Package ‘roastgsa’

May 18, 2024

Type Package
Version 1.2.0
Date 2023-06-13
Title Rotation based gene set analysis
BugReports https://github.com/adricaba/roastgsa/issues
Description This package implements a variety of functions useful for gene set analysis using rotations to approximate the null distribution. It contributes with the implementation of seven test statistic scores that can be used with different goals and interpretations. Several functions are available to complement the statistical results with graphical representations.

Encoding UTF-8
VignetteBuilder knitr
biocViews Microarray, Preprocessing, Normalization, GeneExpression, Survival, Transcription, Sequencing, Transcriptomics, Bayesian, Clustering, Regression, RNASeq, MicroRNAArray, mRNAMicroarray, FunctionalGenomics, SystemsBiology, ImmunoOncology, DifferentialExpression, GeneSetEnrichment, BatchEffect, MultipleComparison, QualityControl, TimeCourse, Metabolomics, Proteomics, Epigenetics, Cheminformatics, ExonArray, OneChannel, TwoChannel, ProprietaryPlatforms, CellBiology, BiomedicalInformatics, AlternativeSplicing, DifferentialSplicing, DataImport, Pathways

Depends R (>= 4.3.0)
Imports parallel, grDevices, graphics, utils, stats, methods, grid, RColorBrewer, gplots, ggplot2, limma, Biobase
Suggests BiocStyle, knitr, markdown, GSEABenchmarkeR, EnrichmentBrowser, preprocessCore, DESeq2
License GPL-3
git_url https://git.bioconductor.org/packages/roastgsa
git_branch RELEASE_3_19
Description

from dragtable v1.0 of Dan Vanderkam.

Usage

dragtable

Format

character vector

Value

Character vector with dragtable
expr.tcg

Source
http://danvk.org/dragtable/

References
kryogenix.org/code/browser/sortable

expr.tcg  Tumor Bladder TCGA data

Description
Counts matrix of RNA-seq study with 19 tumor Bladder Urothelial Carcinoma samples and 19 adjacent healthy tissues

Usage
expr.tcg

Format
matrix

Value
Matrix with expression matrix

Source

References
Tumor Bladder TCGA data

Description
Gene information of RNA-seq study with 19 tumor Bladder Urothelial Carcinoma samples and 19 adjacent healthy tissues

Usage
fd.tcga

Format
DFrame

Value
Data frame with gene symbols

Source

References

Hallmarks homo sapiens gene symbol

Description
Hallmark geneset collection from msigdb

Usage
hallmarks.hs

Format
character list
Value

List with hallmark genes

Source

https://www.gsea-msigdb.org/gsea/downloads.jsp

References


Description

Heatmap showing sample variation for either genes (in a particular gene set) or summarized gene signatures.

Usage

heatmaprgsa_hm(obj, y, intvar, adj.var = NULL, whplot = 1, toplot = TRUE, pathwaylevel = FALSE, mycol = c("black","orange","green","white"), sample2zero = FALSE, rgsa.like=FALSE, psel = NULL, dendrogram = "n", col= bluered(100), trace='none', notecol='black', notecex=1, keysize=.9, cexCol=1.5, Rowv = NULL, Colv = FALSE, las =2, fdrkey = FALSE, quantile.sat = 0.95, order1= NULL, order2 = NULL, sizex =8, sizey =5, ...)

Arguments

obj an object of class 'roastgsa'
y data used for roastgsa
intvar name of variable of interest in obj$formula. If missing, last term of obj$formula is used
adj.var name of covariates in obj$design to adjust using a linear model in the heatmap representation. If NULL no prior adjustment applies
whplot selected pathway. If integer vector, the pathways are selected in the same order as the table in obj$res
toplot whether to plot the heatmap or just return the adjusted expression matrix
pathwaylevel If TRUE, the heatmap shows the variation at the pathway level. Otherwise, the heatmap shows the variation of all genes in the selected pathways.
mycol color for heatmap columns defining the groups of the variable of interest
sample2zero  Only applicable for obj$statistic = "maxmean". If TRUE, expression of genes, whose moderated-t sign is contrary to the roastgsa score, is set to zero for all samples (as part of the maxmean strategy).

rgsa.like  apply roastgsa transformations of data (restandarization and set.statistic operations) samplewise (see details below).

