Package ‘fgsea’

March 3, 2024

Title Fast Gene Set Enrichment Analysis
Version 1.28.0
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 standard approaches to multiple hypothesis correction.
biocViews GeneExpression, DifferentialExpression, GeneSetEnrichment, Pathways
SystemRequirements C++11
Depends R (>= 4.1)
Imports Rcpp, data.table, BiocParallel, stats, ggplot2 (>= 2.2.0), cowplot, grid, fastmatch, Matrix, scales, utils
Suggests testthat, knitr, rmarkdown, reactome.db, AnnotationDbi, parallel, org.Mm.eg.db, limma, GEOquery, msigdbr, aggregation, Seurat
License MIT + file LICENCE
LazyData true
LinkingTo Rcpp, BH
RoxygenNote 7.2.3
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_18
git_last_commit 6c19b1e
git_last_commit_date 2023-10-24
Repository Bioconductor 3.18
Date/Publication 2024-03-03
Author  Gennady Korotkevich [aut],
        Vladimir Sukhov [aut],
        Nikolay Budin [ctb],
        Alexey Sergushichev [aut, cre]

Maintainer  Alexey Sergushichev <alsergbox@gmail.com>

R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>calcGseaStat</td>
<td>3</td>
</tr>
<tr>
<td>calcGseaStatBatchCpp</td>
<td>4</td>
</tr>
<tr>
<td>collapsePathways</td>
<td>4</td>
</tr>
<tr>
<td>collapsePathwaysGeseca</td>
<td>5</td>
</tr>
<tr>
<td>collapsePathwaysORA</td>
<td>6</td>
</tr>
<tr>
<td>exampleExpressionMatrix</td>
<td>7</td>
</tr>
<tr>
<td>examplePathways</td>
<td>7</td>
</tr>
<tr>
<td>exampleRanks</td>
<td>8</td>
</tr>
<tr>
<td>fgsea</td>
<td>8</td>
</tr>
<tr>
<td>fgseaLabel</td>
<td>9</td>
</tr>
<tr>
<td>fgseaMultilevel</td>
<td>10</td>
</tr>
<tr>
<td>fgseaSimple</td>
<td>12</td>
</tr>
<tr>
<td>fgseaSimpleImpl</td>
<td>14</td>
</tr>
<tr>
<td>fora</td>
<td>15</td>
</tr>
<tr>
<td>geseca</td>
<td>16</td>
</tr>
<tr>
<td>gesecaSimple</td>
<td>17</td>
</tr>
<tr>
<td>gmtPathways</td>
<td>18</td>
</tr>
<tr>
<td>mapIdsList</td>
<td>19</td>
</tr>
<tr>
<td>multilevelError</td>
<td>20</td>
</tr>
<tr>
<td>multilevelImpl</td>
<td>20</td>
</tr>
<tr>
<td>plotCoregulationProfile</td>
<td>21</td>
</tr>
<tr>
<td>plotCoregulationProfileReduction</td>
<td>22</td>
</tr>
<tr>
<td>plotCoregulationProfileSpatial</td>
<td>23</td>
</tr>
<tr>
<td>plotEnrichment</td>
<td>24</td>
</tr>
<tr>
<td>plotEnrichmentData</td>
<td>24</td>
</tr>
<tr>
<td>plotGesecaTable</td>
<td>26</td>
</tr>
<tr>
<td>plotGseaTable</td>
<td>27</td>
</tr>
<tr>
<td>reactomePathways</td>
<td>28</td>
</tr>
<tr>
<td>writeGmtPathways</td>
<td>29</td>
</tr>
</tbody>
</table>

