Package: autoGO (via r-universe)

February 27, 2025

Title Auto-GO: Reproducible, Robust and High Quality Ontology Enrichment Visualizations

Version 1.0.1

Description Auto-GO is a framework that enables automated, high quality Gene Ontology enrichment analysis visualizations. It also features a handy wrapper for Differential Expression analysis around the 'DESeq2' package described in Love et al. (2014) <doi:10.1186/s13059-014-0550-8>. The whole framework is structured in different, independent functions, in order to let the user decide which steps of the analysis to perform and which plot to produce.

License MIT + file LICENSE

Encoding UTF-8

BugReports https://github.com/mpallocc/auto-go/issues

RoxygenNote 7.3.2

LazyData true

Depends R (>= 3.5.0), readr (>= 2.1.2), dplyr (>= 1.1.0), enrichR (>= 3.4)

Imports ape, ComplexHeatmap, DESeq2, dichromat, ggplot2, ggrepel, grDevices, GSVA, imguR, msigdbr, openxlsx, purrr, RColorBrewer, reshape2, stats, stringr, SummarizedExperiment, textshape, tibble, tidyr, tidyselect, utils

Suggests knitr, rlang, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Isabella Grassucci [aut] (<https://orcid.org/0000-0002-2484-8570>), Eleonora Sperandio [aut] (<https://orcid.org/0000-0002-8572-6788>), Fabio Ticconi [cre] (<https://orcid.org/0000-0001-8905-6657>), Alice Massacci [aut] (<https://orcid.org/0000-0001-7780-809X>), Lorenzo D'Ambrosio [aut] (<https://orcid.org/0000-0002-9656-7436>), Matteo Pallocca [aut] (<https://orcid.org/0000-0002-7756-3579>)

Maintainer Fabio Ticconi <fabio.ticconi@gmail.com>

Repository CRAN

Date/Publication 2025-02-27 16:50:06 UTC

Config/pak/sysreqs libmagick++-dev gsfonts libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev perl libx11-dev

Contents

`barplotGO` *barplotGO*

Description

The function barplotGO.R implement the barplot of the first 15 enrichment terms.

For each enrichment result table a barplot is produced. Results are stored in the "enrichment_plots" subfolder for each comparison.

Usage

```
barplotGO(
  enrich_tables,
  title = NULL,
  outfolder = NULL,
  outfile = "barplotGO.png",
  from_autoGO = TRUE
)
```

Arguments

enrich_tables	Dataframe containing the enrichment results or a path to your .tsv file containing the enrichment results. Columns 'Term' and 'Adjusted.P.Value' are required.
title	Default to NULL, only specify if from_autoGO is FALSE. When enrich_tables is not from autoGO and thus from read_enrich_tables, the user can specify title and subtitle of the plot as 2-element character vector, for example: c("This is the title", "this is the subtitle")
outfolder	Default to NULL, only specify if from_autoGO is FALSE. The name to assign to the folder for output saving.
outfile	Default to "barplotGO.png", is ignored if from_autoGO is TRUE. The name of the barplot filename.
from_autoGO	Default is TRUE, set to FALSE if the enrichment tables you want to use are not from a differential expression analysis.

Value

No return value. Files will be produced as part of normal execution.

Examples

```
## Not run:
barplotGO(
  enrich_tables = enrich_tables,
  title = NULL,
  outfolder = NULL,
  outfile = NULL,
  from_autoGO = TRUE
)

## End(Not run)
```

---

choose_database *choose_database*

Description

It allows the user to choose the databases on which to perform the enrichment analysis. Either it returns all the possible databases or a subset of them.

Usage

```
choose_database(db_search = NULL)
```

Arguments

db_search	(Default = NULL), is the string pattern to be matched against the list of enrichR databases. Any matching DBs will be returned.

Value

List of database names as a character vector.

Examples

```
## Not run:
choose_database(db_search = "KEGG")

## End(Not run)
```

`comparisons` *Example comparison data.*

Description

It is the comparison table needed by DESeq2, constituted by the column *treatment* and by the column *control*.

