

# Package ‘massiveGST’

July 22, 2025

**Type** Package

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

**Version** 1.2.4

**Date** 2025-03-25

**Maintainer** Stefano Maria Pagnotta <pagnotta@unisannio.it>

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

**Depends** R (>= 4.4), formattable (>= 0.2.1), WriteXLS (>= 6.7.0),  
igraph (>= 2.1.4), visNetwork (>= 2.1.2)

**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] (ORCID:  
<<https://orcid.org/0000-0002-8298-9777>>)

**Repository** CRAN

**Date/Publication** 2025-03-28 12:50:01 UTC

## Contents

cut_by_logit2NES . . . . .	2
cut_by_NES . . . . .	3
cut_by_significance . . . . .	4
geneSets.sim . . . . .	6
get_geneProfile . . . . .	7
get_geneSets_from_local_files . . . . .	8
h.all.v2024.1.Hs.symbols.gmt . . . . .	9
massiveGST . . . . .	9
massiveORT . . . . .	11

plot.mGST . . . . .	12
pre_ranked_list . . . . .	14
save_as_tsv . . . . .	15
save_as_xls . . . . .	16
summary.mGST . . . . .	17
write_geneSets_to_gmt . . . . .	18

<b>Index</b>	<b>20</b>
--------------	-----------

---

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 (2022) [doi:10.3390/e24050739](https://doi.org/10.3390/e24050739)

### 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
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)

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

---

## 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 (2022) [doi:10.3390/e24050739](https://doi.org/10.3390/e24050739)

**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
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)

# 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 (2022) [doi:10.3390/e24050739](https://doi.org/10.3390/e24050739)

**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
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)
# 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"))
```

---

geneSets.sim

*Compute the similarities between a collection of gene sets.*

---

## Description

Compute the similarities between a collection of gene sets using a convex function of the Jaccard and overlap indices.

## Usage

```
geneSets.sim(gs, eps = 0.25)
```

## Arguments

gs	a character vector of gene-sets.
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.

## Details

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

## Value

returns an object of class "dist", where the values are the similarities between gene sets.

## Author(s)

Stefano M. Pagnotta

## References

Cerulo, Pagnotta (2022) [doi:10.3390/e24050739](https://doi.org/10.3390/e24050739)

## See Also

[dist](#)

**Examples**

```
library(massiveGST)

# get the gene-sets
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)[1:5]

# compute the similarities
geneSets.sim(geneSets)
ssim <- geneSets.sim(geneSets)
ssim <- as.matrix(ssim)
diag(ssim) <- 1
ssim
```

---

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**

[write\\_geneSets\\_to\\_gmt](#)

**Examples**

```
library(massiveGST)

# getting one collection
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
length(geneSets <- get_geneSets_from_local_files(fname))
head(geneSets)

# getting two or more collections
geneSets <- get_geneSets_from_local_files(c(fname, fname))
```



```
length(geneSets)
```

---

```
h.all.v2024.1.Hs.symbols.gmt
      hallmark gene sets
```

---

### Description

This is the hallmark gene sets collection from the Molecular Signatures Databas.

### Author(s)

Stefano M. Pagnotta

### References

Liberzon et al. "The Molecular Signatures Database (MSigDB) hallmark gene set collection" *Bioinformatics*, Volume 27, Issue 12, June 2011, Pages 1739-1740 [doi:10.1093/bioinformatics/btr260](https://doi.org/10.1093/bioinformatics/btr260)

---

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

---

### Description

Perform a competitive gene set enrichment analysis by applying the Mann-Withney-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 of 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 (2022) [doi:10.3390/e24050739](https://doi.org/10.3390/e24050739)

**See Also**

[summary.mGST](#), [plot.mGST](#), [cut\\_by\\_logit2NES](#), [cut\\_by\\_NES](#), [cut\\_by\\_significance](#), [hallmark gene sets](#)

**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
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)

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

ans
```

---

massiveORT	<i>A wrapper to fisher.test to get over representation analysis of gene sets.</i>
------------	---

---

### Description

The function massiveORT essentially is a wrapper to the function fisher.test in charge to 1) arrange the input to feed fisher.test in sequence for each gene set, 2) arrange the output in a data frame compatible with the other function of the package, and 3) compute the universe of genes for the analysis.

