

Package ‘massiveGST’

March 28, 2022

Type Package

Title Competitive Gene Sets Test with the Mann-Whitney-Wilcoxon Test

Version 1.0

Date 2022-03-22

Maintainer Stefano Maria Pagnotta <pagnotta@unisannio.it>

Description Friendly implementation of the Mann-Whitney-Wilcoxon test for competitive gene set enrichment analysis.

Depends R (>= 4.1.0), formattable (>= 0.2.1), msigdbr (>= 7.4.0), WriteXLS (>= 6.3.0), igraph (>= 1.2.6), visNetwork (>= 2.0.9)

Suggests knitr, rmarkdown

License GPL (>= 3)

URL <<https://github.com/stefanoMP/massiveGST>>, <<http://www.massivegenesetstest.org/>>

VignetteBuilder knitr

NeedsCompilation no

Author Stefano Maria Pagnotta [aut, cre, cph] (<<https://orcid.org/0000-0002-8298-9777>>)

Repository CRAN

Date/Publication 2022-03-28 07:20:02 UTC

R topics documented:

cut_by_logit2NES	2
cut_by_NES	3
cut_by_significance	4
get_geneProfile	6
get_geneSets_from_local_files	7
get_geneSets_from_msigdbr	8
massiveGST	9
plot.mGST	10
pre_ranked_list	12

save_as_tsv	13
save_as_xls	14
summary.mGST	15
write_geneSets_to_gmt	16

Index	18
--------------	-----------

cut_by_logit2NES	<i>Trim the table of results.</i>
------------------	-----------------------------------

Description

This function trims the table of results from massiveGST function retaining the rows with a logit2NES below the specified threshold.

Usage

```
cut_by_logit2NES(ttable, logit2NES_threshold = 0.58)
```

Arguments

ttable	a data frame of "mGST" class coming from massiveGST function.
logit2NES_threshold	a real value

Value

A data frame.

Note

the functions cut_by_NES, cut_by_logit2NES, and cut_by_significance can be nested.

Author(s)

Stefano M. Pagnotta

References

Cerulo, Pagnotta (2021) doi: [10.1101/2021.02.15.431228](https://doi.org/10.1101/2021.02.15.431228)

See Also

[massiveGST](#), [cut_by_NES](#), [cut_by_significance](#),
[summary.mGST](#), [plot.mGST](#)

Examples

```
library(massiveGST)

# get the gene profile
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "pre_ranked_list.txt")
geneProfile <- get_geneProfile(fname)

# get the gene-sets
geneSets <- get_geneSets_from_msigdb(category = "H", what = "gene_symbol")

# run the function
ans <- massiveGST(geneProfile, geneSets, alternative = "two.sided")

head(ans)

cut_by_logit2NES(ans)
cut_by_logit2NES(cut_by_significance(ans))

plot(cut_by_logit2NES(ans))
```

cut_by_NES

Trim the table of results.

Description

This function trims the table of results from massiveGST function retaining the rows with a NES below the specified threshold.

Usage

```
cut_by_NES(ttable, NES_threshold = 0.6)
```

Arguments

ttable a data frame of 'mGST' class coming from massiveGST function.
NES_threshold a real value between 0.0 and 1.

Value

A data frame.

Note

the functions cut_by_NES, cut_by_logit2NES, and cut_by_significance can be nested. In the case the test has alternative = 'two.sided', it is better to use cut_by_logit2NES for a symmetric trim of both directions.

Author(s)

Stefano M. Pagnotta

References

Cerulo, Pagnotta (2021) doi: [10.1101/2021.02.15.431228](https://doi.org/10.1101/2021.02.15.431228)

See Also

[massiveGST](#), [cut_by_logit2NES](#), [cut_by_significance](#), [summary.mGST](#), [plot.mGST](#)

Examples

```
library(massiveGST)

# get the gene profile
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "pre_ranked_list.txt")
geneProfile <- get_geneProfile(fname)

# get the gene-sets
geneSets <- get_geneSets_from_msigdbr(category = "H", what = "gene_symbol")

# run the function
ans <- massiveGST(geneProfile, geneSets, alternative = "greater")

head(ans)
cut_by_NES(ans, NES_threshold = .65)
summary(cut_by_NES(ans, NES_threshold = .65))
```

cut_by_significance *Trim the table of results.*

Description

This function trims the table of results from massiveGST function according to the significance required.

