# Package: yaGST (via r-universe)

May 24, 2026

**Type** Package

**Title** Competitive gene set and regulon tests.

**Version** 2017.08.25

**Date** 2017-08-01

**Author** Stefano M. Pagnotta

**Maintainer** Stefano M. Pagnotta <smpagnotta@gmail.com>

**Description** This is a collection of wrappers to the Wilcoxon test to run competitive gene set and regulon tests.

**License** GPL (>= 3)

**Imports** ggplot2, doParallel

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Depends** doParallel (>= 1.0.10), R (>= 3.0)

**Repository** https://zaoqu-liu.r-universe.dev

**Date/Publication** 2017-11-02 10:44:55 UTC

**RemoteUrl** https://github.com/miccec/yaGST

**RemoteRef** master

**RemoteSha** 56227df3ae183070c9d156af11c306ee799435e6

## Contents

yaGST-package *Competitive gene set and regulon tests.*

## Description

This is a collection of wrappers to the Wilcoxon test to run competitive gene set and regulon tests.

## Details

The DESCRIPTION file: This package was not yet installed at build time.

## Author(s)

Stefano M. Pagnotta Maintainer: Stefano M. Pagnotta <smpagnotta@gmail.com>

eeMWW *eeMWW*

## Description

This function implements the Easy Ensemble, together with the Mann-Witney-Wilcox test, to detect the genes associated with few samples (minority set) being a subset of a larger collection of samples (majority set).

## Usage

```
eeMWW(ddata, minoritySet, runs = 1000)
```

## Arguments

| | |
|---|---|
| ddata | a matrix where the samples are by rows and the features are in the columns. |
| minoritySet | a character vector of the minority set matching some row names of ddata. |
| runs | number of resampling. |

## Details

The EasyEnsemble (EE) resampling scheme is an Undersampling technique aimed to compare few samples (minority set), carrying some phenotype, to a larger collection of samples (majority set) unrelated with the phenotype. We implement the EE with the Mann-Whitney-Wilcoxon test (MWW) to compare the minority set, of dimension m, with a randomly selected collection of 2*m samples from the majority set.

## Value

a named vector of real values.

## Note

We suggest running the function in a parallel setup.

## Author(s)

Stefano M. Pagnotta

## References

Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou - *Exploratory Undersampling for Class-Imbalance Learning* - IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS?PART B: CYBERNETICS, VOL. 39, NO. 2, APRIL 2009

## See Also

mwwGST

## Examples

```
require(yaGST)
nr <- 100; nc <- 1000
# generate a data-matrix with nr samples, and nc features
exprData <- matrix(rpois(nc * nr, 100), nrow = nr, ncol = nc)
colnames(exprData) <- paste0("feat", 1:nc)
rownames(exprData) <- paste0("sam", 1:nr)

# increase the first 3 samples (minority set) of 10% of the original intensity
# of the first 30 features (later the gene-set)
exprData[1, 1:30] <- exprData[1, 1:30]* runif(30, min = 1, max = 1.10)
exprData[2, 1:30] <- exprData[1, 1:30]* runif(30, min = 1, max = 1.10)
exprData[3, 1:30] <- exprData[1, 1:30]* runif(30, min = 1, max = 1.10)
samples_of_interest <- rownames(exprData)[1:3] # minority set

# running in parallel
library(doParallel)
# adjust the number of CPUs as needed
cl <- makePSOCKcluster(3)
clusterApply(cl, floor(runif(length(cl), max = 10000000)), set.seed)
registerDoParallel(cl)
ans_eeMWW <- eeMWW(exprData, samples_of_interest)
stopCluster(cl)

# set the gene-set and run the enrichment analysis
geneSet <- colnames(exprData)[1:30]
(tmp <- mwwGST(ans_eeMWW, geneSet))
plot(tmp, rankedList = ans_eeMWW)
```

gmt2GO *Read a .gmt file and generate a list of gene set sequences.*

## Usage

```
gmt2GO(what)
```

## Arguments

| | |
|---|---|
| what | either a character string naming a .gmt file or a list of a character string naming .gmt files |

