Package: GSVA (via r-universe)

July 17, 2024

Version 1.53.4

Title Gene Set Variation Analysis for Microarray and RNA-Seq Data

Depends R (>= 3.5.0)

Imports methods, stats, utils, graphics, S4Vectors, IRanges, Biobase, SummarizedExperiment, GSEABase, Matrix (>= 1.5-0), parallel, BiocParallel, SingleCellExperiment, SpatialExperiment, sparseMatrixStats, DelayedArray, DelayedMatrixStats, HDF5Array, BiocSingular

Suggests BiocGenerics, RUnit, BiocStyle, knitr, rmarkdown, limma, RColorBrewer, org.Hs.eg.db, genefilter, edgeR, GSVAdata, shiny, shinydashboard, ggplot2, data.table, plotly, future, promises, shinybusy, shinyjs

Description Gene Set Variation Analysis (GSVA) is a non-parametric, unsupervised method for estimating variation of gene set enrichment through the samples of a expression data set. GSVA performs a change in coordinate systems, transforming the data from a gene by sample matrix to a gene-set by sample matrix, thereby allowing the evaluation of pathway enrichment for each sample. This new matrix of GSVA enrichment scores facilitates applying standard analytical methods like functional enrichment, survival analysis, clustering, CNV-pathway analysis or cross-tissue pathway analysis, in a pathway-centric manner.

License GPL (>= 2)

VignetteBuilder knitr

URL https://github.com/rcastelo/GSVA

BugReports https://github.com/rcastelo/GSVA/issues

Encoding UTF-8

biocViews FunctionalGenomics, Microarray, RNASeq, Pathways, GeneSetEnrichment

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Repository https://bioc.r-universe.dev

RemoteUrl https://github.com/bioc/GSVA

RemoteRef HEAD

RemoteSha 6db44567fd281ff26aad8360d0a397f8408b06d7

Contents

`computeGeneSetsOverlap` *Compute gene-sets overlap*

Description

Calculates the overlap among every pair of gene-sets given as input.

This function calculates the overlap between every pair of gene sets of the input argument `gSets`. Before this calculation takes place, the gene sets in `gSets` are firstly filtered to discard genes that do not match to the identifiers in `uniqGenes`. Secondly, they are further filtered to meet the minimum and/or maximum size specified with the arguments `minSize` and `maxSize`. The overlap between two gene sets is calculated as the number of common genes between the two gene sets divided by the smallest size of the two gene sets.

Usage

```
## S4 method for signature 'list,character'
computeGeneSetsOverlap(gSets, uniqGenes, minSize = 1, maxSize = Inf)

## S4 method for signature 'list,ExpressionSet'
computeGeneSetsOverlap(gSets, uniqGenes, minSize = 1, maxSize = Inf)
```

```
## S4 method for signature 'GeneSetCollection,character'
computeGeneSetsOverlap(gSets, uniqGenes, minSize = 1, maxSize = Inf)

## S4 method for signature 'GeneSetCollection,ExpressionSet'
computeGeneSetsOverlap(gSets, uniqGenes, minSize = 1, maxSize = Inf)
```

Arguments

gSets	Gene sets given either as a `list` or a `GeneSetCollection` object.
uniqGenes	Vector of unique genes to be considered when calculating the overlaps.
minSize	Minimum size.
maxSize	Maximum size.

Value

A gene-set by gene-set matrix of the overlap among every pair of gene sets.

Author(s)

J. Guinney

References

Hänzelmann, S., Castelo, R. and Guinney, J. GSVA: Gene set variation analysis for microarray and RNA-Seq data. *BMC Bioinformatics*, 14:7, 2013.

See Also

`filterGeneSets`

Examples

```
geneSets <- list(set1=as.character(1:4), set2=as.character(4:10))
computeGeneSetsOverlap(geneSets, unique(unlist(geneSets)))
```

---

deduplicateGeneSets *Handling of Duplicated Gene Set Names*

Description

Offers a choice of ways for handling duplicated gene set names that may not be suitable as input to other gene set analysis functions.

