Package ‘GSALightning’

July 21, 2023

Type Package
Title Fast Permutation-based Gene Set Analysis
Version 1.28.0
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_17
git_last_commit dcbc41f
git_last_commit_date 2023-04-25
Date/Publication 2023-07-21
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


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**

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

**Value**

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

**References**


**Examples**

```r
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 implemented in the GSA package. Restandardization is also supported.

**Usage**

```r
GSALight(eset, fac, gs, nperm = NULL, tests = c('unpaired','paired'), method = c("maxmean","mean", "absmean"), 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**


**See Also**

`permTestLight`, `targetGenes`

**Examples**

```r
# 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)
y <- factor(c(rep('Control',10),rep('Experiment',10)))

# create some random gene sets
genesisets=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="")

GSAmxmean <- 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(GSAmxmean)
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**

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

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eset</td>
<td>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.</td>
</tr>
<tr>
<td>fac</td>
<td>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.</td>
</tr>
<tr>
<td>nperm</td>
<td>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.</td>
</tr>
<tr>
<td>tests</td>
<td>The tests to performed. Can be either the default &quot;unpaired&quot; for unpaired T-tests or &quot;paired&quot; for paired T-tests.</td>
</tr>
<tr>
<td>method</td>
<td>Modification of the T-test statistics for each individual gene for hypothesis testing. The default &quot;mean&quot; option uses the typical T-statistics. The other option &quot;absmean&quot; uses the absolute value of the T-statistics (which results in a two-sided test).</td>
</tr>
<tr>
<td>npermBreaks</td>
<td>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.</td>
</tr>
<tr>
<td>verbose</td>
<td>Should the progress be reported? Default = TRUE.</td>
</tr>
</tbody>
</table>

**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.
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


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

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)

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