Usage

```
cut_by_significance(ttable,
  level_of_significance = 0.05,
  where = c("BH.value", "bonferroni", "p.value")
)
```

Arguments

`ttable` a data frame of "mGST" class coming from massiveGST function.
`level_of_significance`
a real value between 0.0 and 1.
`where` a character string specifying where the level_of_significance has to be applied to the output; must be one of "p.value", "BH.value" (default), and "bonferroni"

Details

BH.value is the adjustment of p-values according to Benjamini and Hockberg's method; B.value is the adjustment of p-values according to Bonferroni's method.

Value

A data frame.

Note

the functions `cut_by_NES`, `cut_by_logit2NES`, and `cut_by_significance` can be nested.

Author(s)

Stefano M. Pagnotta

References

Cerulo, Pagnotta (2021) doi: [10.1101/2021.02.15.431228](https://doi.org/10.1101/2021.02.15.431228)

See Also

[massiveGST](#), [cut_by_logit2NES](#), [cut_by_NES](#), [summary.mGST](#), [plot.mGST](#)

Examples

```
library(massiveGST)

# get the gene profile
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "pre_ranked_list.txt")
geneProfile <- get_geneProfile(fname)

# get the gene-sets
geneSets <- get_geneSets_from_msigdb(category = "H", what = "gene_symbol")

# run the function
ans <- massiveGST(geneProfile, geneSets, alternative = "two.sided")

head(ans)
cut_by_significance(ans)
```

```
cut_by_significance(ans, level_of_significance = 0.05, where = "p")
cut_by_logit2NES(cut_by_significance(ans))

summary(cut_by_significance(ans, level_of_significance = 0.05, where = "bonferroni"))

plot(cut_by_significance(ans, level_of_significance = 0.05, where = "bonferroni"))
```

get_geneProfile	<i>Load a gene-profile from a txt file.</i>
-----------------	---

Description

Load a gene-profile from a txt file.

Usage

```
get_geneProfile(ffile)
```

Arguments

ffile a character string or a list of a character pointing to a local file

Details

The txt file contains two columns separated by a tabulation. The first column is the gene name (or entrez, ensembl, etc); the second column are the numeric values associated with each gene. The profile do not need to be sorted.

As an example, see the file in /massiveGST/extdata/pre_ranked_list.txt

See the path in the example below.

Value

A named list of numeric values.

Author(s)

Stefano M. Pagnotta

See Also

[pre_ranked_list](#)

Examples

```
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "pre_ranked_list.txt")
fname
geneProfile <- get_geneProfile(fname)
class(geneProfile)
head(geneProfile)
tail(geneProfile)
```

`get_geneSets_from_local_files`

Load the gene-sets collection from local gmt files

Description

Load the gene-sets collection from local gmt files

Usage

```
get_geneSets_from_local_files(ffiles)
```

Arguments

`ffiles` a character string or a list of a character pointing to local files

Value

A vector list of gene-sets

Author(s)

Stefano M. Pagnotta

See Also

[get_geneSets_from_msigdbr](#), [write_geneSets_to_gmt](#)

Examples

```
library(massiveGST)

tmp <- get_geneSets_from_msigdbr(category = "H", what = "gene_symbol")

fname1 <- file.path(tempdir(), "h1.gmt")
write_geneSets_to_gmt(tmp, fileName = fname1)

fname2 <- file.path(tempdir(), "h2.gmt")
write_geneSets_to_gmt(tmp, fileName = fname2)
```

```
# getting one collection
geneSets <- get_geneSets_from_local_files(fname1)
length(geneSets)

# getting two collections
geneSets <- get_geneSets_from_local_files(c(fname1, fname2))
length(geneSets)
```

```
get_geneSets_from_msigdb
```

Get the gene-sets from the msigdb package.

