Package 'GSALightning'

November 12, 2026

Type Package

Title Fast Permutation-based Gene Set Analysis

Version 1.40.0

Date 2015-12-30

Author Billy Heung Wing Chang

Maintainer Billy Heung Wing Chang <billyheungwing@gmail.com>

Depends R (>= 3.3.0)

Imports Matrix, data.table, stats

Description GSALightning provides a fast implementation of permutation-based gene set analysis for two-sample problem. This package is particularly useful when testing simultaneously a large number of gene sets, or when a large number of permutations is necessary for more accurate p-values estimation.

License GPL (>=2)

URL https://github.com/billyhw/GSALightning

BugReports https://github.com/billyhw/GSALightning/issues

Suggests knitr, rmarkdown

VignetteBuilder knitr

biocViews Software, BiologicalQuestion, GeneSetEnrichment, DifferentialExpression, GeneExpression, Transcription

NeedsCompilation no

git_url https://git.bioconductor.org/packages/GSALightning

git_branch RELEASE_3_23

git_last_commit 2604572

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

GSALightning-package 2
expression 3
GSALight 3
permTestLight 6
sampleInfo 7
targetGenes 8
wilcoxTest 9

Index **11**

`GSALightning-package` *Fast Permutation-based Gene Set Analysis*

Description

GSALightning provides a fast implementation of permutation-based gene set analysis for two-sample problem. This package is particularly useful when testing simultaneously a large number of gene sets, or when a large number of permutations is necessary for more accurate p-values estimation.

Details

Index: This package was not yet installed at build time.

Author(s)

Billy Heung Wing Chang

Maintainer: Billy Heung Wing Chang <billyheungwing@gmail.com>

References

BHW Chang and W Tian (2015). GSA-Lightning: Ultra Fast Permutation-based Gene Set Analysis. Bioinformatics. doi: 10.1093/bioinformatics/btw349

B Efron and RJ Tibshirani (2007). "On testing the significance of sets of genes." The annals of applied statistics 1(1):107-129

See Also

`GSALight`, `permTestLight`.

expression *Breast Cancer Data from The Cancer Genome Atlas (TCGA)*

Description

This is a subset of the breast cancer expression data set obtained from The Cancer Genome Atlas (TCGA) consortium and processed by the Pan-Cancer project group. The data was downloaded originally using the ELMER Bioconductor package. The gene names have been converted to gene symbols in this data.

Usage

```
data("expression")
```

Value

A matrix with 909 genes and 1218 subjects, of which 114 are controls.

References

The Cancer Genome Atlas (2012) Comprehensive molecular characterization of human colon and rectal cancer. Nature 487:330-337.

Weinstein, John N., et al. "The cancer genome atlas pan-cancer analysis project." Nature genetics 45.10 (2013): 1113-1120.

Yao, L., et al. "Inferring regulatory element landscapes and transcription factor networks from cancer methylomes." Genome biology 16.1 (2015): 105-105.

Examples

```
data(expression)
str(expression)
```

GSALight *Fast Permutation-based Gene Set Analysis*

Description

GSALight is a fast implementation of two-sample permutation-based gene set analysis. It supports the mean or absolute mean version of the permutation T-test as implemeneted in the GSA package. Restandardization is also supported.

Usage

```
GSALight(eset, fac, gs, nperm = NULL, tests = c('unpaired','paired'), method = c("maxmean","mean", "abs
  minsize = 1, maxsize = Inf, restandardize = TRUE,
  npermBreaks = 2000, rmGSGenes = c('stop', 'gene', 'gs'), verbose = TRUE)
```

Arguments

eset	The expression matrix. Each row is a gene, and each column is a subject/sample. The gene names must be presented as the row names.
fac	Subject labels, for unpaired T-test, either a factor or something that can be coerced into a factor (e.g. 0 and 1, Experiment and Control). For paired T-test, fac must be an integer vector of 1,-1,2,-2,..., where each number represents a pair, and the sign represents the conditions.
gs	The gene sets. Most commonly, gs is a list, where each element (named after the gene set) is a vector of genes belonging to the gene set. It can also be a binary sparse matrix, where each row is a gene set, and each column is a gene. For each row (i.e. a gene set), the row entry is 1 if the corresponding gene belongs to the gene set, and 0 otherwise. Alternatively it can be a data table, where the first column (must be named "geneSet") contains the names of the gene set, and the second column (must be named "gene") are the gene set genes.
nperm	Number of permutations. If unspecified, nperm will be set as the total number of gene sets divided by 0.05 times 2. This should be sufficient to estimate accurate p-values for Bonferroni Correction under significance level alpha = 0.05.
tests	The tests to performed. Can be either the default "unpaired" for unpaired T-tests or "paired" for paired T-tests.
method	The method to combine the T-test statistics for each individual gene into a gene set test statistics. The default "maxmean" uses the maxmean statistics proposed by Efron (2007). Other options are "mean", i.e. the mean of the statistics of the genes inside a gene set, and "absmean", the mean of the absolute value of the statistics.
minsize	Minimum gene set size (default = 1, i.e. even gene set with a single gene is allowed).
maxsize	Maximum gene set size (default = Inf, i.e. no upper limit).
restandardize	Should restandardization be performed? This is typically recommended to avoid excessive number of significant gene sets.
npermBreaks	The batch size. When the number of permutation nperm is large, the permutations are broken into batches, and that permutation are performed with the batches sequentially run. Default is 2000.
rmGSGenes	What to do if there gene sets with genes without expression measurements. Default is 'stop', and will return an error if there are gene set genes with missing expression measurements. If rmGSGenes is "gene", genes with missing expression measurements are removed from the gene sets. If rmGSGenes = 'gs', gene sets with non-measured genes are removed.
verbose	Should the progress be reported? Default = TRUE.