psel  character vector with probesets (one per gene) to be used for roastgsa statistic in a microarray experiment.

dendrogram  heatmap.2 parameter. Character string indicating whether to draw 'none', 'row', 'column' or 'both' dendrograms. Defaults to 'n'.

col  heatmap.2 parameter. Colors used for the image.

trace  heatmap.2 parameter. Character string indicating whether a solid "trace" line should be drawn across 'row's or down 'column's, 'both' or 'none'. The distance of the line from the center of each color-cell is proportional to the size of the measurement.

notecol  heatmap.2 parameter. Color of note.

notecex  heatmap.2 parameter. Size of note.

keysize  heatmap.2 parameter. Numeric value indicating the size of the key.

cexCol  heatmap.2 parameter. Cex.axis in for the column axis labeling.

Rowv  heatmap.2 parameter. Determines if and how the row dendrogram should be reordered.

Colv  heatmap.2 parameter. Determines if and how the col dendrogram should be reordered.

las  orientation of x axis.

fdrkey  if TRUE, the BH adjusted p-value for every pathway tested is printed in the plot. Only considered when pathwaylevel = TRUE.

quantile.sat  numeric between 0.5 and 1 used to saturate high values at such specified quantile (used to avoid extreme values in the visualization).

order1  genes order. If NULL its ordered based on the moderated-t statistics.

order2  samples order. If NULL its ordered using the information of the variable of interest.

sizex  size of x axis.

sizey  size of y axis.

...  Arguments passed to or from other methods to the low level.

Details

This heatmap considers $n + 1$ columns ($n$ being the sample size). The first column represents the moderated-t statistic (or a restandarization of the same in case of competitive testing). The other columns confine the expression data scaled by the standard error of the estimated coefficient in the model and centered (if rgsa.like = TRUE). In such case, the cross product of all data columns and the design matrix equals the first column of the heatmap, and the average of the first column of the heatmap equals the observed roastgsa test statistic (at least when the set.statistic used is either mean or maxmean).
Value

a data.frame object with source data for heatmap representation

Author(s)

Adria Caballe Mestres

References


See Also

roastgsa and plotStats and plotGSEA

Examples

```r
y <- array(rnorm(10000),dim = c(1000,10))
covar <- data.frame(oui = factor(c(rep(0,5),rep(1,5))))
colnames(y) <- rownames(covar) <- paste0("sample",1:10)
rownames(y) <- paste0("gene",1:1000)
form <- as.formula(~ oui)
index <- lapply(1:10, function(o) sample(1:1000,50))
names(index) <- paste0("gset",1:10)

roastgsa1 <- roastgsa(y, covar, form = form, self.contained = TRUE,
                     set.statistic = "maxmean", index = index, nrot = 200,
                     mccores = 1, execution.info = FALSE)

heatmaprgsa_hm(roastgsa1, y, intvar = "oui", whplot = 1, toplot = TRUE,
                pathwaylevel = FALSE, mycol = c("black","orange","green","white"),
                sample2zero = FALSE)

heatmaprgsa_hm(roastgsa1, y, intvar = "oui", whplot = 1:10, toplot = TRUE,
                pathwaylevel = TRUE, mycol = c("black","orange","green","white"),
                sample2zero = FALSE)
```

htmlrgsa

写作html document with roastgsa output
Usage

htmlrgsa(obj, htmlpath = "", htmlname = "file.html", plotpath = "", plotstats = TRUE, plotgsea = TRUE, indheatmap = TRUE, ploteffsize = TRUE, links_plots = list(stats=NULL, gsea=NULL, heatmap=NULL, effsize=NULL), y, whplots = NULL, geneDEhtmlfiles = NULL, tit = ", margins = c(5,16), sizesHeatmap = c(1200, 800), typeheatmap = c("heatmap.2","ggplot2"), intvar, adj.var = NULL, mycol, varrot, psel = NULL, sortable, dragtable, ...)
htmlrgsa

psel character vector with probesets (one per gene) to be used for roastgsa statistic in a microarray experiment

sortable internal data loaded with roastgsa package. Permits sorting columns in html tables.

dragtable internal data loaded with roastgsa package. Permits dragging elements in html tables.

... Arguments passed to or from other methods to the low level.

Details

This function permits to explore a html-table with the statistical results and graphical representation of the top gene sets obtained from an object of class roastgsa.