Description

This is a wrapper for extraction a gene-sets collection as a vector list to match the data structure for massiveGST function.

Usage

```
get_geneSets_from_msigdb(category, what, subcategory = NULL, species = "Homo sapiens")
```

Arguments

category	MSigDB collection abbreviation, such as H or C1.
what	a character string specifying the code representation of the genes; must be one of "gene_symbol", "entrez_gene", "ensembl_gene", "human_gene_symbol", "human_entrez_gene", "human_ensembl_gene";
subcategory	MSigDB sub-collection abbreviation, such as CGP or BP; NULL (default)
species	Species name, such as 'Homo sapiens' or 'Mus musculus'.

Value

A vector list of gene-sets

Author(s)

Stefano M. Pagnotta

See Also

[msigdb](#)

Examples

```
library(massiveGST)

# get the gene-sets
geneSets <- get_geneSets_from_msigdb(category = "H", what = "gene_symbol")

class(geneSets)
head(geneSets, 3)
```

massiveGST

massive Gene-Sets Test with Mann-Whitney-Wilcoxon statistics.

Description

Perform a competitive gene set enrichment analysis by applying the Mann-Whitney-Wilcoxon test.

Usage

```
massiveGST(gene_profile, gene_sets,
           cols_to_remove = NULL,
           alternative = c("two.sided", "less", "greater")
           )
```

Arguments

gene_profile	a named list of values; the names have to match the names fo genes in the gene-set.
gene_sets	a character vector of gene-sets
cols_to_remove	a list of colnames to eventually remove from the output
alternative	a character string specifying the alternative hypothesis of the MWW test; must be one of "two.sided" (default), "greater" or "less".

Value

A data frame with columns

size	Original size of the gene-set
actualSize	Size of the gene-set after the match with the gene-profile
NES	(Normalized Enrichment Score) the strength of the association of the gene-set with the gene profile; also the percentile rank of the gene-set in the universe of the genes outside the gene-set.
odd	odd transformation of the NES
logit2NES	logit transformation of the NES
abs_logit2NES	absolute value of the logit2NES in the case of "two.sided" alternative
p.value	p-values associated with the gene-set

BH.value	Benjamini and Hockberg adjustment of the p.values
B.value	Bonferroni adjustment of the p.values
relevance	marginal ordering of the table

Author(s)

Stefano M. Pagnotta

References

Cerulo, Pagnotta (2021) doi: [10.1101/2021.02.15.431228](https://doi.org/10.1101/2021.02.15.431228)

See Also

[summary.mGST](#), [plot.mGST](#), [cut_by_logit2NES](#), [cut_by_NES](#), [cut_by_significance](#)

Examples

```
library(massiveGST)

# get the gene profile
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "pre_ranked_list.txt")
geneProfile <- get_geneProfile(fname)

# get the gene-sets
geneSets <- get_geneSets_from_msigdb(category = "H", what = "gene_symbol")

# run the function
ans <- massiveGST(geneProfile, geneSets, alternative = "two.sided")

ans
```

plot.mGST

Graphical rendering of the enrichment analysis.

Description

This function displays the enrichment analysis results both as a bar-plot and a network of gene-sets.