Index  30
**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**

```r
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 following 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;
Examples

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

calcGseaStatBatchCpp  Calculates GSEA statistic values for all gene sets in 'selectedStats' list.

Description
Takes $O(n + m \cdot k \log k)$ 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 than ‘pval.threshold’.

nperm Number of permutations to test for independence, should be several times greater than ‘1/pval.threshold’. Default value: ‘10/pval.threshold’.

gseaParam GSEA parameter, same as for ‘fgsea()’

Value

Named list with two elements: ‘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 ‘mainPathways’ 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]

collapsePathwaysGeseca

Usage

collapsePathwaysGeseca(
  gesecaRes,
  pathways,
  E,
  center = TRUE,
  scale = FALSE,
  eps = min(c(1e-50, gesecaRes$pval)),
  checkDepth = 10,
)
collapsePathwaysORA

Arguments

gesecaRes Table with results of running geseca(), 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 ‘gesecaRes’.
E expression matrix, the same as in ‘geseca()’.
center a logical value indicating whether the gene expression should be centered to have zero mean before the analysis takes place. The default is TRUE. The value is passed to scale.
scale a logical value indicating whether the gene expression should be scaled to have unit variance before the analysis takes place. The default is FALSE. The value is passed to scale.
eps eps parameter for internal gesecaMultilevel runs. Default: \( \min(c(1e^{-50}, gesecaRes$pval)) \)
checkDepth how much pathways to check against
nproc If not equal to zero sets BPPARAM to use nproc workers (default = 0).
BPPARAM Parallelization parameter used in bplapply.

collapsePathwaysORA \hspace{1cm} \textit{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

\begin{verbatim}
collapsePathwaysORA(foraRes, pathways, genes, universe, pval.threshold = 0.05)
\end{verbatim}

Arguments

goraRes 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’.
exampleExpressionMatrix

**Value**

Named list with two elements: ‘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 ‘mainPathways’ 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]

**exampleExpressionMatrix**

*Example of expression values obtained for GSE14308.*

**Description**

Expression data was obtained by preprocessing the GSE14308 dataset. For the matrix of gene expression value, the following steps were performed:

- expression values were log2-scaled
- quantile-type normalization was performed between arrays
- rows were collapsed by ‘ENTREZID’
- rows were sorted in descending order by mean expression value per gene
- finally, top-10,000 genes were taken

The exact script is available as system.file("gen_gse14308_expression_matrix.R", package="fgsea")

**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(
  pathways,
  stats,
  minSize = 1,
  maxSize = length(stats) - 1,
  gseaParam = 1,
  ...
)

Arguments

pathways  List of gene sets to check.
stats  Named vector of gene-level stats. Names should be the same as in 'pathways'
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.
gseaParam  GSEA parameter value, all gene-level statis are raised to the power of 'gseaParam'
...  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,  # 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. Minimal possible nominal p-value is about 1/nperm
  minSize = 1,  # Minimal size of a gene set to test. All pathways below the threshold are excluded.
  maxSize = nrow(mat) - 1,  # Maximal size of a gene set to test. All pathways above the threshold are excluded.
  nproc = 0,  # If not equal to zero sets BPPARAM to use nproc workers (default = 0).
  gseaParam = 1,  # GSEA parameter value, all gene-level statis are raised to the power of 'gseaParam' before calculation of GSEA enrichment scores.
  BPPARAM = NULL  # 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)`.

Examples

```r
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 = length(stats) - 1,
  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"). By default ("std") the enrichment score is computed as in the original GSEA. The "pos" and "neg" score types are intended to be used for one-tailed tests (i.e. when one is interested only in positive ("pos") or negative ("neg") enrichment).
nproc        If not equal to zero sets BPPARAM to use nproc workers (default = 0).
gseaParam    GSEA parameter value, all gene-level stats 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
fgseaSimple

- 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 = length(stats) - 1,
  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. Minimal possible nominal p-value is about 1/nperm
### fgseaSimple

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>minSize</strong></td>
<td>Minimal size of a gene set to test. All pathways below the threshold are excluded.</td>
</tr>
<tr>
<td><strong>maxSize</strong></td>
<td>Maximal size of a gene set to test. All pathways above the threshold are excluded.</td>
</tr>
<tr>
<td><strong>scoreType</strong></td>
<td>This parameter defines the GSEA score type. Possible options are (&quot;std&quot;, &quot;pos&quot;, &quot;neg&quot;). By default (&quot;std&quot;) the enrichment score is computed as in the original GSEA. The &quot;pos&quot; and &quot;neg&quot; score types are intended to be used for one-tailed tests (i.e. when one is interested only in positive (&quot;pos&quot;) or negative (&quot;neg&quot;) enrichment).</td>
</tr>
<tr>
<td><strong>nproc</strong></td>
<td>If not equal to zero sets BPPARAM to use nproc workers (default = 0).</td>
</tr>
<tr>
<td><strong>gseaParam</strong></td>
<td>GSEA parameter value, all gene-level stats are raised to the power of 'gseaParam' before calculation of GSEA enrichment scores.</td>
</tr>
<tr>
<td><strong>BPPARAM</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

