Package 'fgsea'

April 12, 2022

Title Fast Gene Set Enrichment Analysis

Version 1.20.0

Description The package implements an algorithm for fast gene set enrichment analysis. Using the fast algorithm allows to make more permutations and get more fine grained p-values, which allows to use accurate stantard approaches to multiple hypothesis correction.

biocViews GeneExpression, DifferentialExpression, GeneSetEnrichment, Pathways

SystemRequirements C++11

Depends R (>= 3.3)

Imports Rcpp, data.table, BiocParallel, stats, ggplot2 (>= 2.2.0), gridExtra, grid, fastmatch, Matrix, utils

Suggests testthat, knitr, rmarkdown, reactome.db, AnnotationDbi, parallel, org.Mm.eg.db, limma, GEOquery

License MIT + file LICENCE

LazyData true

LinkingTo Rcpp, BH

RoxygenNote 7.1.1

Encoding UTF-8

VignetteBuilder knitr

URL https://github.com/ctlab/fgsea/

BugReports https://github.com/ctlab/fgsea/issues

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

git_branch RELEASE_3_14

git_last_commit b704f81

git_last_commit_date 2021-10-26

Date/Publication 2022-04-12

Author Gennady Korotkevich [aut],
Vladimir Sukhov [aut],
Nikolay Budin [ctb],
Alexey Sergushichev [aut, cre]

Maintainer Alexey Sergushichev <alsergbox@gmail.com>

R topics documented:

`calcGseaStat` *Calculates GSEA statistics for a given query gene set*

Description

Takes *O(k log k)* time, where *k* is a size of 'selectedSize'.

Usage

```
calcGseaStat(
  stats,
  selectedStats,
  gseaParam = 1,
  returnAllExtremes = FALSE,
  returnLeadingEdge = FALSE,
  scoreType = c("std", "pos", "neg")
)
```

Arguments

stats	Named numeric vector with gene-level statistics sorted in decreasing order (order is not checked).
selectedStats	Indexes of selected genes in the ‘stats‘ array.
gseaParam	GSEA weight parameter (0 is unweighted, suggested value is 1).
returnAllExtremes	If TRUE return not only the most extreme point, but all of them. Can be used for enrichment plot
returnLeadingEdge	If TRUE return also leading edge genes.
scoreType	This parameter defines the GSEA score type. Possible options are ("std", "pos", "neg")

Value

Value of GSEA statistic if both returnAllExtremes and returnLeadingEdge are FALSE. Otherwise returns list with the folowing elements:

- res – value of GSEA statistic
- tops – vector of top peak values of cumulative enrichment statistic for each gene;
- bottoms – vector of bottom peak values of cumulative enrichment statistic for each gene;
- leadingGene – vector with indexes of leading edge genes that drive the enrichment, see http://software.broadinstitute.org/gsea/doc/GSEAUserGuideTEXT.htm#_Running_a_Leading.

Examples

```
data(exampleRanks)
data(examplePathways)
ranks <- sort(exampleRanks, decreasing=TRUE)
es <- calcGseaStat(ranks, na.omit(match(examplePathways[[1]], names(ranks))))
```

---

calcGseaStatBatchCpp *Calculates GSEA statistic valus for all gene sets in ‘selectedStats‘ list.*

Description

Takes *O(n + mKlogK)* time, where n is the number of genes, m is the number of gene sets, and k is the mean gene set size.

Usage

```
calcGseaStatBatchCpp(stats, selectedGenes, geneRanks)
```

Arguments

stats	Numeric vector of gene-level statistics sorted in decreasing order
selectedGenes	List of integer vector with integer gene IDs (from 1 to n)
geneRanks	Integer vector of gene ranks

Value

Numeric vector of GSEA statistics of the same length as ‘selectedGenes‘ list

---

collapsePathways *Collapse list of enriched pathways to independent ones.*

Description

Collapse list of enriched pathways to independent ones.