Usage

```
## S3 method for class 'mGST'
plot(x,
     gene_sets = NULL,
     order_by = "logit2NES",
     top = 30,
```

```

    eps = 0.25,
    as.network = FALSE,
    similarity_threshold = 1/3,
    manipulation = FALSE,
    autoResize = TRUE,
    ...
)

```

Arguments

x	a data structure coming from the massiveGST function
gene_sets	a character vector of gene-sets; mandatory for the network display
order_by	a character string specifying which should be the ordering in the bar-plot; must be one of "relevance", "NES", "logit2NES" (default), "p.value", "BH.value", and "bonferroni". These are the same options of summary.mGST
top	an integer value controlling how many gene-sets have to be displayed in the bar-plot; top = 30 (default)
as.network	a logical value to switch to a network display; as.network = FALSE (default)
similarity_threshold	a real value to cut the similarities between gene-sets below this value; similarity_threshold = 1/3 (default)
eps	a real value between 0.0 and 1.0 controlling the contribution of the Jaccard and overlap similarities to their convex combination; eps = 0.25 (default), see details.
manipulation	a logical value allowing to manipulate the network; manipulation = FALSE (default); see visOptions
autoResize	a logical value allowing to resize the network; autoResize = TRUE (default); see visOptions
...	other graphical parameters

Details

This function display the results of enrichment analysis both as a bar-plot and a network.

The network rendering is with the visNetwork package.

The similarity between the gene-set is computed a convex combination of the Jaccard and overlap similarities. See the reference for further details.

Value

In the case of network display, an object from the visNetwork package.

Author(s)

Stefano M. Pagnotta

References

Cerulo, Pagnotta (2021) doi: [10.1101/2021.02.15.431228](https://doi.org/10.1101/2021.02.15.431228)

See Also

[massiveGST](#), [visNetwork](#), [visOptions](#)

Examples

```
library(massiveGST)

# get the gene profile
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "pre_ranked_list.txt")
geneProfile <- get_geneProfile(fname)

# get the gene-sets
geneSets <- get_geneSets_from_msigdb(category = "H", what = "gene_symbol")

# run the function
ans <- massiveGST(geneProfile, geneSets, alternative = "two.sided")

# to get the bar-plot
plot(cut_by_significance(ans, level_of_significance = 0.01))

# to get the network of the gene-sets
plot(cut_by_significance(ans, level_of_significance = 0.01),
     gene_sets = geneSets, as.network = TRUE)
```

pre_ranked_list

FGFR3-TACC3 fusion positive gene profile

Description

This gene-profile comes from the paper in reference. It compares 9 FGFR3-TACC3 fusion positive samples versus 535 other samples in the GBM study from TCGA (Agilent platform).

Author(s)

Stefano M. Pagnotta

References

Frattini et al. "A metabolic function of FGFR3-TACC3 gene fusions in cancer" *Nature volume 553*, 2018 doi: [10.1038/nature25171](https://doi.org/10.1038/nature25171)

`save_as_tsv`*Save the results in tab-separated value file*

Description

Save the data frame coming from the massiveGST function as tab-separated value.

Usage

```
save_as_tsv(x, file_name = "massiveGST.tsv", sep = "\t", ...)
```

Arguments

<code>x</code>	a data frame of "mGST" class coming from massiveGST function.
<code>file_name</code>	a character value ("massiveGST.tsv" as default)
<code>sep</code>	a character value
<code>...</code>	Arguments to be passed to methods

Value

No return value.

Author(s)

Stefano M. Pagnotta

See Also

[massiveGST](#)

Examples

```
library(massiveGST)

# get the gene profile
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "pre_ranked_list.txt")
geneProfile <- get_geneProfile(fname)

# get the gene-sets
geneSets <- get_geneSets_from_msigdbr(category = "H", what = "gene_symbol")

# run the function
ans <- massiveGST(geneProfile, geneSets, alternative = "two.sided")

# save the results
fname <- file.path(tempdir(), "massiveGST_results.tsv")
```

```
save_as_tsv(ans, file_name = fname)
```

save_as_xls

Save the results in xls file format

Description

Save the data frame coming from the massiveGST function as Excel 2003 (XLS) or Excel 2007 (XLSX) files

Usage

```
save_as_xls(x, file_name = "massiveGST.xls", ...)
```

Arguments

x	a data frame of "mGST" class coming from massiveGST function.
file_name	a character value ("massiveGST.xls" as default)
...	Arguments to be passed to methods

Value

No return value.