### 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)'.

### Examples

```r
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

**Description**

Runs preranked gene set enrichment analysis for preprocessed input data.

**Usage**

```r
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)';
fora

Simple overrepresentation analysis based on hypergeometric test

Description

Simple overrepresentation analysis based on hypergeometric test

Usage

fora(pathways, genes, universe, minSize = 1, maxSize = length(universe) - 1)

Arguments

- **pathways**: List of gene sets to check.
- **genes**: Set of query genes
- **universe**: A universe from which '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))

geseca

Runs multilevel Monte-Carlo variant for performing gene sets co-regulation analysis

Description

This function is based on the adaptive multilevel splitting Monte Carlo approach and allows to estimate arbitrarily small P-values for the task of analyzing variance along a set of genes.

Usage

geseca(
  pathways,
  E,
  minSize = 1,
  maxSize = nrow(E) - 1,
  center = TRUE,
  scale = FALSE,
  sampleSize = 101,
  eps = 1e-50,
  nproc = 0,
  BPPARAM = NULL,
  nPermSimple = 1000
)

Arguments

pathways List of gene sets to check.
E expression matrix, rows corresponds to genes, columns corresponds to samples.
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.
center a logical value indicating whether the gene expression should be centered to have zero mean before the analysis takes place. The default is TRUE. The value is passed to scale.
scale a logical value indicating whether the gene expression should be scaled to have unit variance before the analysis takes place. The default is FALSE. The value is passed to scale.
sampleSize sample size for conditional sampling.
**Value**

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

- **pathway** – name of the pathway as in `names(pathways)`;
- **pctVar** – percent of explained variance along gene set;
- **pval** – P-value that corresponds to the gene set score;
- **padj** – a BH-adjusted p-value;
- **size** – size of the pathway after removing genes not present in `rownames(E)`.

**Examples**

data("exampleExpressionMatrix")
data("examplePathways")
gr <- geseca(examplePathways, exampleExpressionMatrix, minSize=15, maxSize=500)

---

**Description**

This function is based on the rude Monte Carlo sampling approach and P-value calculation accuracy is limited to ‘1 / nperm’ value.

**Usage**

```r
  gesecaSimple(
    pathways,
    E,
    minSize = 1,
    maxSize = nrow(E) - 1,
    center = TRUE,
    scale = FALSE,
    nperm = 1000,
    nproc = 0,
    BPPARAM = NULL
  )
```
gmtPathways

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pathways</td>
<td>List of gene sets to check.</td>
</tr>
<tr>
<td>E</td>
<td>Expression matrix, rows correspond to genes, columns correspond to samples.</td>
</tr>
<tr>
<td>minSize</td>
<td>Minimal size of a gene set to test. All pathways below the threshold are excluded.</td>
</tr>
<tr>
<td>maxSize</td>
<td>Maximal size of a gene set to test. All pathways above the threshold are excluded.</td>
</tr>
<tr>
<td>center</td>
<td>A logical value indicating whether the gene expression should be centered to have zero mean before the analysis takes place. The default is TRUE. The value is passed to scale.</td>
</tr>
<tr>
<td>scale</td>
<td>A logical value indicating whether the gene expression should be scaled to have unit variance before the analysis takes place. The default is FALSE. The value is passed to scale.</td>
</tr>
<tr>
<td>nperm</td>
<td>Number of permutations to do. Minimal possible nominal p-value is about 1/nperm</td>
</tr>
<tr>
<td>nproc</td>
<td>If not equal to zero sets BPPARAM to use nproc workers (default = 0).</td>
</tr>
<tr>
<td>BPPARAM</td>
<td>Parallelization parameter used in bplapply.</td>
</tr>
</tbody>
</table>

Value

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

- pathway – name of the pathway as in 'names(pathways)';
- pctVar – percent of explained variance along gene set;
- pval – P-value that corresponds to the gene set score;
- padj – a BH-adjusted p-value;
- size – size of the pathway after removing genes not present in 'rownames(E)'.

Examples

```r
data("exampleExpressionMatrix")
data("examplePathways")
gesecaRes <- gesecaSimple(examplePathways, exampleExpressionMatrix, minSize=15, maxSize=500)
```

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

Description

Returns a list of pathways from a GMT file.