Usage

```
comparisons
```

Format

A data frame with 6 rows and 2 variable:

`counts` *Example raw count data.*

Description

Genes as rows and samples as columns.

Usage

```
counts
```

Format

A data frame with 29744 rows and 19 variable:

deseq_analysis *Differential Gene Expression Analysis*

Description

This function allows to perform a differential gene expression analysis using the DESeq2 package.

The principal DESeq2 workflow is employed. Raw counts are rounded, if it is necessary, because integers are needed for DESeq2 to run. In case the user provides an .rds file, the tool makes sure that in the assay of the SummarizedExperiment only counts are stored. Then the function DESeqDataSetFromMatrix (or DESeqDataSet for .rds) is employed. A prefiltering is applied in order to remove all genes having sum along the subjects less than 10. The Differential Expression Analysis is performed by employing the function DE and the normalized data matrix is stored in the working directory. This will be useful for further analysis and visualizations. Results are extracted for each comparison of interest. Subfolders will be generated in the "outfolder" (default: ./results) with the name of the comparisons made. NOTE: as standard we use the nomenclature "CONTROL_vs_TREATMENT", i.e. the control is on the left. Inside each comparison subfolder we will find a .tsv file with the complete differential analysis and other subfolder based on the pvalue and the log2FC thresholds; inside this we will find a .tsv file with the results of the only filtered genes. This subfolders will be divided in other subfolders as "up_genes", "down_genes" and "up_down_genes". Look the path flow chart at the end of this tutorial (Figure 1).

Usage

```
deseq_analysis(
  counts,
  groups,
  comparisons,
  padj_threshold = 0.05,
  log2FC_threshold = 0,
  pre_filtering = TRUE,
  save_excel = FALSE,
  outfolder = "./results",
  del_csv = ","
)
```

Arguments

counts	The path to raw counts file. Accepted file formats are tab or comma-separated files (.tsv, .csv), .txt files, .rds. Genes must be on rows, samples on columns.
groups	Sample information table needed by DESeq2 (e.g. 'colData'). A data frame with at least two columns: one for samples, one for a grouping variable (See examples).
comparisons	Table of comparisons based on the grouping variable in 'groups' table (See examples). It should be a data.frame with column 'treatment' and column 'control'. It is possible to provide the path to a .txt file.
padj_threshold	Threshold value for adjusted p-value significance (Defaults to 0.05).

log2FC_threshold
: Threshold value for log2(Fold Change) for considering genes as differentially expressed.

pre_filtering
: Removes genes which sum in the raw counts is less than 10 (Default = TRUE).

save_excel
: Allows to save all the output tables in .xlsx format (Default = FALSE).

outfolder
: The name to assign to the folder for output saving. (Default = "./results").

del_csv
: Specify the delimiter of the .csv file, default is ",". This is because opening .csv files with Excel messes up the format and changes the delimiter in ";".

Value

No return value. Files will be produced as part of normal execution.

Examples

```
sample <- c("Pat_1", "Pat_2", "Pat_3", "Pat_4", "Pat_5", "Pat_6")
group <- c("CTRL", "CTRL", "TREAT_A", "TREAT_A", "TREAT_B", "TREAT_B")
groups <- data.frame(sample, group)
treatment <- c("TREAT_A", "TREAT_B", "TREAT_A")
control <- c("CTRL", "CTRL", "TREAT_B")
comparisons <- data.frame(treatment, control)
## Not run:
deseq_analysis(counts,
  groups,
  comparisons,
  padj_threshold = 0.05,
  log2FC_threshold = 0,
  pre_filtering = T,
  save_excel = F,
  outfolder = "./results",
  del_csv = ","
)

## End(Not run)
```

---

filtering_DE *Filtering DESeq2 results*

Description

We could be in a position to carry out multiple filters on the results of the differential analysis, in order not to repeat all the deseq_analysis.R code which provides for the actual computation of the differential analysis, the filtering_DE.R function has been implemented to be able to filter the file(s) " * _allres.tsv " and generate all the folders and files associated with the specific filters applied.