Author(s)

Stefano M. Pagnotta

See Also

[WriteXLS](#), [massiveGST](#)

Examples

```
library(massiveGST)

# get the gene profile
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "pre_ranked_list.txt")
geneProfile <- get_geneProfile(fname)

# get the gene-sets
geneSets <- get_geneSets_from_msigdb(category = "H", what = "gene_symbol")

# run the function
ans <- massiveGST(geneProfile, geneSets, alternative = "two.sided")
```

```
# save the results
fname <- file.path(tempdir(), "massiveGST_results.xls")
save_as_xls(ans, file_name = fname)
```

summary.mGST

Generate summary tables

Description

This method handles the result of massiveGST function, to provide views of the table.

Usage

```
## S3 method for class 'mGST'
summary(object,
  cols_to_remove = "link",
  order_by = c("relevance", "NES", "logit2NES", "p.value", "BH.value", "bonferroni"),
  top = NULL,
  as.formattable = FALSE,
  ...
)
```

Arguments

object	a data structure coming from the massiveGST function
cols_to_remove	A character list of the columns to remove from the output.
order_by	a character string specifying which marginal ordering has to be applied to the output; must be one of "relevance" (default), "NES", "logit2NES", "p.value", "BH.value", and "bonferroni"
top	an integer to trim the table to the first 'top' rows.
as.formattable	a logical value (default = FALSE) to provide a formatted output with the help of formattable package.
...	Arguments to be passed to methods

Value

A data frame.

Author(s)

Stefano M. Pagnotta

See Also

[massiveGST](#)

Examples

```
library(massiveGST)

# get the gene profile
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "pre_ranked_list.txt")
geneProfile <- get_geneProfile(fname)

# get the gene-sets
geneSets <- get_geneSets_from_msigdbr(category = "H", what = "gene_symbol")

# run the function
ans <- massiveGST(geneProfile, geneSets, alternative = "two.sided")

summary(ans)
summary(ans, as.formattable = TRUE, order_by = "NES", top = 10)
```

write_geneSets_to_gmt *Save a collection of gene-sets in a .gmt file format.*

Description

Write a collection of gene sets as arranged in this package in a gmt file format.

Usage

```
write_geneSets_to_gmt(gs, fileName)
```

Arguments

gs	a character vector of gene-sets
fileName	a character value; "gene_sets.gmt" (default)

Value

No return value.

Author(s)

Stefano M. Pagnotta

See Also

[get_geneSets_from_msigdbr](#), [get_geneSets_from_local_files](#)

Examples

```
library(massiveGST)

# get the gene-sets
geneSets <- get_geneSets_from_msigdb(category = "H", what = "gene_symbol")

# save the gene-sets
fname <- file.path(tempdir(), "hallmarks.gmt")
write_geneSets_to_gmt(geneSets, fileName = fname)
```

Index

`cut_by_logit2NES`, [2](#), [4](#), [5](#), [10](#)
`cut_by_NES`, [2](#), [3](#), [5](#), [10](#)
`cut_by_significance`, [2](#), [4](#), [4](#), [10](#)

`get_geneProfile`, [6](#)
`get_geneSets_from_local_files`, [7](#), [16](#)
`get_geneSets_from_msigdb`, [7](#), [8](#), [16](#)

`massiveGST`, [2](#), [4](#), [5](#), [9](#), [12–15](#)
`msigdb`, [8](#)

`plot.mGST`, [2](#), [4](#), [5](#), [10](#), [10](#)
`pre_ranked_list`, [6](#), [12](#)

`save_as_tsv`, [13](#)
`save_as_xls`, [14](#)
`summary.mGST`, [2](#), [4](#), [5](#), [10](#), [11](#), [15](#)

`visNetwork`, [12](#)
`visOptions`, [11](#), [12](#)

`write_geneSets_to_gmt`, [7](#), [16](#)
`WriteXLS`, [14](#)