description

This function will plot an interactive html plot of the results of get_FCs.R That is to say, a plot of the genes undergoing translational regulation, coloured by RegMode. Genes whose RegMode was Undeterminable or Undetermined are omitted.

Usage

```
plotDTA(
  FC_results,
  save_plot = FALSE,
  path = file.path(tempdir(), "plot.html")
)
```

Arguments

`FC_results`	A dataframe containing the counts from RNA and RIBO samples.
`save_plot`	Boolean. Default is FALSE. If TRUE, will save plot to a specified directory or a temporary one if none are given.
`path`	A string, pointing to where to save the html plot. If none is given, the plot will be saved to a temporary directory. This parameter will be ignored if save_plot is set to FALSE.

Value

An interactive html plot.

Examples

```
# Creates a mock dataframe for this demonstration
df <- data.frame(
  Identifier = c("Gene A", "Gene B", "Gene C", "Gene D"),
  RegMode = c("Buffered", "Exclusive", "Undeterminable", "No Change"),
  RNA_FC = c(-0.40, -0.5, NA, 0.01),
  RIBO_FC = c(0.19, -0.3, 0.8, -0.02)
)
result <- plotDTA(df)
```

`prepareTerapadogData` *Prepare Data by Loading and Validating RNA, RIBO Counts, and Metadata. This function reads RNA and RIBO count files, checks input data validity and merges them into a single numerical matrix (expression.data). It also prepares the metatadata needed by padog (exp_de).*

Description

Prepare Data by Loading and Validating RNA, RIBO Counts, and Metadata. This function reads RNA and RIBO count files, checks input data validity and merges them into a single numerical matrix (expression.data). It also prepares the metatadata needed by padog (exp_de).

Usage

```
prepareTerapadogData(
  path_to_RNA_counts,
  path_to_RIBO_counts,
  path_to_metadata,
  analysis.group.1,
  analysis.group.2
)
```

Arguments

`path_to_RNA_counts`
: A string representing the file path to the RNA counts data file (.csv or .tsv).

`path_to_RIBO_counts`
: A string representing the file path to the RIBO counts data file (.csv or .tsv).

`path_to_metadata`
: The file path to the metadata file (.csv or .tsv).

`analysis.group.1`
: A string specifying the baseline group in the experiment (e.g., WT, control, etc.).

`analysis.group.2`
: A string specifying the target group to compare against the baseline (e.g., mutant, disease, treatment, etc.).

Value

A list containing two data frames: expression.data and exp_de.

Examples

```
# Data is also available in the "extdata" folder of this package.
# The path will be automatically generated for the purpose of this example
rna_file <- system.file("extdata", "rna_counts.tsv",
package = "terapadog")
ribo_file <- system.file("extdata", "ribo_counts.tsv",
package = "terapadog")
sample_file <- system.file("extdata", "sample_info.tsv",
 package = "terapadog")
 # Use the paths to load the files.
prepared_data <- prepareTerapadogData(rna_file, ribo_file,
sample_file, "1", "2")
# Unpacks the expression.data and exp_de from the output
expression.data <- prepared_data$expression.data
exp_de <- prepared_data$exp_de
# For sake of brevity, only the data frame's head will be printed out
print(head(expression.data))
print(head(exp_de))
```

`terapadog` *Performs the main Gene Set Enrichement Analysis, by applying a modified version of the PADOG algorithm to genes undergoing changes in TE.*

Description

Performs the main Gene Set Enrichement Analysis, by applying a modified version of the PADOG algorithm to genes undergoing changes in TE.

Usage

```
terapadog(
  esetm = NULL,
  exp_de = NULL,
  paired = FALSE,
  gslist = "KEGGRESTpathway",
  organism = "hsa",
  gs.names = NULL,
  NI = 1000,
  Nmin = 3,
  verbose = TRUE
)
```

Arguments

esetm	A matrix containing the counts from RNA and RIBO samples. Rownames must be ensembl GENEIDs, while column names must be sample names. Refer to prepareTerapadogData.R to prepare input data.
exp_de	A dataframe containing information regarding the samples. It has number of rows equal to the columns of esetm. It has a formatted vocabulary, but can be obtained by running prepareTerapadogData.R.
paired	Logical. Specify is the study has a paired design or not. If it does, be sure that the pairs are specified in the "Block" column of the exp_de dataframe.
gslist	A list of named character vectors. Each vector is named after a KEGG pathway ID and each element within the vector is an ENSEMBL gene ID for a gene part of said pathway.
organism	A three letter string giving the name of the organism supported by the "KEGGREST" package.
gs.names	Character vector with the names of the gene sets. If specified, must have the same length as gslist.
NI	Number of iterations allowed to determine the gene set score significance p-values.
Nmin	The minimum size of gene sets to be included in the analysis.
verbose	Logical. If true, shows number of iterations done.

Value

A dataframe with the PADOG score for each pathway in exam.

Index