The function automatically searches inside the folders where_results and outfolder the file(s) (See ?deseq_analysis()) "_allres.tsv" and generates folders and files in the same folders with the new filters for foldchange and pvalue respectively.

Usage

```
filtering_DE(
  padj_threshold = 0.05,
  log2FC_threshold = 1,
  outfolder = "./results",
  save_excel = FALSE
)
```

Arguments

padj_threshold	(Default = 0.05) Threshold value for adjusted p-value filtering.
log2FC_threshold	(Default = 0) Threshold value for log2(Fold Change) filtering.
outfolder	(Default = "./results") Name of the folder in which the new output is written.
save_excel	(Default = FALSE) Write output in MS Excel file format (.xlsx).

Value

No return value. Files will be produced as part of normal execution.

Examples

```
## Not run:
filtering_DE(
  padj_threshold = 0.05,
  log2FC_threshold = 1,
  outfolder = "./results",
  save_excel = F
)

## End(Not run)
```

---

groups *Example group data.*

Description

It is the sample information table needed by DESeq2, constituted by the column *sample*, including sample barcodes and by the column *group* including the condition to which each sample belongs.

Usage

```
groups
```

Format

A data frame with 18 rows and 2 variable:

heatmapGO *HeatmapGO*

Description

If the analysis has been performed on more conditions it is interest to have a look at the difference in the enrichment results between the groups. This can be performed by heatmapGO().

The function automatically reads all the enrichment results of the chosen database. A heatmap is produced for each database, all the terms are merged together and a filter is applied as follows: only terms with a significant pvalue (i.e. less than padj_threshold) in at least one comparison will be retained and plotted. These plots will be saved in the "Comparison_Heatmap" folder. In order to have readable plots, if many terms are enriched for a database several images will be created (indexed _1, _2, ...).

Usage

```
heatmapGO(
  db,
  outfolder = "./results",
  log2FC_threshold = 0,
  padj_threshold = 0.05,
  min_term_per_row = 2,
  which_list = c("up_genes", "down_genes", "up_down_genes", "not_from_DE")
)
```

Arguments

db	Database of choice to plot the heatmap. It has to be one for which the enrichment analysis has been performed.
outfolder	The name to assign to the folder for output saving. (Default = "./results"). NOTE: please add "/" at the end.
log2FC_threshold	Threshold value for log2(Fold Change) for considering genes as differentially expressed (Default = 0).
padj_threshold	Threshold value for adjusted p-value significance (Defaults to 0.05).
min_term_per_row	Minimum of comparisons or enriched lists on which a certain term must be significant (Defaults to 2).
which_list	One of c("up_genes", "down_genes","up_down_genes", "not_from_DE"): select data to plot. Respectively, only up regulated genes (up_genes), only down regulated genes ("down_genes"), enrichment on both up and down regulated genes (up_down_genes) or select "not_from_DE" if the enrichment will be made on a list of genes that does not come from a differential expression analysis.

Value

No return value. Files will be produced as part of normal execution.

Examples

```
## Not run:
heatmapGO(
  db = "GO_Biological_Process_2021",
  outfolder = "./results",
  log2FC_threshold = 0,
  padj_threshold = 0.05,
  min_term_per_row = 3,
  which_list = "down_genes"
)

## End(Not run)
```

---

lolliGO *lolliGO*

Description

The function lolliGO.R implement the lollipop plot of the first 20 enrichment terms.

For each enrichment result table a lollipop plot is produced. Results are stored in the "enrichment_plots" subfolder for each comparison.

Usage

```
lolliGO(
  enrich_tables,
  title = NULL,
  outfolder = NULL,
  outfile = "lolliGO.png",
  from_autoGO = TRUE
)
```

Arguments

enrich_tables	Dataframe containing the enrichment results or a path to your .tsv file containing the enrichment results. Columns 'Term' and 'Adjusted.P.Value' are required.
title	Default to NULL, only specify if from_autoGO is FALSE. When enrich_tables is not from autoGO and thus from read_enrich_tables, the user can specify title and subtitle of the plot as 2-element character vector, for example: c("This is the title", "this is the subtitle")
outfolder	Default to NULL, only specify if from_autoGO is FALSE. The name to assign to the folder for output saving.