Details

The speed performance of GSALight is sensitive to npermBreaks. Setting npermBreaks small can save memory, but will take longer to run. Setting npermBreaks large can speed up GSALight, but may run into memory issues. If GSALight is running slow, consider increasing npermBreaks. If GSALight is running into memory issues, consider reducing npermBreaks. The default 2000 can typically provide a reasonable balance between speed and memory.

Value

A data frame with the p-values, the q-values (via Benjamini-Hochberg FDR control method), the gene set statistics, and the gene set size. If method = 'mean', then the p-values and q-values for up-regulation and down-regulation are reported. Method = 'absmean' corresponds to a two-sided test of absolute changes in expression, hence only one set of p-values and one set of q-values will be reported.

Author(s)

Billy Heung Wing Chang

References

BHW Chang and W Tian (2015). GSA-Lightning: Ultra Fast Permutation-based Gene Set Analysis. Bioinformatics. doi: 10.1093/bioinformatics/btw349

B Efron and RJ Tibshirani (2007). "On testing the significance of sets of genes." The annals of applied statistics 1(1):107-129

See Also

permTestLight, targetGenes

Examples

```
# see the vignette for more examples
# this example is adapted from R GSA package (Efron 2007)

set.seed(100)
x <- matrix(rnorm(1000*20),ncol=20)
rownames(x) <- paste("g",1:1000,sep="")
dd <- sample(1:1000,size=100)

u <- matrix(2*rnorm(100),ncol=10,nrow=100)
x[dd,11:20] <- x[dd,11:20]+u
y <- factor(c(rep('Control',10),rep('Experiment',10)))

#create some random gene sets
genesets=vector("list",50)
for(i in 1:50){
 genesets[[i]]=paste("g",sample(1:1000,size=30),sep="")
}
names(genesets)=paste("set",as.character(1:50),sep="")

GSAmaxmean <- GSALight(x, y, genesets, nperm = 1000, method = 'maxmean', restandardize = FALSE, rmGSGenes = 'gene')
GSAmean <- GSALight(x, y, genesets, nperm = 1000, method = 'mean', restandardize = FALSE, rmGSGenes = 'gene')
GSAabs <- GSALight(x, y, genesets, nperm = 1000, method = 'absmean', restandardize = FALSE, rmGSGenes = 'gene')

head(GSAmaxmean)
head(GSAmean)
head(GSAabs)
```

permTestLight *Fast Single-Gene Permutation Test*

Description

A fast permutation-testing procedure for two-sample single-gene differential expression analysis.

Usage

```
permTestLight(eset, fac, nperm, tests = c('unpaired','paired'), method = c("mean", "absmean"),
       npermBreaks = 2000, verbose = TRUE)
```

Arguments

eset	The expression matrix. Each row is a gene, and each column is a subject/sample. The gene names must be presented as the row names.
fac	Subject labels, for unpaired T-test, either a factor or something that can be coerced into a factor (e.g. 0 and 1, Experiment and Control). For paired T-test, fac must be an integer vector of 1,-1,2,-2,..., where each number represents a pair, and the sign represents the conditions.
nperm	Number of permutations. If unspecified, nperm will be set as the total number of gene sets divided by 0.05 times 2. This should be sufficient to estimate accurate p-values for Bonferroni Correction under significance level alpha = 0.05.
tests	The tests to performed. Can be either the default "unpaired" for unpaired T-tests or "paired" for paired T-tests.
method	Modification of the T-test statistics for each individual gene for hypothesis testing. The default "mean" option uses the typical T-statistics. The other option "absmean" uses the absolute value of the T-statistics (which results in a two-sided test).
npermBreaks	The batch size. When the number of permutation nperm is large, the permutations are broken into batches, and that permutation are performed with the batches sequentially run. Default is 2000.
verbose	Should the progress be reported? Default = TRUE.

Details

The speed performance is sensitive to npermBreaks. Setting npermBreaks small can save memory, but will take longer to run. Setting npermBreaks large can speed up the process, but may run into memory issues. If the function is running slow, consider increase npermBreaks. If the function is running into memory issues, consider reducing npermBreaks. The default 2000 typically can provide a reasonable balance between speed and memory.

Value

A data frame with the p-values, the q-values (via Benjamini-Hochberg FDR control method), the gene statistics, and the gene set size. If method = 'mean', then the p-values and q-values for up-regulation and down-regulation are reported. Method = 'absmean' corresponds to a two-sided test of absolute changes in expression, hence only one set of p-values and one set of q-values will be reported.