Usage

```
deduplicateGeneSets(
  geneSets,
  deduplUse = c("first", "drop", "union", "smallest", "largest")
)
```

Arguments

geneSets	A named list of gene sets represented as character vectors of gene IDs as e.g. returned by readGMT.
deduplUse	A character vector of length 1 specifying one of several methods to handle duplicated gene set names. Duplicated gene set names are explicitly forbidden by the GMT file format specification but can nevertheless be encountered in the wild. The available choices are: • first (the default): drops all gene sets whose names are duplicated according to the base R function and retains only the first occurence of a gene set name. • drop: removes *all* gene sets that have a duplicated name, including its first occurrence. • union: replaces gene sets with duplicated names by a single gene set containing the union of all their gene IDs. • smallest: drops gene sets with duplicated names and retains only the smallest of them, i.e. the one with the fewest gene IDs. If there are several smallest gene sets, the first will be selected. • largest: drops gene sets with duplicated names and retains only the largest of them, i.e. the one with the most gene IDs. If there are several largest gene sets, the first will be selected.

Value

A named list of gene sets that represented as character vectors of gene IDs.

---

filterGeneSets *Filter gene sets*

Description

Filters gene sets through a given minimum and maximum set size.

This function filters the input gene sets according to a given minimum and maximum set size.

Usage

```
## S4 method for signature 'list'
filterGeneSets(gSets, minSize = 1, maxSize = Inf)

## S4 method for signature 'GeneSetCollection'
filterGeneSets(gSets, minSize = 1, maxSize = Inf)
```

Arguments

gSets	Gene sets given either as a list or a GeneSetCollection object.
minSize	Minimum size.
maxSize	Maximum size.

Value

A collection of gene sets that meet the given minimum and maximum set size.

Author(s)

J. Guinney

References

Hänzelmann, S., Castelo, R. and Guinney, J. GSVA: Gene set variation analysis for microarray and RNA-Seq data. *BMC Bioinformatics*, 14:7, 2013.

See Also

computeGeneSetsOverlap

Examples

```
geneSets <- list(set1=as.character(1:4), set2=as.character(4:10))
filterGeneSets(geneSets, minSize=5)
```

---

geneSets *Retrieve or Determine Gene Sets*

Description

Retrieves or determines the gene sets that have been used or would be used in a `gsva()` gene set analysis. These are not necessarily the same as the input gene sets. See Details.

Usage

```
## S4 method for signature 'GsvaMethodParam'
geneSets(obj)

## S4 method for signature 'SummarizedExperiment'
geneSets(obj)

## S4 method for signature 'SingleCellExperiment'
geneSets(obj)

## S4 method for signature 'SpatialExperiment'
geneSets(obj)

## S4 method for signature 'GsvaExprData'
geneSets(obj)

## S4 method for signature 'GsvaMethodParam'
```

```
geneSetSizes(obj)

## S4 method for signature 'GsvaExprData'
geneSetSizes(obj)
```

Arguments

obj	An object of one of the following classes: • An expression data object of one of the classes described in GsvaExprData that is the return value of a call to gsva(). • A parameter object of one of the classes described in GsvaMethodParam that could be used in a call to gsva().

Details

The gene sets used in a gsva() gene set analysis, or just their sizes, may be a valuable input to subsequent analyses. However, they are not necessarily the same as the original input gene sets, or their sizes: based on user choices, the gene annotation used, or presence/absence of genes in gene sets and expression data set, gsva() may have to modify them during the preparation of an analysis run. In order to make use of these gene sets or their sizes, you can either

- retrieve them from the object returned by gsva() by passing this object to geneSets() or geneSetSizes(), or
- predict them by calling geneSets() or geneSetSizes() on the parameter object that would also be passed to gsva(). This is much slower and should only be done if you do not intend to run an actual gene set analysis.

geneSetSizes() is a convenience wrapper running lengths() on the list of gene sets returned by geneSets().

Value

The geneSets() methods return a named list of character vectors where each character vector contains the gene IDs of a gene set. The geneSetSizes() methods return a named integer vector of gene set sizes.

---

gsva *Gene Set Variation Analysis*

Description

Estimates GSVA enrichment scores. The API of this function has changed in the Bioconductor release 3.18 and this help page describes the new API. The old API is defunct and will be removed in the next Bioconductor release. If you are looking for the documentation of the old API to the gsva() function, please consult GSVA-pkg-defunct.