By default four plots are considered for each gene set of interest: plotStats, plotGSEA, heatmaprgsa_hm and plotEffsSignatureSize. The first three can be computed from the 'roastgsa' object, whereas for plotEffsSignatureSize, an object of class 'varrotrand' (see varrotrand) with the estimated rotation score variances for randomly selected gene sets of several sizes has to be defined at first.

Value

It saves an html table with the main results of the roastgsa hypothesis testing.

Author(s)

Adria Caballe Mestres

References


See Also

roastgsa

Examples

data(sorttable)
data(dragtable)

y <- array(rnorm(10000),dim = c(1000,10))
covar <- data.frame(voi = factor(c(rep(0,5),rep(1,5))))
colnames(y) <- rownames(covar) <- paste0("sample",1:10)
rownames(y) <- paste0("gene",1:1000)
form <- as.formula("~ voi")
index <- lapply(1:10, function(o) sample(1:1000,50))
names(index) <- paste0("gset",1:10)

roastgsa1 <- roastgsa(y, covar, form = form, self.contained = TRUE,
set.statistic = "maxmean", index = index, nrot = 200,
Description

KEGG genesets obtained with limma function getGeneKEGGLinks

Usage

kegg.hs

Format

character list

Value

List with KEGG genes

Source

https://www.kegg.jp/kegg/rest/keggapi.html

References

pd.tcga  

Tumor Bladder TCGA data

Description
Sample information of RNA-seq study with 19 tumor Bladder Urothelial Carcinoma samples and 19 adjacent healthy tissues

Usage
pd.tcga

Format
DFrame

Value
Data frame with sample info

Source

References

plot.roastgsa  

roastgsa plot

Description
Plot for roastgsa objects

Usage
## S3 method for class 'roastgsa'
plot(x, type = c("stats","GSEA"), whplot = 1,
      maintitle = "", gsainfo = TRUE, cex.sub = 0.8, lwd = 2, ...)

Arguments

x an object of class 'roastgsa'
type plot type, either 'stats' or 'GSEA'
whplot selected pathway. If integer vector, the pathways are selected in the same order as observed in the obj$res table
maintitle plot main title. If maintitle == "", the name of the pathway in obj is printed
gsainfo if TRUE, the subtitle shows the GSA main results
cex.sub cex for subtitle
lwd line width
... Arguments passed to or from other methods to the low level.

Details

Details for using 'type = stats' in the plot are given in plotStats. Details for using 'type = GSEA' in the plot are given in plotGSEA.

Value

plot object with the graphical representation of roastgsa results.

Author(s)

Adria Caballe Mestres

References


See Also

roastgsa and plotStats

Examples

y <- array(rnorm(10000),dim = c(1000,10))
covar <- data.frame(voi = factor(c(rep(0,5),rep(1,5))))
colnames(y) <- rownames(covar) <- paste0("sample",1:10)
rownames(y) <- paste0("gene",1:1000)
form <- as.formula("~ voi")
index <- lapply(1:10, function(o) sample(1:1000,50))
names(index) <- paste0("gset",1:10)

roastgsa1 <- roastgsa(y, covar, form = form, self.contained = TRUE,
   set.statistic = "maxmean", index = index, nrot = 200,
   mcores = 1, execution.info = FALSE)
plot(roastgsa1, type = "stats", whplot = 1, gsainfo =TRUE, maintitle =
plot.ssGSA

"", statistic = "mean")

---

plot.ssGSA

Plot single sample Gene Set Analysis

Description

Scatter plot of single sample z-score summarized data

Usage

## S3 method for class 'ssGSA'
plot(x, orderby, whplot = 1, col = "black", samplename = FALSE, 
maintitle = "", ssgsaInfo = TRUE, cex.sub = 0.8, ...)

Arguments

x object of class 'ssGSA'
orderby numeric or factor vector of the same size and order of data columns used for ssGSA. It sets the x-axis of the plot
whplot selected pathway. If integer vector, the pathways are selected in the same order as the table in x$res
col color of scatterplot points
samplename whether to show or not the names of the samples instead of points
maintitle plot main title. If maintitle = "", the name of the pathway in obj is printed
ssgsaInfo if TRUE, the subtitle shows the ssGSA results
cex.sub cex for subtitle
... Arguments passed to or from other methods to the low level.