Usage

```
collapsePathways(
  fgseaRes,
  pathways,
  stats,
  pval.threshold = 0.05,
  nperm = 10/pval.threshold,
  gseaParam = 1
)
```

Arguments

fgseaRes	Table with results of running fgsea(), should be filtered by p-value, for example by selecting ones with padj < 0.01.
pathways	List of pathways, should contain all the pathways present in ‘fgseaRes‘.
stats	Gene-level statistic values used for ranking, the same as in ‘fgsea()‘.
pval.threshold	Two pathways are considered dependent when p-value of enrichment of one pathways on background of another is greater then ‘pval.threshold‘.
nperm	Number of permutations to test for independence, should be several times greater than ‘1/pval.threhold‘. Default value: ‘10/pval.threshold‘.
gseaParam	GSEA parameter, same as for ‘fgsea()‘

Value

Named list with two elments: ‘mainPathways‘ containing IDs of pathways not reducable to each other, and ‘parentPathways‘ with vector describing for all the pathways to which ones they can be reduced. For pathways from ‘mainPathwyas‘ vector ‘parentPathways‘ contains ‘NA‘ values.

Examples

```
data(examplePathways)
data(exampleRanks)
fgseaRes <- fgsea(examplePathways, exampleRanks, nperm=10000, maxSize=500)
collapsedPathways <- collapsePathways(fgseaRes[order(pval)][padj < 0.01],
                                      examplePathways, exampleRanks)
mainPathways <- fgseaRes[pathway %in% collapsedPathways$mainPathways][
                         order(-NES), pathway]
```

`collapsePathwaysORA` *Collapse list of enriched pathways to independent ones. Version for ORA hypergeometric test.*

Description

Collapse list of enriched pathways to independent ones. Version for ORA hypergeometric test.

Usage

```
collapsePathwaysORA(foraRes, pathways, genes, universe, pval.threshold = 0.05)
```

Arguments

`foraRes`	Table with results of running fgsea(), should be filtered by p-value, for example by selecting ones with padj < 0.01.
`pathways`	List of pathways, should contain all the pathways present in ‘fgseaRes‘.
`genes`	Set of query genes, same as in ‘fora()‘
`universe`	A universe from whiche ‘genes‘ were selected, same as in ‘fora()‘
`pval.threshold`	Two pathways are considered dependent when p-value of enrichment of one pathways on background of another is greater then ‘pval.threshold‘.

Value

Named list with two elments: ‘mainPathways‘ containing IDs of pathways not reducable to each other, and ‘parentPathways‘ with vector describing for all the pathways to which ones they can be reduced. For pathways from ‘mainPathwyas‘ vector ‘parentPathways‘ contains ‘NA‘ values.

Examples

```
data(examplePathways)
data(exampleRanks)
foraRes <- fora(examplePathways, genes=tail(names(exampleRanks), 200), universe=names(exampleRanks))
collapsedPathways <- collapsePathwaysORA(foraRes[order(pval)][padj < 0.01],
                                         examplePathways,
                                         genes=tail(names(exampleRanks), 200),
                                         universe=names(exampleRanks))

mainPathways <- foraRes[pathway %in% collapsedPathways$mainPathways][
                        order(pval), pathway]
```

examplePathways *Example list of mouse Reactome pathways.*

Description

The list was obtained by selecting all the pathways from ‘reactome.db‘ package that contain mouse genes. The exact script is available as system.file("gen_reactome_pathways.R", package="fgsea")

exampleRanks *Example vector of gene-level statistics obtained for Th1 polarization.*

Description

The data were obtained by doing differential expression between Naive and Th1-activated states for GEO dataset GSE14308. The exact script is available as system.file("gen_gene_ranks.R", package="fgsea")

fgsea *Wrapper to run methods for preranked gene set enrichment analysis.*

Description

This function provide an interface to two existing functions: fgseaSimple, fgseaMultilevel. By default, the fgseaMultilevel function is used for analysis. For compatibility with the previous implementation you can pass the ‘nperm‘ argument to the function.

Usage

```
fgsea(...)
```

Arguments

...	optional arguments for functions fgseaSimple, fgseaMultilevel

Value

A table with GSEA results. Each row corresponds to a tested pathway.

Examples

```
data(examplePathways)
data(exampleRanks)
fgseaRes <- fgsea(examplePathways, exampleRanks, maxSize=500)
# Testing only one pathway is implemented in a more efficient manner
fgseaRes1 <- fgsea(examplePathways[1], exampleRanks)
```

fgseaLabel *Runs label-permuring gene set enrichment analysis.*

Description

Runs label-permuring gene set enrichment analysis.