Usage

```
## S4 method for signature 'plageParam'
gsva(param, verbose = TRUE, BPPARAM = SerialParam(progressbar = verbose))

## S4 method for signature 'zscoreParam'
gsva(param, verbose = TRUE, BPPARAM = SerialParam(progressbar = verbose))

## S4 method for signature 'ssgseaParam'
gsva(param, verbose = TRUE, BPPARAM = SerialParam(progressbar = verbose))

## S4 method for signature 'gsvaParam'
gsva(param, verbose = TRUE, BPPARAM = SerialParam(progressbar = verbose))
```

Arguments

param	A parameter object of one of the following classes: • A gsvaParam object built using the constructor function gsvaParam. This object will trigger gsva() to use the GSVA algorithm by Hänzelmann et al. (2013). • A plageParam object built using the constructor function plageParam. This object will trigger gsva() to use the PLAGE algorithm by Tomfohr et al. (2005). • A zscoreParam object built using the constructor function zscoreParam. This object will trigger gsva() to use the combined z-score algorithm by Lee et al. (2008). • A ssgseaParam object built using the constructor function ssgseaParam. This object will trigger gsva() to use the ssGSEA algorithm by Barbie et al. (2009).
verbose	Gives information about each calculation step. Default: TRUE.
BPPARAM	An object of class BiocParallelParam specifying parameters related to the parallel execution of some of the tasks and calculations within this function.

Value

A gene-set by sample matrix of GSVA enrichment scores stored in a container object of the same type as the input expression data container. If the input was a base matrix or a dgCMatrix object, then the output will be a base matrix object with the gene sets employed in the calculations stored in an attribute called geneSets. If the input was an ExpressionSet object, then the output will be also an ExpressionSet object with the gene sets employed in the calculations stored in an attributed called geneSets. If the input was an object of one of the classes described in GsvaExprData, such as a SingleCellExperiment, then the output will be of the same class, where enrichment scores will be stored in an assay called es and the gene sets employed in the calculations will be stored in the rowData slot of the object under the column name gs.

References

Barbie, D.A. et al. Systematic RNA interference reveals that oncogenic KRAS-driven cancers require TBK1. *Nature*, 462(5):108-112, 2009. DOI

Hänzelmann, S., Castelo, R. and Guinney, J. GSVA: Gene set variation analysis for microarray and RNA-Seq data. *BMC Bioinformatics*, 14:7, 2013. DOI

Lee, E. et al. Inferring pathway activity toward precise disease classification. *PLoS Comp Biol*, 4(11):e1000217, 2008. DOI

Tomfohr, J. et al. Pathway level analysis of gene expression using singular value decomposition. *BMC Bioinformatics*, 6:225, 2005. DOI

See Also

plageParam, zscoreParam, ssgseaParam, gsvaParam

Examples

```
library(GSVA)
library(limma)

p <- 10 ## number of genes
n <- 30 ## number of samples
nGrp1 <- 15 ## number of samples in group 1
nGrp2 <- n - nGrp1 ## number of samples in group 2

## consider three disjoint gene sets
geneSets <- list(set1=paste("g", 1:3, sep=""),
                 set2=paste("g", 4:6, sep=""),
                 set3=paste("g", 7:10, sep=""))

## sample data from a normal distribution with mean 0 and st.dev. 1
y <- matrix(rnorm(n*p), nrow=p, ncol=n,
            dimnames=list(paste("g", 1:p, sep="") , paste("s", 1:n, sep="")))

## genes in set1 are expressed at higher levels in the last 'nGrp1+1' to 'n' samples
y[geneSets$set1, (nGrp1+1):n] <- y[geneSets$set1, (nGrp1+1):n] + 2

## build design matrix
design <- cbind(sampleGroup1=1, sampleGroup2vs1=c(rep(0, nGrp1), rep(1, nGrp2)))

## fit linear model
fit <- lmFit(y, design)

## estimate moderated t-statistics
fit <- eBayes(fit)

## genes in set1 are differentially expressed
topTable(fit, coef="sampleGroup2vs1")

## build GSVA parameter object
gsvapar <- gsvaParam(y, geneSets, maxDiff=TRUE)

## estimate GSVA enrichment scores for the three sets
gsva_es <- gsva(gsvapar)

## fit the same linear model now to the GSVA enrichment scores
```

```
fit <- lmFit(gsva_es, design)

## estimate moderated t-statistics
fit <- eBayes(fit)

## set1 is differentially expressed
topTable(fit, coef="sampleGroup2vs1")
```

GsvaExprData-class GsvaExprData *class*

Description

Virtual superclass of expression data classes supported by GSVA.

Details

GSVA supports expression data matrices in a growing number of containers and representations. This class union allows to store any of these in a slot of another class as well as defining common methods for all of them.