## Value

a vector of lists (see GO2gmt)

## Author(s)

Stefano M. Pagnotta

## See Also

GO2gmt, rankedList

## Examples

```
library(yaGST)
data("rankedList")

# create a collection of gene sets
GO <- vector("list", 2)
GO[[1]] <- sample(head(names(rankedList), 5000), 50)
# your reference lik for the gene set
attr(GO[[1]], "link") <- "http://www.enjoy_the_silence.dm"

GO[[2]] <- sample(head(names(rankedList), 5000), 50)
attr(GO[[2]], "link") <- "http://www.imagine.jl"
names(GO) <- c("geneSet_1", "geneSet_2")
GO

# save the collection
GO2gmt(GO, "~/my_GO_collection.gtm")
#########
# load a .gmt file
my_GO_collection <- gmt2GO("~/my_GO_collection.gtm")
summary(my_GO_collection)
head(my_GO_collection$geneSet_1)
attr(my_GO_collection[[1]], "link")
attr(my_GO_collection[[1]], "ontology")
```

GO2gmt *Generate a .gmt file from a list of gene set sequences.*

## Usage

```
GO2gmt(GO_, fileName)
```

## Arguments

| | |
|---|---|
| GO_ | a named vector list. |
| fileName | a character string naming a file |

## Author(s)

Stefano M. Pagnotta

## See Also

gmt2GO, rankedList

## Examples

```
library(yaGST)
data("rankedList")

# create a collection of gene sets
GO <- vector("list", 2)
GO[[1]] <- sample(head(names(rankedList), 5000), 50)
# your reference lik for the gene set
attr(GO[[1]], "link") <- "http://www.enjoy_the_silence.dm"

GO[[2]] <- sample(head(names(rankedList), 5000), 50)
attr(GO[[2]], "link") <- "http://www.imagine.jl"
names(GO) <- c("geneSet_1", "geneSet_2")
GO

# save the collection
GO2gmt(GO, "~/my_GO_collection.gtm")
#########
# load a .gmt file
my_GO_collection <- gmt2GO("~/my_GO_collection.gtm")
summary(my_GO_collection)
head(my_GO_collection$geneSet_1)
attr(my_GO_collection[[1]], "link")
attr(my_GO_collection[[1]], "ontology")
```

mwwExtGST *Competitive Regulon Test*

## Description

Run a competitive test to highlight whether a regulon, with postive and negative gene associated with a transcription factor, is highly ranked in a sequence of gene values.

## Usage

```
mwwExtGST(rankedList, geneSetUp, geneSetDown, minLenGeneSet = 15, moreDetails = FALSE, verbose = TRUE)
```

## Arguments

| | |
|---|---|
| rankedList | numeric vector of data values where the names are the genes names |
| geneSetUp | a character list of genes having a positive association with the transcription factor. |
| geneSetDown | a character list of genes having a negative association with the transcription factor. |
| minLenGeneSet | minimum dimension of the pooled geneSet |
| moreDetails | a logical indicating whether the output includes the rankedList (necessary to plot the enrichment) |
| verbose | a logical indicating to suppress or not the messages; it's TRUE by default. |

## Details

The rankedList has to be a named sequence of values where the genes associated with the phenotype are positvive values, while those not associated are negative. This is necessary because the doubledRankedList is set as c(rankedList, -rankedList).

## Value

| | |
|---|---|
| call | a character string of the call of the function. |
| alternative | a character string describing the alternative hypothesis. |
| originalGeneSetCount | the length of the pooled positive and negative gene-sets. |
| geneSetUp | the same character list given in input. |
| geneSetDown | the same character list given in input. |
| actualGeneSet | the list of pooled positive and negative genes as comes from the intesection between the gene-set and the ranked-list. |
| actualGeneSetCount | the length of the actualGeneSet. |
| doubleRankedList | the doubled ranked-list given in input; this slot is filled whether moreDetails is TRUE. Seeq details |

| | |
|---|---|
| lengthOfRankedList | the length of the ranked-list given in input |
| statistic | the value of the Mann-Whitney-Wilcox test statistic. |
| nes | the value of the normalized enrichment score. |
| pu | is the probability unbalance, i.e. the ratio of nes to 1-nes. |
| log.pu | the log2 transformation of the pu. |
| p.value | the p-value for the test. |

## Note

This function adatpts the enrichment analysis methodology from Lim at el. (2009) to the mwwGST function.