`outfile` Default to "lolliGO.png", is ignored if from_autoGO is TRUE. The name of the lolli filename.

`from_autoGO` Default is TRUE, set to FALSE if the enrichment tables you want to use are not from a differential expression analysis.

Value

No return value. Files will be produced as part of normal execution.

Examples

```
## Not run:
lolliGO(
  enrich_tables = enrich_tables,
  title = NULL,
  outfolder = NULL,
  outfile = NULL,
  from_autoGO = TRUE
)

## End(Not run)
```

---

read_enrich_tables *Read enrichment results from tables*

Description

Helper function to read all the enrichment results in order to proceed with the visualization in an automated way.

Usage

```
read_enrich_tables(
  enrich_table_path = "./results",
  log2FC_threshold = 0,
  padj_threshold = 0.05,
  which_list = c("up_genes", "down_genes", "up_down_genes", "everything"),
  from_autoGO = TRUE,
  files_format = NULL
)
```

Arguments

`enrich_table_path`
Specify the full path to the folder where enrichment tables have to read from (all fitting files in any subdirectory will be loaded).

`log2FC_threshold`
Threshold value for log2(Fold Change) for considering genes as differentially expressed (default = 0).

padj_threshold	Threshold value for adjusted p-value significance (Defaults to 0.05).
which_list	It can be: "up_genes","down_genes","up_down_genes","everything". Select a list of genes to perform the enrichment. Respectively, only up regulated genes (up_genes), only down regulated genes (down_genes), both up and down regulated genes (up_down_genes), or (everything) allow to load all the three kind of lists separately and it is employed also for lists not from differential analysis.
from_autoGO	Default is TRUE, set to FALSE if the lists you want to upload are not from a differential expression analysis.
files_format	Default is NULL, when from_autoGO = FALSE it is mandatory to provide the extension of the list of genes you want to upload.

Value

List of enrichment tables, each one being a tibble object.

Examples

```
## Not run:
enrich_tables <- read_enrich_tables(
  enrich_table_path = "./results",
  log2FC_threshold = 0,
  padj_threshold = 0.05,
  which_list = "down_genes",
  from_autoGO = T,
  files_format = NULL
)

## End(Not run)
```

---

read_gene_lists *Read all the genes lists*

Description

Function employed for reading all the gene lists on which the enrichment will be performed.

Usage

```
read_gene_lists(
  gene_lists_path = "./results",
  log2FC_threshold = 0,
  padj_threshold = 0.05,
  which_list = c("up_genes", "down_genes", "up_down_genes", "everything"),
  from_autoGO = TRUE,
  files_format = NULL
)
```

Arguments

gene_lists_path	Specify the full path to the folder where the gene lists have to read from (all fitting files in any subdirectory will be loaded).
log2FC_threshold	Threshold value for log2(Fold Change) for considering genes as differentially expressed (default = 0).
padj_threshold	Threshold value for adjusted p-value significance (Defaults to 0.05).
which_list	It can be: "up_genes","down_genes","up_down_genes","everything". Select a list of genes to perform the enrichment. Respectively, both up and down regulated genes (up_down_genes), only up regulated genes (up_genes), only down regulated genes (down_genes), or (everything) allow to load all the three kind of lists separately and it is employed also for lists not from differential analysis..
from_autoGO	Default is TRUE, set FALSE if the gene list you want to upload are not from a differential expression analysis.
files_format	(Default = NULL). When from_autoGO = FALSE it is mandatory to provide the extension of the list of genes to upload.

Value

List of gene lists, each one being a one-dimensional data.frame.

Examples

```
## Not run:
gene_lists <- read_gene_lists(
  gene_lists_path = "./results",
  log2FC_threshold = 0,
  padj_threshold = 0.05,
  which_list = "down_genes",
  from_autoGO = T,
  files_format = NULL
)

## End(Not run)
```

---

ssgsea_wrapper *ssGSEA*

Description

Single-sample Gene Set Enrichment Analysis. This kind of analysis is recommended when there are too few samples. The function will generate an enrichment score for each sample and associated visualizations.