Usage

```
fgseaLabel(
  pathways,
  mat,
  labels,
  nperm,
  minSize = 1,
  maxSize = Inf,
  nproc = 0,
  gseaParam = 1,
  BPPARAM = NULL
)
```

Arguments

pathways	List of gene sets to check.
mat	Gene expression matrix. Row name should be the same as in 'pathways'
labels	Numeric vector of labels for the correlation score of the same length as the number of columns in 'mat'
nperm	Number of permutations to do. Minimial possible nominal p-value is about 1/nperm
minSize	Minimal size of a gene set to test. All pathways below the threshold are excluded.
maxSize	Maximal size of a gene set to test. All pathways above the threshold are excluded.
nproc	If not equal to zero sets BPPARAM to use nproc workers (default = 0).
gseaParam	GSEA parameter value, all gene-level statis are raised to the power of 'gseaParam' before calculation of GSEA enrichment scores.
BPPARAM	Parallelization parameter used in bplapply. Can be used to specify cluster to run. If not initialized explicitly or by setting 'nproc' default value 'bpparam()' is used.

Value

A table with GSEA results. Each row corresponds to a tested pathway. The columns are the following:

- pathway – name of the pathway as in ‘names(pathway)‘;
- pval – an enrichment p-value;
- padj – a BH-adjusted p-value;
- ES – enrichment score, same as in Broad GSEA implementation;
- NES – enrichment score normalized to mean enrichment of random samples of the same size;
- nMoreExtreme‘ – a number of times a random gene set had a more extreme enrichment score value;
- size – size of the pathway after removing genes not present in ‘names(stats)‘.
- leadingEdge – vector with indexes of leading edge genes that drive the enrichment, see http://software.broadinstitute.org/gsea/doc/GSEAUserGuideTEXT.htm#_Running_a_Leading.

Examples

```
library(limma)
library(GEOquery)
es <- getGEO("GSE19429", AnnotGPL = TRUE)[[1]]
exprs(es) <- normalizeBetweenArrays(log2(exprs(es)+1), method="quantile")
es <- es[!grepl("///", fData(es)$`Gene ID`), ]
es <- es[fData(es)$`Gene ID` != "", ]
es <- es[order(apply(exprs(es), 1, mean), decreasing=TRUE), ]
es <- es[!duplicated(fData(es)$`Gene ID`), ]
rownames(es) <- fData(es)$`Gene ID`

pathways <- reactomePathways(rownames(es))
mat <- exprs(es)
labels <- as.numeric(as.factor(gsub(" .*", "", es$title)))
fgseaRes <- fgseaLabel(pathways, mat, labels, nperm = 1000, minSize = 15, maxSize = 500)
```

`fgseaMultilevel` *Runs preranked gene set enrichment analysis.*

Description

This feature is based on the adaptive multilevel splitting Monte Carlo approach. This allows us to exceed the results of simple sampling and calculate arbitrarily small P-values.

Usage

```
fgseaMultilevel(
  pathways,
  stats,
  sampleSize = 101,
  minSize = 1,
  maxSize = Inf,
```

```
  eps = 1e-50,
  scoreType = c("std", "pos", "neg"),
  nproc = 0,
  gseaParam = 1,
  BPPARAM = NULL,
  nPermSimple = 1000,
  absEps = NULL
)
```

Arguments

pathways	List of gene sets to check.
stats	Named vector of gene-level stats. Names should be the same as in 'pathways'
sampleSize	The size of a random set of genes which in turn has size = pathwaySize
minSize	Minimal size of a gene set to test. All pathways below the threshold are excluded.
maxSize	Maximal size of a gene set to test. All pathways above the threshold are excluded.
eps	This parameter sets the boundary for calculating the p value.
scoreType	This parameter defines the GSEA score type. Possible options are ("std", "pos", "neg")
nproc	If not equal to zero sets BPPARAM to use nproc workers (default = 0).
gseaParam	GSEA parameter value, all gene-level statis are raised to the power of 'gseaParam' before calculation of GSEA enrichment scores.
BPPARAM	Parallelization parameter used in bplapply. Can be used to specify cluster to run. If not initialized explicitly or by setting 'nproc' default value 'bpparam()' is used.
nPermSimple	Number of permutations in the simple fgsea implementation for preliminary estimation of P-values.
absEps	deprecated, use 'eps' parameter instead