### Usage

```
massiveORT(gene_list, gene_sets, universe = NULL,
           alternative = c("greater", "less", "two.sided"))
```

### Arguments

gene_list	a list of gene names, or gene ids that have to match the corresponding in the gene-set.
gene_sets	a character vector of gene-sets.
universe	a list of gene, or gene ids, that defines the universe for the analysis (see details); NULL by default.
alternative	a character string specifying the alternative hypothesis of the fisher.test; must be one of "two.sided", "greater" (default) or "less".

### Details

This function allows to define externally or compute the universe of reference of the analysis. By default (universe = NULL), the universe is computed starting from the gene names contributing at least once in each gene set.

### Value

A data frame with columns

universe_size	size of the universe of genes.
geneList_size	size of intersection between the gene list and the universe.
geneSet_size	size of intersection between the gene set and the universe.
geneList_in_GenesSet_size	size of the intersection between the geneList and the geneSet.
odds_ratio	odd ratio coming from the fisher.test
log2_odds_ratio	log2 transformation of odds_ratio.
p.value	p-values associated with the gene-set coming from the fisher.test
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 (2022) [doi:10.3390/e24050739](https://doi.org/10.3390/e24050739)

**See Also**

[fisher.test](#), [cut\\_by\\_significance](#), [hallmark gene sets](#)

**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)
geneList <- names(head(geneProfile, 1000))

# get the gene-sets
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)

# run the function
ans <- massiveORT(geneList, geneSets)
cut_by_significance(ans)

plot(ans, geneSets, as.network = TRUE)
```

---

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 <a href="#">summary.mGST</a>
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) <a href="#">visNetwork::visOptions()</a>
autoResize	a logical value allowing to resize the network; resize = TRUE (default) <a href="#">visNetwork::visOptions()</a>
...	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 (2022) [doi:10.3390/e24050739](https://doi.org/10.3390/e24050739)

**See Also**

[massiveGST](#), [visNetwork::visNetwork\(\)](#), [visNetwork::visOptions\(\)](#), [hallmark gene sets](#)

**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
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)

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

# to get the bar-plot
plot(cut_by_significance(ans, level_of_significance = 0.1))

# to get the network of the gene-sets
plot(cut_by_significance(ans, level_of_significance = 0.1),
     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
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)

# 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::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
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)
```



```
# 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
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)

# 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**

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

**Value**

No return value.

**Author(s)**

Stefano M. Pagnotta

**See Also**

[get\\_geneSets\\_from\\_local\\_files](#)

**Examples**

```
library(massiveGST)

# get the gene-sets
fname <- system.file("extdata", package="massiveGST")
fname <- file.path(fname, "h.all.v2024.1.Hs.symbols.gmt")
geneSets <- get_geneSets_from_local_files(fname)

# 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](#), [12](#)

`dist`, [6](#)

`fisher.test`, [12](#)

`geneSets.sim`, [6](#)  
`get_geneProfile`, [7](#)  
`get_geneSets_from_local_files`, [8](#), [19](#)

`h.all.v2024.1.Hs.symbols.gmt`, [9](#)  
hallmark gene sets, [10](#), [12](#), [14](#)  
hallmark gene sets  
    (`h.all.v2024.1.Hs.symbols.gmt`),  
    [9](#)

`massiveGST`, [2](#), [4](#), [5](#), [9](#), [14–16](#), [18](#)  
`massiveORT`, [11](#)

`plot.mGST`, [2](#), [4](#), [5](#), [10](#), [12](#)  
`pre_ranked_list`, [7](#), [14](#)

`save_as_tsv`, [15](#)  
`save_as_xls`, [16](#)  
`summary.mGST`, [2](#), [4](#), [5](#), [10](#), [13](#), [17](#)

`visNetwork::visNetwork()`, [14](#)  
`visNetwork::visOptions()`, [13](#), [14](#)

`write_geneSets_to_gmt`, [8](#), [18](#)  
`WriteXLS::WriteXLS()`, [16](#)