Usage

gmtPathways(gmt.file)```
mapIdsList

Arguments

gmt.file  Path to a GMT file.

Value

A list of vectors with gene sets.

Examples

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

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

Description

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

Usage

```r
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

```r
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 equivalent 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
)
plotCoregulationProfile

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.

plotCoregulationProfile

*Plots expression profile of a gene set*

Description

Plots expression profile of a gene set.

Usage

```r
plotCoregulationProfile(
  pathway,  # Gene set to plot.
  E,        # matrix with gene expression values
  center = TRUE,  # a logical value indicating whether the gene expression should be centered to have zero mean before the analysis takes place. The default is TRUE. The value is passed to scale.
  scale = FALSE,  # a logical value indicating whether the gene expression should be scaled to have unit variance before the analysis takes place. The default is FALSE. The value is passed to scale.
  titles = colnames(E),
  conditions = NULL
)
```

Arguments

- **pathway**
- **E**
- **center**
- **scale**
plotCoregulationProfileReduction

Plot a spatial expression profile of a gene set

**Value**

ggplot object with the coregulation profile plot

**Description**

Plot a spatial expression profile of a gene set

**Usage**

```r
plotCoregulationProfileReduction(
  pathway,
  object,
  title = NULL,
  assay = DefaultAssay(object),
  reduction = NULL,
  colors = c("darkblue", "lightgrey", "darkred"),
  guide = "colourbar",
  ...)
```

**Arguments**

- `pathway` Gene set to plot or a list of gene sets (see details below)
- `object` Seurat object
- `title` plot title
- `assay` assay to use for obtaining scaled data, preferably with
- `reduction` reduction to use for plotting (one of the ‘Seurat::Reductions(object)’)
- `colors` vector of three colors to use in the color scheme
- `guide` option for ‘ggplot2::scale_color_gradientn’ to control for presence of the color legend the same universe of genes in the scaled data
- `...` additional arguments for Seurat::FeaturePlot

**Value**

ggplot object (or a list of objects) with the coregulation profile plot

When the input is a list of pathways, pathway names are used for titles. A list of ggplot objects a returned in that case.
plotCoregulationProfileSpatial

*Plot a spatial expression profile of a gene set*

**Description**

Plot a spatial expression profile of a gene set

**Usage**

```r
plotCoregulationProfileSpatial(
    pathway,
    object,
    title = NULL,
    assay = DefaultAssay(object),
    colors = c("darkblue", "lightgrey", "darkred"),
    guide = "colourbar"
)
```

**Arguments**

- `pathway`: Gene set to plot or a list of gene sets (see details below)
- `object`: Seurat object
- `title`: plot title
- `assay`: assay to use for obtaining scaled data, preferably with the same universe of genes in the scaled data
- `colors`: vector of three colors to use in the color scheme
- `guide`: option for `ggplot2::scale_color_gradientn` to control for presence of the color legend the same universe of genes in the scaled data

**Value**

`ggplot` object (or a list of objects) with the coregulation profile plot

When the input is a list of pathways, pathway names are used for titles. A list of `ggplot` objects a returned in that case.
plotEnrichment

Plots GSEA enrichment plot. For more flexibility use `plotEnrichmentData` function.

Description

Plots GSEA enrichment plot. For more flexibility use `plotEnrichmentData` function.

Usage

\[ \text{plotEnrichment}(\text{pathway}, \text{stats}, \text{gseaParam} = 1, \text{ticksSize} = 0.2) \]

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

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

plotEnrichmentData

Returns data required for doing an enrichment plot.

Description

Returns data required for doing an enrichment plot.

Usage

```r
plotEnrichmentData(\text{pathway}, \text{stats}, \text{gseaParam} = 1) \]
```
plotEnrichmentData

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pathway</td>
<td>Gene set to plot.</td>
</tr>
<tr>
<td>stats</td>
<td>Gene-level statistics.</td>
</tr>
<tr>
<td>gseaParam</td>
<td>GSEA parameter.</td>
</tr>
</tbody>
</table>

Value

returns list with the following data:
* 'curve' - data.table with the coordinates of the enrichment curve;
* 'ticks' - data.table with statistic entries for each pathway gene, adjusted with gseaParam;
* 'stats' - data.table with statistic values for all of the genes, adjusted with gseaParam;
* 'posES', 'negES', 'spreadES' - values of the positive enrichment score, negative enrichment score, and difference between them;
* 'maxAbsStat' - maximal absolute value of statistic entries, adjusted with gseaParam