Details

This graphic is a great alternative to explore gene set variation at sample level. This is sometimes ignored when doing GSEA, where classic representations (e.g., plotGSEA) show gene variation after averaging out the sample differences within each experimental condition.

Value

plot object with the graphical representation of ssGSA results

Author(s)

Adria Caballe Mestres
References


See Also

ssGSA

Examples

```r
y <- array(rnorm(10000), dim = c(1000,10))
covar <- data.frame(voi = factor(c(rep(0,5),rep(1,5))))
colnames(y) <- rownames(covar) <- paste0("sample",1:10)
rownames(y) <- paste0("gene",1:1000)
form <- as.formula(~ voi)
index <- lapply(1:10, function(o) sample(1:1000,50))
names(index) <- paste0("gset",1:10)
design <- model.matrix(form, covar)

ssgsa1 <- ssGSA(y, obj=NULL, design = design, contrast = 2, index = index,
    method = c("GScor"))
plot(ssgsa1, orderby = covar$voi, whplot = 1)
```

Description

Approximation of effective signature size under gene randomization

Usage

```r
ploteffsignaturesize(obj, varrot, whplot = 1, ...)
```

Arguments

- **obj**: an object of class `roastgsa`
- **varrot**: an object of class `varrotrand` (see `varrotrand`) with estimated rotation score variances for randomly selected genesets of several sizes.
- **whplot**: selected pathway. If integer vector, the pathways are selected in the same order as the table in `obj$res`
- **...**: Arguments passed to or from other methods to the low level.
Details

The plot shows the approximated probability of obtaining a test statistic variance (under rotations of the residual space of the data) as extreme as the observed when generating randomly gene sets of several sizes.

Value

plot object with the effective signature size representation of roastgsa results

Author(s)

Adria Caballe Mestres

References


See Also

varrotrand and roastgsa

Examples

```r
y <- array(rnorm(10000),dim = c(1000,10))
covar <- data.frame(voi = factor(c(rep(0,5),rep(1,5))))
colnames(y) <- rownames(covar) <- paste0("sample",1:10)
rownames(y) <- paste0("gene",1:1000)
form <- as.formula("~ voi")
index <- lapply(1:10, function(o) sample(1:1000,50))
names(index) <- paste0("gset",1:10)

roastgsa1 <- roastgsa(y, covar, form = form, self.contained = TRUE,
set.statistic = "maxmean", index = index, nrot = 100,
mccores = 1, execution.info = FALSE)

varrot <- varrotrand(roastgsa1, y,
    testedsizes = c(seq(5,50, by=5), seq(55,200,by=10)),
nrep = 50)
ploteffsignaturesize(roastgsa1, varrot, whplot = 2)
```
plotGSEA

GSEA plot

Description
GSEA plot for roastgsa objects

Usage

plotGSEA(obj, whplot = 1, maintitle = "", gsainfo = TRUE, cex.sub = 0.8, lwd = 2, ...)

Arguments

obj an object of class 'roastgsa'
whplot selected pathway. If integer vector, the pathways are selected in the same order as observed in the obj$res table
maintitle plot main title. If maintitle == "", the name of the pathway in obj is printed
gsainfo if TRUE, the subtitle shows the GSA main results
cex.sub cex for subtitle
lwd line width
... Arguments passed to or from other methods to the low level.

Details
Standard representation of Kolmogorov-Smirnov GSEA enrichment score.

Value
plot object with the GSEA representation of roastgsa results

Author(s)
Adria Caballe Mestres

References


See Also

roastgsa and plotStats
Examples

```r
y <- array(rnorm(10000), dim = c(1000, 10))
covar <- data.frame(voi = factor(c(rep(0, 5), rep(1, 5))))
colnames(y) <- rownames(covar) <- paste0("sample", 1:10)
rownames(y) <- paste0("gene", 1:1000)
form <- as.formula(~ voi)
index <- lapply(1:10, function(o) sample(1:1000, 50))
names(index) <- paste0("gset", 1:10)

roastgsa1 <- roastgsa(y, covar, form = form, self.contained = TRUE,
                         set.statistic = "maxmean", index = index, nrot = 200,
                         mccores = 1, execution.info = FALSE)
plotGSEA(roastgsa1, whplot = 1, gsainfo = TRUE, maintitle = 
            ", statistic = "mean")
```

plotStats

**General GSA plot**

**Description**

General gene set analysis plot showing the ordered moderated-t statistics for the selected pathway