Author(s)

Billy Heung Wing Chang

References

BHW Chang and W Tian (2015). GSA-Lightning: Ultra Fast Permutation-based Gene Set Analysis. Bioinformatics. doi: 10.1093/bioinformatics/btw349

See Also

GSALight

Examples

```
# see the vignette for more examples
# this example is adapted from R GSA package (Efron 2007)

set.seed(100)
x <- matrix(rnorm(1000*20),ncol=20)
rownames(x) <- paste("g",1:1000,sep="")
dd <- sample(1:1000,size=100)

u <- matrix(2*rnorm(100),ncol=10,nrow=100)
x[dd,11:20] <- x[dd,11:20]+u
y <- factor(c(rep('Control',10),rep('Experiment',10)))

results <- permTestLight(x, y, nperm = 1000, method = 'mean')
head(results)
```

---

sampleInfo *Sample Information for the Breast Cancer Data from The Cancer Genome Atlas (TCGA)*

Description

This is the subject information data for the breast cancer expression data set obtained from The Cancer Genome Atlas (TCGA) consortium, and processed by the Pan-Cancer project group. The data was downloaded originally using the ELMER Bioconductor package.

Usage

```
data("sampleInfo")
```

Value

A data frame. The element "TN" is the subject label ("Experiment" = cancer patients, and "Control" = the control subjects).

References

The Cancer Genome Atlas (2012) Comprehensive molecular characterization of human colon and rectal cancer. Nature 487:330-337.

Weinstein, John N., et al. "The cancer genome atlas pan-cancer analysis project." Nature genetics 45.10 (2013): 1113-1120.

Yao, L., et al. "Inferring regulatory element landscapes and transcription factor networks from cancer methylomes." Genome biology 16.1 (2015): 105-105.

Examples

```
data(sampleInfo)
str(sampleInfo)
```

---

targetGenes *Target Genes of Distal Regulatory Elements*

Description

A list containing the target genes of 104636 distal regulatory elements from the human genome. The original list is available in the supplementary data of the reference stated below. The gene names has been tranformed to gene symbol already.

Usage

```
data("targetGenes")
```

Value

A list of distal regulatory elements and their target genes.

Source

http://nar.oxfordjournals.org/content/early/2013/09/03/nar.gkt785/suppl/DC1

References

Lu, Yulan, Yuanpeng Zhou, and Weidong Tian. "Combining Hi-C data with phylogenetic correlation to predict the target genes of distal regulatory elements in human genome." Nucleic acids research (2013): gkt785.

Examples

```
data(targetGenes)
str(targetGenes)
```

`wilcoxTest` *Single-Gene Mann Whitney Wilcoxon Test*

Description

A two-sample single-gene differential expression analysis using the Mann Whitney Wilcoxon Test.

Usage

```
wilcoxTest(eset, fac, tests = c("unpaired", "paired"))
```

Arguments

`eset`	The expression matrix. Each row is a gene, and each column is a subject/sample. The gene names must be presented as the row names.
`fac`	Subject labels, for unpaired T-test, either a factor or something that can be coerced into a factor (e.g. 0 and 1, Experiment and Control). For paired T-test, fac must be an integer vector of 1,-1,2,-2,..., where each number represents a pair, and the sign represents the conditions.
`tests`	The tests to performed. Can be either the default "unpaired" for unpaired T-tests or "paired" for paired T-tests.

Details

This function performs Mann Whitney Wilcoxon test (a.k.a Mann Whitney U test and Wilcoxon Rank Sum test) for all genes in eset. This function is built on the wilcox.test() function in the "stats" package, but is structured to align with the usage of permTestLight. We included this function in case users may want to compare the permutation test results with the results of a standard, non-permutation-based nonparametric test.

Value

A data frame with the p-values, the q-values (via Benjamini-Hochberg FDR control method). The p-values and q-values for up-regulation and down-regulation are reported.

Author(s)

Billy Heung Wing Chang

See Also

`permTestLight`, `wilcox.test`

Examples

```
# see the vignette for more examples
# this example is adapted from R GSA package (Efron 2007)

### NOT RUN ###
set.seed(100)
x <- matrix(rnorm(1000*20),ncol=20)
rownames(x) <- paste("g",1:1000,sep="")
dd <- sample(1:1000,size=100)

u <- matrix(2*rnorm(100),ncol=10,nrow=100)
x[dd,11:20] <- x[dd,11:20]+u
y <- factor(c(rep('Control',10),rep('Experiment',10)))

results <- wilcoxTest(x, y, tests = "unpaired")
head(results)
```

Index

∗ **package**
GSALightning-package, 2

expression, 3

GSALight, *2*, 3, *7*
GSALightning *(*GSALightning-package*)*, 2
GSALightning-package, 2

permTestLight, *2*, *5*, 6, *9*

sampleInfo, 7

targetGenes, *5*, 8

wilcox.test, *9*
wilcoxTest, 9