Package 'GSEAplot'

August 1, 2020

Type Package

Title GSEA Analysis and Plotting Function

Version 0.1.0

Author Sarah Innis, Kelsie Reinaltt

Maintainer <klr5fk@virginia.edu>

Description
This function builds off of R code released by the Broad Institute for GSEA analysis.
It ouputs the important results in a more user friendly manner and generates clearer plots.

License N/A

Depends ggplot2, ggthemes, dplyr, gridExtra

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

---

add_to_database   *Add to Existing Gene Set Database*

Description

The function add_to_database returns a database comprised of an existing database and new data, which are paramters provded by the user. The dataset that results from this function will be formatted to be compatible with GSEA functions.

Usage

```
add_to_database(database = "", addition = "")
```

Arguments

database	Formatted existing database.
addition	Gene sets to be added to existing database.

Value

A formatted database with new sets included.

Examples

```
add_to_database(database=hallmark.gs,addition=new_geneset)

add_to_database(database=custom_db,addition=new_geneset)
```

---

create_geneset_db   *Format New Geneset Database*

Description

The create_geneset_db function allows users to create a new database. Given a list wich meets certain specifications, this function will reformat the data in the list to be saved into a database form compatible with GSEAplot functions.

Usage

```
create_geneset_db(database = "")
```

Arguments

database	A list, with the name of the list being the names of the genesets and elements of the list as the description/source and gene symbols.

Value

Database, saved as formatted_db, which may be saved to the work environment and an used in later GSEA analyses.

Examples

```
create_geneset_db(custom_db)

create_geneset_db(modified_db)
```

---

create_phenoinput	*Create phenoinput*

Description

The create_phenoinput function translates phenotypes into a format computable by the GSEA algorithm given a list of phenotypes.

Usage

```
create_phenoinput(ann = "")
```

Arguments

ann   A list of the phenotypes for a sample. There should be two phenotypes (e.g., female and male).

Value

A list of the phenotypes converted into a GSEAplot computable format.

Examples

```
create_phenoinput(gtex_ann)
```

---

database_key	*View Available Geneset Databases*

Description

The function database_key returns the name of the rda file the user may be interested in given the common name of the geneset database. The function may also return the name of all databases if the input is "all".

Usage

```
database_key(database_name = "")
```

Arguments

database_name   The input for this paramter needs to be "all", or the description of the database of interest.

Value

If the input for database_name is "all", the function will return the descriptons for all databases. If the user inputs the descripton of a database, the function will return the file name of the database of interest.

Examples

```
database_key("all")

database_key("hallmark all")

database_key("motif transcription factor targets")
```

---

get_genesets   *View Gene Sets within Database*

Description

This function returns the names of the sets within a database given the file name for the database.

Usage

```
get_genesets(database_file = "")
```

Arguments

database_file   Name of the rda file for the geneset database within the package.

Value

The names of the genesets within the database.

Examples

```
get_genesets(hallmark.gs)

get_genesets(C3.tft.gs)
```

---

get_genesymbols   *Get Gene Symbols*

Description

The get_genesymbols function returns the gene symbols within each gene set given the file name of the database of interest.

Usage

```
get_genesymbols(database_file = "")
```

Arguments

database_file   Name of the rda file for the geneset database within the package.

Value

The gene symbols within each gene set in a database.

Examples

```
get_genesymbols(hallmark.gs)

get_genesymbols(C3.tft.gs)

get_genesymbols(custom_db)
```

GSEAplots   *Execute GSEA Analysis*

Description

Analyzes genetic expression data and determines whether defined gene sets show statistically significant differences with respect to two phenotypes.

Usage

```
GSEAplots(
  input.ds.name = "",
  input.cls.name = "",
  gene.set.input = "",
  doc.string = "",
  nperm = 1000,
  bar_percent = 0.1,
  gap_percent = 0.1,
  under_percent = 0.1,
  upper_percent = 0.1,
  color_line = "black",
  color_tick = "black",
  abs.val = F
)
```

Arguments

input.ds.name	Name of the gct expression file.
input.cls.name	
	Name of the cls phenotype file.
gene.set.input	
	Name of the geneset file.
doc.string	Name of the output folder for analysis and for naming output files.
nperm	number of permutations.
bar_percent	proportional height of tick mark to window.

`gap_percent`	proportional height between minimum enrichment score and top of tick mark to the window.
`under_percent`	proportional height of white space under tick marks to the window size.
`upper_percent`	proportional height of white space over enrichment graph to the window size.
`color_line`	color of enrichment score line in plot pdf.
`color_tick`	color to tick marks on plot.
`abs.val`	Default is false. Determines whether genes are ranked according to signal to noise or absolute value of signal to noise (when abs.val=T).

Details

GSEA analysis is computed using the Broad Institute's R source code. Genes are ranked according to signal to noise ratio (difference in means/sum of standard deviations for the two phenotypes).