See Also

matrix, dgCMatrix, ExpressionSet, SummarizedExperiment, SingleCellExperiment, SpatialExperiment

---

GsvaGeneSets-class GsvaGeneSets *class*

Description

Virtual superclass of gene set classes supported by GSVA.

Details

GSVA supports gene sets in either a list of character vectors or an object of class GSEABase::GeneSetCollection. This class union allows to store any of these in a slot of another class as well as defining common methods for them.

See Also

list, GeneSetCollection

GsvaMethodParam-class GsvaMethodParam *class*

Description

Virtual superclass of method parameter classes supported by GSVA.

A virtual superclass of the GSVA packages' method-specific parameter classes.

Details

GSVA implements four single-sample gene set analysis methods: PLAGE, combined z-scores, ssGSEA, and GSVA. All of them take at least an expression data matrix and one or many gene sets as input. This virtual class provides the necessary slots for this minimum parameter set and serves as all GSVA method parameter classes,

The GSVA package implements four single-sample gene set analysis methods (PLAGE, combined z-scores, ssGSEA, and GSVA) and a respective method-specific parameter class that is used to invoke each of them with a matching set of parameters.

See Also

GsvaExprData, GsvaGeneSets, zscoreParam, plageParam, ssgseaParam, gsvaParam

plageParam, zscoreParam, ssgseaParam, gsvaParam

gsvaParam-class gsvaParam *class*

Description

Method-specific parameters for the GSVA method.

Objects of class gsvaParam contain the parameters for running the GSVA method.

Usage

```
gsvaParam(
  exprData,
  geneSets,
  assay = NA_character_,
  annotation = NA_character_,
  minSize = 1,
  maxSize = Inf,
  kcdf = c("Gaussian", "Poisson", "none"),
  tau = 1,
  maxDiff = TRUE,
  absRanking = FALSE
)
```

Arguments

exprData	The expression data. Must be one of the classes supported by GsvaExprData. Type help(GsvaExprData) to consult the available classes.
geneSets	The gene sets. Must be one of the classes supported by GsvaGeneSets.
assay	The name of the assay to use in case exprData is a multi-assay container, otherwise ignored. By default, the first assay is used.
annotation	The name of a Bioconductor annotation package for the gene identifiers occurring in the row names of the expression data matrix. This can be used to map gene identifiers occurring in the gene sets if those are provided in a GeneSetCollection. By default gene identifiers used in expression data matrix and gene sets are matched directly.
minSize	Minimum size of the resulting gene sets after gene identifier mapping. By default, the minimum size is 1.
maxSize	Maximum size of the resulting gene sets after gene identifier mapping. By default, the maximum size is Inf.
kcdf	Character vector of length 1 denoting the kernel to use during the non-parametric estimation of the cumulative distribution function of expression levels across samples. By default, kcdf="Gaussian" which is suitable when input expression values are continuous, such as microarray fluorescent units in logarithmic scale, RNA-seq log-CPMs, log-RPKMs or log-TPMs. When input expression values are integer counts, such as those derived from RNA-seq experiments, then this argument should be set to kcdf="Poisson".
tau	Numeric vector of length 1. The exponent defining the weight of the tail in the random walk performed by the GSVA (Hänzelmann et al., 2013) method. The default value is 1 as described in the paper.
maxDiff	Logical vector of length 1 which offers two approaches to calculate the enrichment statistic (ES) from the KS random walk statistic. • FALSE: ES is calculated as the maximum distance of the random walk from 0. This approach produces a distribution of enrichment scores that is bimodal, but it can give large enrichment scores to gene sets whose genes are not concordantly activated in one direction only. • TRUE (the default): ES is calculated as the magnitude difference between the largest positive and negative random walk deviations. This default value gives larger enrichment scores to gene sets whose genes are concordantly activated in one direction only.
absRanking	Logical vector of length 1 used only when maxDiff=TRUE. When absRanking=FALSE (default) a modified Kuiper statistic is used to calculate enrichment scores, taking the magnitude difference between the largest positive and negative random walk deviations. When absRanking=TRUE the original Kuiper statistic that sums the largest positive and negative random walk deviations is used.

Details

In addition to the two common parameter slots inherited from [GsvaMethodParam], this class has slots for the two method-specific parameters of the GSVA method described below.

In addition to an expression data set and a collection of gene sets, GSVA takes four method-specific parameters as described below.

Value

A new gsvaParam object.