Value

A table with GSEA results. Each row corresponds to a tested pathway. The columns are the following

- pathway – name of the pathway as in 'names(pathway)';
- pval – an enrichment p-value;
- padj – a BH-adjusted p-value;
- log2err – the expected error for the standard deviation of the P-value logarithm.
- ES – enrichment score, same as in Broad GSEA implementation;
- NES – enrichment score normalized to mean enrichment of random samples of the same size;
- size – size of the pathway after removing genes not present in 'names(stats)'.
- leadingEdge – vector with indexes of leading edge genes that drive the enrichment, see http://software.broadinstitute.org/gsea/doc/GSEAUserGuideTEXT.htm#_Running_a_Leading.

Examples

```
data(examplePathways)
data(exampleRanks)
fgseaMultilevelRes <- fgseaMultilevel(examplePathways, exampleRanks, maxSize=500)
```

`fgseaSimple` *Runs preranked gene set enrichment analysis.*

Description

The function takes about *O(nk^{3/2})* time, where *n* is number of permutations and *k* is a maximal size of the pathways. That means that setting ‘maxSize‘ parameter with a value of ~500 is strongly recommended.

Usage

```
fgseaSimple(
  pathways,
  stats,
  nperm,
  minSize = 1,
  maxSize = Inf,
  scoreType = c("std", "pos", "neg"),
  nproc = 0,
  gseaParam = 1,
  BPPARAM = NULL
)
```

Arguments

`pathways`	List of gene sets to check.
`stats`	Named vector of gene-level stats. Names should be the same as in ’pathways’
`nperm`	Number of permutations to do. Minimial possible nominal p-value is about 1/nperm
`minSize`	Minimal size of a gene set to test. All pathways below the threshold are excluded.
`maxSize`	Maximal size of a gene set to test. All pathways above the threshold are excluded.
`scoreType`	This parameter defines the GSEA score type. Possible options are ("std", "pos", "neg")
`nproc`	If not equal to zero sets BPPARAM to use nproc workers (default = 0).
`gseaParam`	GSEA parameter value, all gene-level statis are raised to the power of ‘gseaParam‘ before calculation of GSEA enrichment scores.
`BPPARAM`	Parallelization parameter used in bplapply. Can be used to specify cluster to run. If not initialized explicitly or by setting ‘nproc‘ default value ‘bpparam()‘ is used.

Value

A table with GSEA results. Each row corresponds to a tested pathway. The columns are the following:

- pathway – name of the pathway as in ‘names(pathway)‘;
- pval – an enrichment p-value;
- padj – a BH-adjusted p-value;
- ES – enrichment score, same as in Broad GSEA implementation;
- NES – enrichment score normalized to mean enrichment of random samples of the same size;
- nMoreExtreme‘ – a number of times a random gene set had a more extreme enrichment score value;
- size – size of the pathway after removing genes not present in ‘names(stats)‘.
- leadingEdge – vector with indexes of leading edge genes that drive the enrichment, see http://software.broadinstitute.org/gsea/doc/GSEAUserGuideTEXT.htm#_Running_a_Leading.

Examples

```
data(examplePathways)
data(exampleRanks)
fgseaRes <- fgseaSimple(examplePathways, exampleRanks, nperm=10000, maxSize=500)
# Testing only one pathway is implemented in a more efficient manner
fgseaRes1 <- fgseaSimple(examplePathways[1], exampleRanks, nperm=10000)
```

`fgseaSimpleImpl` *Runs preranked gene set enrichment analysis for preprocessed input data.*

Description

Runs preranked gene set enrichment analysis for preprocessed input data.