Value

pp - A list pp which includes : plots, gene.set.reference.matrix, gene.set.leading, report1, report2, ES.

plots - A list of ggplot objects, this can act as an input to the function plot.ES() to output a pdf.

gene.set.reference.matrix - A list of each gene set within a gene set database and the gene symbols corresponding to each set.

gene.set.leading - a similar structure to gene.set.reference.matrix but only contains the gene symbols within each gene set that are part of the leading edge set.

report1 - Summary of GSEA analysis data for the first phenotype.

report2 - Summary of GSEA analysis data for the second phenotype.

ES - This object contains the enrichment scores and enrichment tags used to create the plots described earlier. The user can use this information to customize plots as they wish.

References

Subramanian, Tamayo, et al. (2005), PNAS 102, 15545-15550, http://www.broad.mit.edu/gsea/

Examples

```
pp = GSEAplots(input.ds.name=expr.input, input.cls.name=pheno.input,
    gene.set.input=gene.set.input, doc.string="GSEA_plots", nperm=1000,
    abs.val=F, bar_percent=0.1, gap_percent=0.1, under_percent=0.1,
    upper_percent=0.1, color_line="black", color_tick="black")

custom_results= GSEAplots(input.ds.name=expr.input, input.cls.name=pheno.input,
    gene.set.input=gene.set.input, doc.string="custom_results", nperm=1000,
    bar_percent=0.1, gap_percent=0.1, under_percent=0.1, upper_percent=0.1,
    color_line="black", color_tick="black", abs.val=F)
```

GSEA_Source   *GSEA Source Code Functions*

Description

The following functions were used and published by the Broad Institute as a part of an attempt to make Gene Set Enrichment Analysis available in R. These functions are used within the functions written in this package. They are not meant to be used by the user but are made available if they need to be used.

Usage

```
GSEA.GeneRanking(
  A,
  class.labels,
  gene.labels,
  nperm,
  permutation.type = 0,
  sigma.correction = "GeneCluster",
  fraction = 1,
  replace = F,
  reverse.sign = F,
  abs.val = F
)

GSEA.EnrichmentScore(
  gene.list,
  gene.set,
  weighted.score.type = 1,
  correl.vector = NULL
)

OLD.GSEA.EnrichmentScore(gene.list, gene.set)

GSEA.EnrichmentScore2(
  gene.list,
  gene.set,
  weighted.score.type = 1,
  correl.vector = NULL
)

GSEA.HeatMapPlot(
  V,
  row.names = F,
  col.labels,
  col.classes,
  col.names = F,
  main = " ",
  xlab = " ",
  ylab = " "
)
```

```
GSEA.Res2Frame(filename = "NULL")

GSEA.Gct2Frame(filename = "NULL")

GSEA.Gct2Frame2(filename = "NULL")

GSEA.ReadClsFile(file = "NULL")

GSEA.Threshold(V, thres, ceil)

GSEA.VarFilter(V, fold, delta, gene.names = "NULL")

GSEA.NormalizeRows(V)

GSEA.NormalizeCols(V)

GSEA(
  input.ds,
  input.cls,
  gene.ann = "",
  gs.db,
  gs.ann = "",
  output.directory = "",
  output.directory2 = "",
  doc.string = "GSEA.analysis",
  non.interactive.run = F,
  reshuffling.type = "sample.labels",
  nperm = 1000,
  weighted.score.type = 1,
  nom.p.val.threshold = -1,
  fwer.p.val.threshold = -1,
  topgs = 10,
  adjust.FDR.q.val = F,
  reverse.sign = F,
  preproc.type = 0,
  random.seed = 123456,
  perm.type = 0,
  fraction = 1,
  replace = F,
  save.intermediate.results = F,
  OLD.GSEA = F,
  use.fast.enrichment.routine = T,
  abs.val = F
)
```

References

Subramanian, Tamayo, et al. (2005), PNAS 102, 15545-15550, http://www.broad.mit.edu/gsea/

---

plot.ES   *GSEA Enrichment Plots*

Description

The plot.ES function creates a pdf of enrichment plots obtained from the GSEA analysis given a list of ggplot objects and a character by which to name the pdf. The pdf is formatted to contain 4 graphs per page and save to the working directory.

Usage

```
## S3 method for class 'ES'
plot(list.of.plots = "", plotname = "")
```

Arguments

list.of.plots   A list of ggplot objects containing the necessary information for an enrichment plot using ggplot2.

plotname   A character by which to name and save the file with the ES plots.

Value

A pdf containing the enrichment plots for all gene sets (with more than 2 genes in a gene set) in a database, saved as the value given plotname.

Examples

```
plot.ES(list.of.plots=pp$plots,plotname="GSEA_plots")
plot.ES(list.of.plots=analaysis$plots,plotname="results_of_analysis")
```