Examples

```r
library(ggplot2)
data(examplePathways)
data(exampleRanks)
pd <- plotEnrichmentData(
  pathway = examplePathways["5991130_Programmed_Cell_Death"],
  stats = exampleRanks
)
with(pd,
  ggplot(data=curve) +
  geom_line(aes(x=rank, y=ES), color="green") +
  geom_ribbon(data=stats,
    mapping=aes(x=rank, ymin=0,
      ymax=stat/maxAbsStat*(spreadES/4)),
    fill="grey") +
  geom_segment(data=ticks,
    mapping=aes(x=rank, y=-spreadES/16,
      xend=rank, yend=spreadES/16),
    size=0.2) +
  geom_hline(yintercept=posES, colour="red", linetype="dashed") +
  geom_hline(yintercept=negES, colour="red", linetype="dashed") +
  geom_hline(yintercept=0, colour="black") +
  theme(
    panel.background = element_blank(),
    panel.grid.major = element_line(color="grey92")
  ) +
  labs(x="rank", y="enrichment score")
)
plotGesecaTable  

Plots table of gene set profiles.

Description

Plots table of gene set profiles.

Usage

plotGesecaTable(
  gesecaRes,  
  pathways,  
  E,  
  center = TRUE,  
  scale = FALSE,  
  colwidths = c(5, 3, 0.8, 1.2, 1.2),  
  titles = colnames(E),  
  colors = c("blue", "white", "red"),  
  pathwayLabelStyle = NULL,  
  headerLabelStyle = NULL,  
  valueStyle = NULL,  
  axisLabelStyle = NULL,  
  axisLabelHeightScale = NULL
)

Arguments

gesecaRes  
Table with geseca results.

pathways  
Pathways to plot table, as in ‘geseca’ function.

E  
gene expression matrix, as in ‘geseca’ function.

center  
a logical value indicating whether the gene expression should be centered to have zero mean before the analysis takes place. The default is TRUE. The value is passed to scale.

scale  
a logical value indicating whether the gene expression should be scaled to have unit variance before the analysis takes place. The default is FALSE. The value is passed to scale.

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.

titles  
sample titles to use an axis labels. Default to ‘colnames(E)’

colors  
vector of three colors to use in the color scheme

pathwayLabelStyle  
list with style parameter adjustments for pathway labels. For example, ‘list(size=10, color="red")’ set the font size to 10 and color to red. See ‘cowplot::draw_text’ for possible options.
headerLabelStyle
  similar to ‘pathwayLabelStyle’ but for the table header.
valueStyle
  similar to ‘pathwayLabelStyle’ but for pctVar and p-value columns.
axisLabelStyle
  list with style parameter adjustments for sample labels. See ‘ggplot2::element_text’
  for possible options.
axisLabelHeightScale
  height of the row with axis labels compared to other rows. When set to ‘NULL’
  the value is determined automatically.

Value
  ggplot object with gene set profile plots

plotGseaTable（
  pathways,
  stats,
  fgseaRes,
  gseaParam = 1,
  colwidths = c(5, 3, 0.8, 1.2, 1.2),
  pathwayLabelStyle = NULL,
  headerLabelStyle = NULL,
  valueStyle = NULL,
  axisLabelStyle = NULL,
  render = NULL
)"

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.
pathwayLabelStyle

list with style parameter adjustments for pathway labels. For example, `list(size=10, color="red")` set the font size to 10 and color to red. See `cowplot::draw_text` for possible options.

headerLabelStyle

similar to `pathwayLabelStyle` but for the table header.

valueStyle

similar to `pathwayLabelStyle` but for NES and p-value columns.

axisLabelStyle

list with style parameter adjustments for stats axis labels. See `ggplot2::element_text` for possible options.

render

(deprecated)

Value

ggplot object with enrichment barcode plots

Examples

data(examplePathways)
data(exampleRanks)
fgseaRes <- fgsea(examplePathways, exampleRanks, minSize=15, maxSize=500)
topPathways <- fgseaRes[head(order(pval), n=15)][order(NES), pathway]
plotGseaTable(examplePathways[topPathways], exampleRanks,
  fgseaRes, gseaParam=0.5)

---

**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"))