Usage

```
fgseaSimpleImpl(
  pathwayScores,
  pathwaysSizes,
  pathwaysFiltered,
  leadingEdges,
  permPerProc,
  seeds,
  toKeepLength,
  stats,
  BPPARAM,
  scoreType
)
```

Arguments

pathwayScores	Vector with enrichment scores for the ‘pathways‘.
pathwaysSizes	Vector of pathways sizes.
pathwaysFiltered	Filtered pathways.
leadingEdges	Leading edge genes.
permPerProc	Parallelization parameter for permutations.
seeds	Seed vector
toKeepLength	Number of ‘pathways‘ that meet the condition for ‘minSize‘ and ‘maxSize‘.
stats	Named vector of gene-level stats. Names should be the same as in ’pathways’
BPPARAM	Parallelization parameter used in bplapply.
scoreType	This parameter defines the GSEA score type. Possible options are ("std", "pos", "neg") Can be used to specify cluster to run. If not initialized explicitly or by setting ‘nproc‘ default value ‘bpparam()‘ is used.

Value

A table with GSEA results. Each row corresponds to a tested pathway. The columns are the following:

- pathway – name of the pathway as in ‘names(pathway)‘;
- pval – an enrichment p-value;
- padj – a BH-adjusted p-value;
- ES – enrichment score, same as in Broad GSEA implementation;
- NES – enrichment score normalized to mean enrichment of random samples of the same size;
- nMoreExtreme‘ – a number of times a random gene set had a more extreme enrichment score value;
- size – size of the pathway after removing genes not present in ‘names(stats)‘.
- leadingEdge – vector with indexes of leading edge genes that drive the enrichment, see http://software.broadinstitute.org/gsea/doc/GSEAUserGuideTEXT.htm#_Running_a_Leading.

---

fora *Simple overrepresentation analysis based on hypergeometric test*

Description

Simple overrepresentation analysis based on hypergeometric test

Usage

```
fora(pathways, genes, universe, minSize = 1, maxSize = Inf)
```

Arguments

pathways	List of gene sets to check.
genes	Set of query genes
universe	A universe from whiche 'genes' were selected
minSize	Minimal size of a gene set to test. All pathways below the threshold are excluded.
maxSize	Maximal size of a gene set to test. All pathways above the threshold are excluded.

Value

A table with ORA results. Each row corresponds to a tested pathway. The columns are the following:

- pathway – name of the pathway as in 'names(pathway)';
- pval – an enrichment p-value from hypergeometric test;
- padj – a BH-adjusted p-value;
- overlap – size of the overlap;
- size – size of the gene set;
- leadingEdge – vector with overlapping genes.

Examples

```
data(examplePathways)
data(exampleRanks)
foraRes <- fora(examplePathways, genes=tail(names(exampleRanks), 200), universe=names(exampleRanks))
```

---

gmtPathways *Returns a list of pathways from a GMT file.*

Description

Returns a list of pathways from a GMT file.

Usage

```
gmtPathways(gmt.file)
```

Arguments

gmt.file	Path to a GMT file.

Value

A list of vectors with gene sets.

Examples

```
pathways <- gmtPathways(system.file(
     "extdata", "mouse.reactome.gmt", package="fgsea"))
```

`mapIdsList` *Effeciently converts collection of pathways using AnnotationDbi::mapIds function. Parameters are the sames as for mapIds except for keys, which is assumed to be a list of vectors.*

Description

Effeciently converts collection of pathways using AnnotationDbi::mapIds function. Parameters are the sames as for mapIds except for keys, which is assumed to be a list of vectors.

Usage

```
mapIdsList(x, keys, column, keytype, ...)
```

Arguments

`x`	the AnnotationDb object. But in practice this will mean an object derived from an AnnotationDb object such as a OrgDb or ChipDb object.
`keys`	a list of vectors with gene ids
`column`	the column to search on
`keytype`	the keytype that matches the keys used
`...`	other parameters passed to AnnotationDbi::mapIds

See Also

AnnotationDbi::mapIds

Examples

```
library(org.Mm.eg.db)
data(exampleRanks)
fgseaRes <- fgsea(examplePathways, exampleRanks, maxSize=500, eps=1e-4)
fgseaRes[, leadingEdge := mapIdsList(org.Mm.eg.db, keys=leadingEdge, column="SYMBOL", keytype="ENTREZID")]
```

`multilevelError` *Calculates the expected error for the standard deviation of the P-value logarithm.*

Description

Calculates the expected error for the standard deviation of the P-value logarithm.