The function ssGSEA is implemented on top of GSVA package. It will produce an heatmap and a violin plot. If chosen, it will produce also the enrichment score table. All the necessary data are already provided with the autoGO package (as the MSigDB gene sets). It is possible to choose whether to perform ssGSEA with all the MSigDB (database) or with a sub-group.

Usage

```
ssgsea_wrapper(
  norm_data = "results/deseq_vst_data.txt",
  gene_id_type = c("gene_symbol", "entrez_gene", "ensembl_gene"),
  write_enrich_tables = FALSE,
  group = NULL,
  outfolder = "./ssgsea",
  full_names = FALSE,
  tpm_norm = FALSE,
  categories = c("C1", "C2", "C3", "C4", "C5", "C6", "C7", "C8", "H")
)
```

Arguments

norm_data	Path of the normalized matrix. By default it is the "deseq_vst_data.txt" computed with the function "deseq_analysis()", but other normalized counts matrix (like TPMs) can be passed to the function. Requirements: first column gene names.
gene_id_type	One of c("gene_symbol", "entrez_gene", "ensembl_gene"). The notation for gene names in the matrix.
write_enrich_tables	Default = FALSE. Set to TRUE to save the matrix with enrichment scores.
group	A matrix for the annotation on the heatmap and for grouping in the statistical analysis. Can be the path to an annotation matrix or dataframe.
outfolder	The name to assign to the folder for output saving. (Default = "ssgsea/"). NOTE: please add "/" at the end.
full_names	Default = FALSE, Terms full names are codified for visualization purposes. Set to TRUE to plot full names instead.
tpm_norm	(Default = FALSE). Set to TRUE to perform a TPM normalization on counts matrix before the analysis.
categories	Default = NULL, all the gene sets are considered. The user can specify one or more categories, like c("C1", "C2", "C3").

---

volcanoplot *Volcano Plot*

Description

Once the differential analysis has been performed, it is possible to visualize the volcano plots employing this function.

The volcano plot is generated by the employment of ggplot2, setting xlimit and ylimit based on the data. If there are genes with pvalue equal to infinity, those are forced to the maximum value of the pvalue. If 'my_comparison' paramater is not provided (default: NULL), the function will extract the name of the first subfolder inside "./results" and use it. The volcano plots are saved in the a subfolder for each comparison (Figure 1).

Usage

```
volcanoplot(
  DE_results,
  my_comparison = NULL,
  highlight_genes = NULL,
  log2FC_thresh = 0,
  padj_thresh = 0.05,
  del_csv = ",",
  outfolder = "./results"
)
```

Arguments

DE_results	DESeq2 results table. Accepts both a file (.tsv, .csv, tab-separated .txt) or a dataframe (see example below). Requires 'log2FoldChange' and 'padj' columns.
my_comparison	The comparison to plot (control_vs_treatment, a_vs_b, ...). If the user is employing the whole workflow use exactly the name of the comparison indicated in the comparison dataframe.
highlight_genes	A (optional) list of genes the user would like to highlight (label) in the volcano plot. It accepts a dataframe, a character vector or a path to a file in .txt format.
log2FC_thresh	Threshold value for log2(Fold Change) to highlight genes as differentially expressed (default = 0).
padj_thresh	Threshold value for adjusted p-value to highlight genes as significant (default = 0.05).
del_csv	Specify the delimiter of the .csv file (default = ","). This is because opening .csv files with Excel messes up the format and changes the delimiter to ";".
outfolder	The name to assign to the folder for output saving. (Default = "./results").

Value

No return value. Files will be produced as part of normal execution.

Examples

```
## Not run:
filename <- "./results/H460.2D_vs_H460.3D.2p/DE_H460.2D_vs_H460.3D.2p_allres.tsv"
volcanoplot(
  DE_results = filename,
  my_comparison = "H460.2D_vs_H460.3D.2p",
  log2FC_thresh = 0,
  padj_thresh = 0.05,
  highlight_genes = NULL,
  del_csv = ",",
  outfolder = "./results"
)

## End(Not run)
```

Index