**Usage**

```r
plotStats(obj, whplot = 1, maintitle = ", statistic = "mean",
          ylimAll = TRUE, ylim = NULL, minpointsDens = 20,
          gsainfo = TRUE, cex.sub = 0.8, lwd = 2, ...)
```

**Arguments**

- `obj` an object of class 'roastgsa'
- `whplot` selected pathway. If integer vector, the pathways are selected in the same order as the table in obj$res
- `maintitle` plot main title. If maintitle = "", the name of the pathway in obj is printed
- `statistic` to be selected from 'mean' or 'median'
- `ylimAll` y limits are found using data from all genesets (if TRUE) or using data from only the plotted geneset (if FALSE). Only if ylim = NULL
- `ylim` vector of size two with y limits
- `minpointsDens` minum number of genes needed to draw the density plot
- `gsainfo` if TRUE, the subtitle shows the enrichment results
- `cex.sub` cex for subtitle
- `lwd` line width
- `...` Arguments passed to or from other methods to the low level.
Details

The statistic argument is used for competitive testing computations of restandardized moderated-
t statistics. If "median", the median of all stats is used for centering and the median absolute
deviation is used for scaling. If "mean", standard normalization applies.
It shows the ordered moderated t-statistics in various formats, area for up- and down- expressed
genes, barcode plot for these ordered values and density.

Value

plot object with a general representation of roastgsa results

Author(s)

Adria Caballe Mestres

References

Berenguer Llergo, Camille Stephan-Otto Attolini bioRxiv 2021.03.23.436604;

See Also

roastgsa and plotGSEA

Examples

y <- array(rnorm(10000),dim = c(1000,10))
covar <- data.frame(oui = factor(c(rep(0,5),rep(1,5))))
colnames(y) <- rownames(covar) <- paste0("sample",1:10)
rownames(y) <- paste0("gene",1:1000)
form <- as.formula("~ oui")
index <- lapply(1:10, function(o) sample(1:1000,50))
names(index) <- paste0("gset",1:10)

roastgsa1 <- roastgsa(y, covar, form = form, self.contained = TRUE,
set.statistic = "maxmean", index = index, nrot = 200,
mccores = 1, execution.info = FALSE)
plotStats(roastgsa1, whplot = 1, maintitle = "general plot", statistic =
"mean")

roastgsa Rotation-based Gene Set Analysis

Description

Gene set analysis using rotations for hypothesis testing. Test statistic options include KS-based
statistics used in GSEA or GSVA as well as summary statistics such as mean, maxmean, median,
absmean and mean.rank
Usage

roastgsa(y, covar, form, contrast = NA, design = NULL, gsetsel, gspath, index = NULL, self.contained = FALSE, set.statistic = "maxmean", psel = NULL, nrot = 9999, minsize = 10, maxsize = 500, mccores = 1, execution.info = TRUE, weights = NULL, shrink.resid = TRUE, normalizeScores = TRUE, ...)  

Arguments

y expression matrix with columns indicating samples and rows indicating genes  
covar data frame with the covariates  
form description of the model to be fitted  
contrast comparison to consider in the model. If NA, the last column of the design matrix is used  
design the design matrix of the experiment. If null, this is calculated using the form and the covar arguments  
gsetsel character string with gene set database to be used in format .gmt. If missing, index argument has to be provided  
gspath path for the gene set database  
index list with index vectors specifying which rows of y are in the testing sets. Either integer indexes with row positions or gene identifiers can be stated. If NULL, the index is computed using information in the gsetsel and gspath arguments  
self.contained competitive test (FALSE) or self contained test (TRUE)  
set.statistic to be chosen from "maxmean" (default), "mean", "mean.rank", "median", "absmean", "GSEA" and "GSVA"  
psel character vector with probesets (one per gene) to be used for roastgsa statistic in a microarray experiment  
nrot number of rotations used for hypothesis testing  
minsize minimum size of the testing sets allowed for hypothesis testing  
maxsize maximum size of the testing sets allowed for hypothesis testing  
mccores the number of cores to use for parallel executions  
execution.info Show (if set to TRUE) the progress-bar of the iterative process  
weights list with the gene weights in each testing set. Only for set.statistic = "maxmean" and "mean". If NULL, weights are assumed to be constant  
shrink.resid if TRUE, the coefficients of the linear model are shrunk towards zero for rotations to increase the power  
normalizeScores transform the moderated t-statistics to z-scores  
... Arguments passed to or from other methods to the low level.
Details