Slots

`kcdf` Character vector of length 1 denoting the kernel to use during the non-parametric estimation of the cumulative distribution function of expression levels across samples. `kcdf="Gaussian"` is suitable when input expression values are continuous, such as microarray fluorescent units in logarithmic scale, RNA-seq log-CPMs, log-RPKMs or log-TPMs. When input expression values are integer counts, such as those derived from RNA-seq experiments, then this argument should be set to `kcdf="Poisson"`.

`tau` Numeric vector of length 1. The exponent defining the weight of the tail in the random walk performed by the GSVA (Hänzelmann et al., 2013) method.

`maxDiff` Logical vector of length 1 which offers two approaches to calculate the enrichment statistic (ES) from the KS random walk statistic.

- `FALSE`: ES is calculated as the maximum distance of the random walk from 0.
- `TRUE`: ES is calculated as the magnitude difference between the largest positive and negative random walk deviations.

`absRanking` Logical vector of length 1 used only when `mx.diff=TRUE`. When `abs.ranking=FALSE` a modified Kuiper statistic is used to calculate enrichment scores, taking the magnitude difference between the largest positive and negative random walk deviations. When `abs.ranking=TRUE` the original Kuiper statistic that sums the largest positive and negative random walk deviations, is used. In this latter case, gene sets with genes enriched on either extreme (high or low) will be regarded as 'highly' activated.

References

Hänzelmann, S., Castelo, R. and Guinney, J. GSVA: Gene set variation analysis for microarray and RNA-Seq data. *BMC Bioinformatics*, 14:7, 2013. DOI

See Also

GsvaExprData, GsvaGeneSets, GsvaMethodParam, plageParam, zscoreParam, ssgseaParam

Examples

```
library(GSVA)
library(GSVAdata)

data(leukemia)
data(c2BroadSets)

## for simplicity, use only a subset of the sample data
ses <- leukemia_eset[1:1000, ]
gsc <- c2BroadSets[1:100]
```

```
gp1 <- gsvaParam(ses, gsc)
gp1
```

---

igsva *Gene Set Variation Analysis*

Description

Starts an interactive GSVA shiny web app.

GSVA assesses the relative enrichment of gene sets across samples using a non-parametric approach. Conceptually, GSVA transforms a p-gene by n-sample gene expression matrix into a g-geneset by n-sample pathway enrichment matrix. This facilitates many forms of statistical analysis in the 'space' of pathways rather than genes, providing a higher level of interpretability.

The `igsva()` function starts an interactive shiny web app that allows the user to configure the arguments of the `gsva()` function and runs it on the computer. Please see the manual page of the `gsva()` function for a description of the arguments and their default and alternative values.

The input data may be loaded from the users workspace or by selecting a CSV file for the expression data, and a GMT file for the gene sets data.

Usage

```
igsva()
```

Value

A gene-set by sample matrix of GSVA enrichment scores after pressing the button 'Save & Close'. This result can be also downloaded as a CSV file with the 'Download' button.

Author(s)

J. Fernández and R. Castelo

References

Hänzelmann, S., Castelo, R. and Guinney, J. GSVA: Gene set variation analysis for microarray and RNA-Seq data. *BMC Bioinformatics*, 14:7, 2013.

See Also

`gsva()`

Examples

```
## Not run:
res <- igsva() ## this will open your browser with the GSVA shiny web app

## End(Not run)
```

---

plageParam-class plageParam *class*

Description

Method-specific parameters for the PLAGE method.

Objects of class plageParam contain the parameters for running the PLAGE method.

Usage

```
plageParam(
  exprData,
  geneSets,
  assay = NA_character_,
  annotation = NA_character_,
  minSize = 1,
  maxSize = Inf
)
```

Arguments

exprData	The expression data. Must be one of the classes supported by GsvaExprData. Type help(GsvaExprData) to consult the available classes.
geneSets	The gene sets. Must be one of the classes supported by GsvaGeneSets.
assay	The name of the assay to use in case exprData is a multi-assay container, otherwise ignored. By default, the first assay is used.
annotation	The name of a Bioconductor annotation package for the gene identifiers occurring in the row names of the expression data matrix. This can be used to map gene identifiers occurring in the gene sets if those are provided in a GeneSetCollection. By default gene identifiers used in expression data matrix and gene sets are matched directly.
minSize	Minimum size of the resulting gene sets after gene identifier mapping. By default, the minimum size is 1.
maxSize	Maximum size of the resulting gene sets after gene identifier mapping. By default, the maximum size is Inf.