Usage

```
multilevelError(pval, sampleSize)
```

Arguments

`pval`	P-value
`sampleSize`	equivavlent to sampleSize in fgseaMultilevel

Value

The value of the expected error

Examples

```
expectedError <- multilevelError(pval=1e-10, sampleSize=1001)
```

`multilevelImpl` *Calculates P-values for preprocessed data.*

Description

Calculates P-values for preprocessed data.

Usage

```
multilevelImpl(
  multilevelPathwaysList,
  stats,
  sampleSize,
  seed,
  eps,
  sign = FALSE,
  BPPARAM = NULL
)
```

Arguments

multilevelPathwaysList	List of pathways for which P-values will be calculated.
stats	Named vector of gene-level stats. Names should be the same as in 'pathways'
sampleSize	The size of a random set of genes which in turn has size = pathwaySize
seed	'seed' parameter from 'fgseaMultilevel'
eps	This parameter sets the boundary for calculating the p value.
sign	This option will be used in future implementations.
BPPARAM	Parallelization parameter used in bplapply. Can be used to specify cluster to run. If not initialized explicitly or by setting 'nproc' default value 'bpparam()' is used.

Value

List of P-values.

---

plotEnrichment *Plots GSEA enrichment plot.*

Description

Plots GSEA enrichment plot.

Usage

```
plotEnrichment(pathway, stats, gseaParam = 1, ticksSize = 0.2)
```

Arguments

pathway	Gene set to plot.
stats	Gene-level statistics.
gseaParam	GSEA parameter.
ticksSize	width of vertical line corresponding to a gene (default: 0.2)

Value

ggplot object with the enrichment plot.

Examples

```
data(examplePathways)
data(exampleRanks)
## Not run:
plotEnrichment(examplePathways[["5991130_Programmed_Cell_Death"]],
               exampleRanks)

## End(Not run)
```

plotGseaTable *Plots table of enrichment graphs using ggplot and gridExtra.*

Description

Plots table of enrichment graphs using ggplot and gridExtra.

Usage

```
plotGseaTable(
  pathways,
  stats,
  fgseaRes,
  gseaParam = 1,
  colwidths = c(5, 3, 0.8, 1.2, 1.2),
  render = TRUE
)
```

Arguments

pathways	Pathways to plot table, as in 'fgsea' function.
stats	Gene-level stats, as in 'fgsea' function.
fgseaRes	Table with fgsea results.
gseaParam	GSEA-like parameter. Adjusts displayed statistic values, values closer to 0 flatten plots. Default = 1, value of 0.5 is a good choice too.
colwidths	Vector of five elements corresponding to column width for grid.arrange. Can be both units and simple numeric vector, in latter case it defines proportions, not actual sizes. If column width is set to zero, the column is not drawn.
render	If true, the plot is rendered to the current device. Otherwise, the grob is returned. Default is true.

Value

TableGrob object returned by grid.arrange.

Examples

```
data(examplePathways)
data(exampleRanks)
fgseaRes <- fgsea(examplePathways, exampleRanks, nperm=1000,
                  minSize=15, maxSize=100)
topPathways <- fgseaRes[head(order(pval), n=15)][order(NES), pathway]
## Not run:
plotGseaTable(examplePathways[topPathways], exampleRanks,
              fgseaRes, gseaParam=0.5)

## End(Not run)
```

reactomePathways *Returns a list of Reactome pathways for given Entrez gene IDs*

Description

Returns a list of Reactome pathways for given Entrez gene IDs

Usage

```
reactomePathways(genes)
```

Arguments

genes	Entrez IDs of query genes.

Value

A list of vectors with gene sets.

Examples

```
data(exampleRanks)
pathways <- reactomePathways(names(exampleRanks))
```

writeGmtPathways *Write collection of pathways (list of vectors) to a gmt file*

Description

Write collection of pathways (list of vectors) to a gmt file

Usage

```
writeGmtPathways(pathways, gmt.file)
```

Arguments

pathways	a named list of vectors with gene ids
gmt.file	name of the output file

Examples

```
data(examplePathways)
writeGmtPathways(examplePathways, tempfile("examplePathways", fileext=".gmt"))
```

Index