We consider 7 different enrichment score functions which we refer by the names of mean, maxmean, median, absmean, mean.rank, GSEA and GSVA. The first four functions (mean, maxmean, median, absmean) are formulated for the two type of testing problems (self-contained and competitive). The mean.rank, GSEA and GSVA are exclusive scores for the competitive approach. The absmean is a non-directional score that can be used to give priority to gene sets with both activator and inhibitor genes. The mean is a democratic score that gives priority to detecting gene sets in which a large fraction of their genes present similar effect sizes going at the same direction. The maxmean (default) falls in between the mean and the absmean scores, being capable to recover both type of gene sets consistently.

Some of the defined sets are composed by genes that interact together in any particular biological condition, leading to intra-gene set correlation structures with high levels of correlation. We encourage the usage of effective signatures size, that can be a proxy for the number of uncorrelated genes in the gene set used for GSA (varrotrand and ploteffsignatureize). Through the argument weights, we provide the possibility to redefining the gene set by weighting the importance of each gene in the list.

GSEA and GSVA scores are computationally much more intensive than the other scores.

Value

return an object of class roastgsa with attributes

"res" data.frame with main results obtained in hypothesis testing. Total genes in the geneset, the number of genes also in the y, the test statistic, the normalized score and the significance of the tests

"stats" Moderated t-statistics for all genes

"contrast" contrast used in a vector form

"index" list with gene set symbols

Author(s)

Adria Caballe Mestres

References


See Also

roast
Examples

```r
y <- array(rnorm(10000), dim = c(1000, 10))
covar <- data.frame(voi = factor(c(rep(0, 5), rep(1, 5))))
colnames(y) <- rownames(covar) <- paste0("sample", 1:10)
rownames(y) <- paste0("gene", 1:1000)
form <- as.formula("~ voi")
index <- lapply(1:10, function(o) sample(1:1000, 50))
names(index) <- paste0("gset", 1:10)

roastgsa1 <- roastgsa(y, covar, form = form, self.contained = TRUE,
                      set.statistic = "maxmean", index = index, nrot = 200,
                      mccores = 1, execution.info = FALSE)
print(roastgsa1)
```

Description

from sorttable v2.0 of Stuart Langridge.

Usage

sortable

Format

character vector

Value

Character vector with sortable

Source

http://www.kryogenix.org/code/browser/sorttable/

References

http://www.kryogenix.org/code/browser/sorttable/
**ssGSA**

*Single sample Gene Set Analysis*

**Description**

Single sample gene set analysis using z-score summarized data for linear model hypothesis testing

**Usage**

```r
ssGSA(y, obj = NULL, design = NULL, contrast = NULL, index = NULL, method = c("GScor","GSadj","zscore"))
```

**Arguments**

- `y` expression matrix with columns indicating samples and rows indicating genes
- `obj` object of class 'roastgsa' used to extract the design, the contrast and the index arguments
- `design` the design matrix of the experiment. Considered only if `obj` is NULL
- `contrast` comparison to consider in the model. Considered only if `obj` is NULL
- `index` list with index vectors specifying which rows of `y` are in the testing sets. Either integer indexes with row positions or gene identifiers can be stated. Considered only if `obj` is NULL
- `method` If "GSadj", a correction variable with the average trend in the data enters in the model as confounding variable. If "GScor", gene signatures are adjusted a priori by subtracting the correction variable values. Check details for more information.

**Details**

A correction by the overall tendency can be done a priori (GScor) or it can be incorporated as a covariate in the linear model (GSadj). The correction variable used here is what we have called the global signature (GS) of the experiment, that for each sample can be calculated as the average z-score of all genes measured in `y`. This GS corrects or centers global technical / sampling directions in the data.