Details

Since this method does not take any method-specific parameters, the parameter class does not add any slots to the common slots inherited from GsvaMethodParam.

PLAGE does not take any method-specific parameters in addition to an expression data set and a collection of gene sets.

Value

A new plageParam object.

References

Tomfohr, J. et al. Pathway level analysis of gene expression using singular value decomposition. *BMC Bioinformatics*, 6:225, 2005. DOI

See Also

GsvaExprData, GsvaGeneSets, GsvaMethodParam, zscoreParam, ssgseaParam, gsvaParam

Examples

```
library(GSVA)
library(GSVAdata)

data(leukemia)
data(c2BroadSets)

## for simplicity, use only a subset of the sample data
ses <- leukemia_eset[1:1000, ]
gsc <- c2BroadSets[1:100]
pp1 <- plageParam(ses, gsc)
pp1
```

---

readGMT *Import Gene Sets from a GMT File*

Description

Imports a list of gene sets from a GMT (Gene Matrix Transposed) format file, offering a choice of ways to handle duplicated gene set names.

Usage

```
readGMT(
  con,
  deduplUse = c("first", "drop", "union", "smallest", "largest", "custom")
)
```

Arguments

con	A connection object or character string containing e.g. a file name or URL. This is directly passed to readLines and hence may contain anything that readLines() can handle.
deduplUse	With the exception of the special method custom, all handling of duplicated gene set names is delegated to function deduplicateGeneSets and this argument is directly passed on. Please see ?deduplicateGeneSets. Using deduplUse=custom allows import of the GMT file for manual inspection and its content and remedy is the user's responsibility. However, gsva() will *not* accept the result for further use unless it is modified to have duplicated gene set names removed.

Value

A named list of gene sets that represented as character vectors of gene IDs.

See Also

readLines, deduplicateGeneSets

---

ssgseaParam-class ssgseaParam *class*

Description

Method-specific parameters for the ssGSEA method.

Objects of class ssgseaParam contain the parameters for running the ssGSEA method.

Usage

```
ssgseaParam(
  exprData,
  geneSets,
  assay = NA_character_,
  annotation = NA_character_,
  minSize = 1,
  maxSize = Inf,
  alpha = 0.25,
  normalize = TRUE,
  checkNA = c("auto", "yes", "no"),
  use = c("everything", "all.obs", "na.rm")
)

## S4 method for signature 'ssgseaParam'
anyNA(x, recursive = FALSE)
```

Arguments

exprData	The expression data. Must be one of the classes supported by GsvaExprData. Type help(GsvaExprData) to consult the available classes.
geneSets	The gene sets. Must be one of the classes supported by GsvaGeneSets.
assay	The name of the assay to use in case exprData is a multi-assay container, otherwise ignored. By default, the first assay is used.
annotation	The name of a Bioconductor annotation package for the gene identifiers occurring in the row names of the expression data matrix. This can be used to map gene identifiers occurring in the gene sets if those are provided in a GeneSetCollection. By default gene identifiers used in expression data matrix and gene sets are matched directly.
minSize	Minimum size of the resulting gene sets after gene identifier mapping. By default, the minimum size is 1.
maxSize	Maximum size of the resulting gene sets after gene identifier mapping. By default, the maximum size is Inf.
alpha	Numeric vector of length 1. The exponent defining the weight of the tail in the random walk performed by the ssGSEA (Barbie et al., 2009) method. The default value is 0.25 as described in the paper.
normalize	Logical vector of length 1; if TRUE runs the ssGSEA method from Barbie et al. (2009) normalizing the scores by the absolute difference between the minimum and the maximum, as described in their paper. Otherwise this last normalization step is skipped.
checkNA	Character string specifying whether the input expression data should be checked for the presence of missing (NA) values. This must be one of the strings "auto" (default), "yes", or "no". The default value "auto" means that the software will perform that check only when the input expression data is provided as a base matrix, an ExpressionSet or a SummarizedExperiment object, while every other type of input expression data container (e.g., SingleCellExperiment, etc.) will not be checked. If checkNA="yes", then the input expression data will be checked for missing values irrespective of the object class of the data container, and if checkNA="no", then that check will not be performed.
use	Character string specifying a policy for dealing with missing values (NAs) in the input expression data argument exprData. It only applies when either checkNA="yes", or checkNA="auto" (see the checkNA parameter. The argument value must be one of the strings "everything" (default), "all.obs", or "na.rm". The policy of the default value "everything" consists of propagating NAs so that the resulting enrichment scores will be NA, whenever one or more of its contributing values is NA, giving a warning when that happens. When use="all.obs", the presence of NAs in the input expression data will produce an error. Finally, when use="na.rm", NA values in the input expression data will be removed from calculations, giving a warning when that happens, and giving an error if no values are left after removing the NA values.
x	An object of class ssgseaParam.
recursive	Not used with x being an object of class ssgseaParam.