## Author(s)

Stefano M. Pagnotta

## References

Lim, W. K., Lyashenko, E. and Califano, A. - *Master regulators used as breast cancer metastasis classifier.* - Pac Symp Biocomput, 504-515 (2009))

## See Also

rankedList, mwwGST

## Examples

```
library(yaGST)
data("rankedList")
positive_gs <- sample(head(names(rankedList), 10000), 200)
negative_gs <- sample(tail(names(rankedList), 10000), 200)
ans <- mwwExtGST(rankedList, positive_gs, negative_gs, moreDetails = TRUE)
ans
plot(ans)
```

---

# mwwGST *Competitive Gene Set Test*

---

## Description

Run a competitive test to highlight whether a gene set is highly ranked in a sequence of gene values on the genes outside the gene-set.

## Usage

```
mwwGST(rankedList, geneSet, minLenGeneSet = 5, alternative = "greater", moreDetails = FALSE, verbose = T
```

## Arguments

| | |
|---|---|
| rankedList | numeric vector of data values where the names are the genes names |
| geneSet | a character list of genes |
| minLenGeneSet | minimum dimension of the geneSet |
| alternative | a character string specifying the alternative hypothesis ("two.sided", "less", "greater"). |
| moreDetails | a logical indicating whether the output includes the rankedList (necessary to plot the enrichment) |
| verbose | a logical indicating to suppress or not the messages; it's TRUE by default. |

## Value

| | |
|---|---|
| call | a character string of the call of the function. |
| alternative | a character string describing the alternative hypothesis. |
| originalGeneSetCount | the length of the gene-set given in input |
| actualGeneSet | the list of genes as comes from the intesection between the gene-set and the ranked-list. |
| actualGeneSetCount | the length of the actualGeneSet |
| rankedList | the ranked-list given in input; this slot is filled whether moreDetails is TRUE |
| lengthOfRankedList | the length of the ranked-list given in input |
| statistic | the value of the Mann-Whitney-Wilcox test statistic. |
| nes | the value of the normalized enrichment score. |
| pu | is the probability unbalance, i.e. the ratio of nes to 1-nes. |
| log.pu | the log2 transformation of the pu. |
| p.value | the p-value for the test. |

## Author(s)

Stefano M. Pagnotta

## See Also

rankedList, mwwExtGST

## Examples

```
library(yaGST)
data("rankedList")
# generate a random data set of dimension 100
geneSet <- sample(head(names(rankedList), 5000), 100)
ans <- mwwGST(rankedList, geneSet, moreDetails = TRUE)
ans
plot(ans)
```

```
# generate a second gene set
geneSet <- sample(tail(names(rankedList), 5000), 100)
ans <- mwwGST(rankedList, geneSet, moreDetails = TRUE)
plot(ans)
```

---

rankedList *An example of pre-ranked list.*

---

## Usage

```
data("rankedList")
```

## Format

The format is: Named num [1:17814] 4.82 4.33 4.25 4.18 4.09 ... - attr(*, "names")= chr [1:17814] "1.48043767313367" "1.37586701949352" "0.212142344466495" "-0.0291897442883637" ...

## Details

This ordered list of genes from the comparison between G-CIMP-Low versus G-CIMP-High in the GBM.

## References

Ceccarelli at al - *Molecular Profiling Reveals Biologically Discrete Subsets and Pathways of Progression in Diffuse Glioma.* - CELL, Volume 164, Issue 3, p550–563, 28 January 2016)

## Examples

```
library(yaGST)
data(rankedList)
head(rankedList, 10)
tail(rankedList, 10)
fivenum(rankedList)
```

# Index