**Value**

return an object of class ssGSA with attributes

- "res" data.frame with main results obtained in hypothesis testing. Total genes in the gene set, the average score, the test statistic, p-value and adjusted p-value.
- "stats" adjusted z-scores matrix

**Author(s)**

Adria Caballe Mestres
References


See Also

plot.ssGSA

Examples

```r
y <- array(rnorm(10000),dim = c(1000,10))
covar <- data.frame(loi = factor(c(rep(0,5),rep(1,5))))
colnames(y) <- rownames(covar) <- paste0("sample",1:10)
rownames(y) <- paste0("gene",1:1000)
form <- as.formula("~ loi")
index <- lapply(1:10, function(o) sample(1:1000,50))
names(index) <- paste0("gset",1:10)
design <- model.matrix(form, covar)

ssgsa1 <- ssGSA(y, obj=NULL, design = design, contrast = 2, index = index,
method = c("GScor"))

varrotrand <- varrotrand(obj = ssgsa1, y, testedsizes = c(3:30,seq(32,50, by=2), seq(55,200,by=5)),
nrep = 200, nrot = NULL,
mccores = NULL, psel = NULL)
```

Description

Computation of the sample variance of rotation scores under gene randomization

Usage

```r
varrotrand(obj, y, testedsizes = c(3:30,seq(32,50, by=2),
seq(55,200,by=5)), nrep = 200, nrot = NULL,
mccores = NULL, psel = NULL)
```

Arguments

- `obj` an object of class 'roastgsa'
- `y` data used in roastgsa call
- `testedsizes` effective sizes to be tested
- `nrep` number of randomly selected gene sets created for each tested effective size
- `nrot` number of rotations used for hypothesis testing
- `mccores` the number of cores to use for parallel executions
- `psel` character vector with probesets (one per gene) to be used for roastgsa statistic in a microarray experiment
Details
When a specific gene that is highly correlated to the rest of the gene set finds an extreme value, even under $H_0$, it is likely that many other genes in the gene set follow it with large values as well. We define the concept of effective signature size of a gene set by the number of randomly selected (not necessarily independent) genes that are needed to achieve comparable variability levels on rotation summary test statistics. This can be viewed as a realistic measure of the total number of independent variables that contribute to the power of the test. The function presented here computes the sample variance of the rotation scores in randomly generated signatures of several sizes. The comparison to the observed variances (using the testing gene sets in the roastgsa call) is done through the function `ploteffsignaturesize`.

Value
return an object of class varrotrand with attributes
- "varrot" matrix nrep x testedsizes with the estimated variance of the rotation scores using nrot rotations
- "testedsizes" effective sizes being tested
- "nrep" number of gene sets created for each tested effective size

Author(s)
Adria Caballe Mestres

References

See Also
`ploteffsignaturesize` to visualize results and `roastgsa` for gsa approach

Examples
```r
y <- array(rnorm(10000),dim = c(1000,10))
covar <- data.frame(loi = factor(c(rep(0,5),rep(1,5))))
colnames(y) <- rownames(covar) <- paste0("sample",1:10)
rownames(y) <- paste0("gene",1:1000)
form <- as.formula("~ loi")
index <- lapply(1:10, function(o) sample(1:1000,50))
names(index) <- paste0("gset",1:10)

roastgsa1 <- roastgsa(y, covar, form = form, self.contained = TRUE, 
set.statistic = "maxmean", index = index, nrot = 100, 
mcores = 1, execution.info = FALSE)

varrot <- varrotrand(roastgsa1, y, 
testedsizes = c(seq(5,50, by=5), seq(55,200,by=10)),
```
varrotrand

n rep = 50)
# Index

* datasets
  - dragtable, 2
  - expr.tcga, 3
  - fd.tcga, 4
  - hallmarks.hs, 4
  - kegg.hs, 10
  - pd.tcga, 11
  - sorttable, 21

  dragtable, 2
  expr.tcga, 3
  fd.tcga, 4
  hallmarks.hs, 4
  heatmap.2, 6
  heatmaprgsa.hm, 5, 8, 9
  htmlrgsa, 7
  kegg.hs, 10
  pd.tcga, 11
  plot.roastgsa, 11
  plot.ssGSA, 13, 23
  plotEffSignaturesize, 8, 9, 14, 20, 24
  plotGSEA, 7–9, 12, 13, 16, 18
  plotStats, 7–9, 12, 16, 17
  print.roastgsa(roastgsa), 18
  print.ssGSA(ssGSA), 22

  roast, 20
  roastgsa, 7–9, 12, 15, 16, 18, 23, 24

  sorttable, 21
  ssGSA, 14, 22

  varrotrand, 8, 9, 14, 15, 20, 23