Details

In addition to the two common parameter slots inherited from [GsvaMethodParam], this class has slots for the two method-specific parameters of the ssGSEA method described below.

In addition to an expression data set and a collection of gene sets, ssGSEA takes two method-specific parameters as described below.

Value

A new ssgseaParam object.

Slots

`alpha` Numeric vector of length 1. The exponent defining the weight of the tail in the random walk performed by the ssGSEA (Barbie et al., 2009) method.

`normalize` Logical vector of length 1. If TRUE runs the ssGSEA method from Barbie et al. (2009) normalizing the scores by the absolute difference between the minimum and the maximum, as described in their paper. Otherwise this last normalization step is skipped.

`checkNA` Character string. One of the strings "auto" (default), "yes", or "no", which refer to whether the input expression data should be checked for the presence of missing (NA) values.

`didCheckNA` Logical vector of length 1, indicating whether the input expression data was checked for the presence of missing (NA) values.

`anyNA` Logical vector of length 1, indicating whether the input expression data contains missing (NA) values.

`use` Character string. One of the strings "everything" (default), "all.obs", or "na.rm", which refer to three different policies to apply in the presence of missing values in the input expression data; see ssgseaParam.

References

Barbie, D.A. et al. Systematic RNA interference reveals that oncogenic KRAS-driven cancers require TBK1. *Nature*, 462(5):108-112, 2009. DOI

See Also

GsvaExprData, GsvaGeneSets, GsvaMethodParam, plageParam, zscoreParam, gsvaParam

Examples

```
library(GSVA)
library(GSVAdata)

data(leukemia)
data(c2BroadSets)

## for simplicity, use only a subset of the sample data
ses <- leukemia_eset[1:1000, ]
gsc <- c2BroadSets[1:100]
sp1 <- ssgseaParam(ses, gsc)
```

sp1

---

zscoreParam-class zscoreParam *class*

Description

Method-specific parameters for the combined z-scores method.

Objects of class zscoreParam contain the parameters for running the combined z-scores method.

Usage

```
zscoreParam(
  exprData,
  geneSets,
  assay = NA_character_,
  annotation = NA_character_,
  minSize = 1,
  maxSize = Inf
)
```

Arguments

exprData	The expression data. Must be one of the classes supported by GsvaExprData. Type help(GsvaExprData) to consult the available classes.
geneSets	The gene sets. Must be one of the classes supported by GsvaGeneSets.
assay	The name of the assay to use in case exprData is a multi-assay container, otherwise ignored. By default, the first assay is used.
annotation	The name of a Bioconductor annotation package for the gene identifiers occurring in the row names of the expression data matrix. This can be used to map gene identifiers occurring in the gene sets if those are provided in a GeneSetCollection. By default gene identifiers used in expression data matrix and gene sets are matched directly.
minSize	Minimum size of the resulting gene sets after gene identifier mapping. By default, the minimum size is 1.
maxSize	Maximum size of the resulting gene sets after gene identifier mapping. By default, the maximum size is Inf.

Details

Since this method does not take any method-specific parameters, the parameter class does not add any slots to the common slots inherited from GsvaMethodParam.

The combined z-scores method does not take any method-specific parameters in addition to an expression data set and a collection of gene sets.

Value

A new zscoreParam object.

References

Lee, E. et al. Inferring pathway activity toward precise disease classification. *PLoS Comp Biol*, 4(11):e1000217, 2008. DOI

See Also

GsvaExprData, GsvaGeneSets, GsvaMethodParam, plageParam, ssgseaParam, gsvaParam

Examples

```
library(GSVA)
library(GSVAdata)

data(leukemia)
data(c2BroadSets)

## for simplicity, use only a subset of the sample data
ses <- leukemia_eset[1:1000, ]
gsc <- c2BroadSets[1:100]
zp1 <- zscoreParam(ses, gsc)
zp1
```

Index