Package ‘GSAR’

April 23, 2016

Type Package

Title Gene Set Analysis in R

Version 1.4.0

Date 2015-07-14

Author Yasir Rahmatallah <`yrahmatallah@uams.edu`>, Galina Glazko <`gvglazko@uams.edu`>

Maintainer Yasir Rahmatallah <`yrahmatallah@uams.edu`>, Galina Glazko <`gvglazko@uams.edu`>

Depends R (>= 3.0.1), igraph (>= 0.7.0)

Suggests MASS, GSVAdata, ALL, tweeDEseqCountData, GSEABase, annotate,
org.Hs.eg.db, Biobase, genefilter, hgu95av2.db, edgeR,
BiocStyle

LazyData yes

biocViews Software, StatisticalMethod, DifferentialExpression

Description Gene set analysis using specific alternative hypotheses. Tests for differential expression, scale and net correlation structure.

License GPL (>=2)

NeedsCompilation no

R topics documented:

GSAR-package *Gene Set Analysis in R*

Description

Package GSAR provides a set of statistical methods for self-contained gene set analysis. It consists of two-sample multivariate nonparametric statistical methods to test a null hypothesis against specific alternative hypotheses, such as differences in shift (function `KStest`), scale (function `RKStest`) or correlation structure (function `GSNCAtest`) between two conditions. It also offers a graphical visualization tool for correlation networks to examine the change in the net correlation structure of a gene set between two conditions based on the minimum spanning trees (function `plotMST2.pathway`). The methods available in this package were proposed in Rahmatallah et. al. 2014 and Friedman and Rafsky 1979. The performance of different methods available in this package was thoroughly tested using simulated data and microarray datasets in Rahmatallah et. al. 2012 and Rahmatallah et. al. 2014. These methods (except `RKStest`) can also be applied to RNA-Seq count data given that proper normalization is used. Proper normalization must take into account both the within-sample differences (mainly gene length) and between-samples differences (library size or sequencing depth). A recent work examining the performance of the tests in package GSAR using simulated and real RNA-Seq dataset was submitted for publication and will be cited in future versions of this package.

Author(s)

Yasir Rahmatallah <yrahmatallah@uams.edu>, Galina Glazko <gvglazko@uams.edu>

Maintainer: Yasir Rahmatallah <yrahmatallah@uams.edu>, Galina Glazko <gvglazko@uams.edu>

References

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2014) Gene sets net correlations analysis (GSNCA): a multivariate differential coexpression test for gene sets. Bioinformatics **30**, 360–368.

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2012) Gene set analysis for self-contained tests: complex null and specific alternative hypotheses. Bioinformatics **28**, 3073–3080.

Friedman J. and Rafsky L. (1979) Multivariate generalization of the Wald-Wolfowitz and Smirnov two-sample tests. Ann. Stat. **7**, 697–717.

See Also

`igraph`.

findMST2 *Union of the First and Second Minimum Spanning Trees*

Description

Find the union of the first and second minimum spanning trees.

Usage

```
findMST2(object, cor.method="pearson", min.sd=1e-3, return.MST2only=FALSE)
```

Arguments

`object` a numeric matrix with columns and rows respectively corresponding to samples and features.

`cor.method` a character string indicating which correlation coefficient is to be computed. Possible values are "`pearson`" (default), "`spearman`" and "`kendall`".

`min.sd` the minimum allowed standard deviation for any feature. If any feature has a standard deviation smaller than `min.sd` the execution stops and an error message is returned.

`return.MST2only`
logical. If `FALSE` (default) a list of length three containing objects of class `igraph` is returned. The first and second are the first and second MSTs respectively. The third is the union of the first and second, MST2. If `TRUE`, an object of class `igraph` containing the MST2 is returned.

Details

This function produces the union of the first and second minimum spanning trees (MSTs) as an object of class `igraph` (check package `igraph` for details). It can as well return the first and second minimum spanning trees when `return.MST2only` is `FALSE` (default). It starts by calculating the correlation (coexpression) matrix and using it to obtain a weighting matrix for a complete graph using the equation $w_{ij} = 1 - |r_{ij}|$ where r_{ij} is the correlation between features i and j and w_{ij} is the weight of the link between vertices (nodes) i and j in the graph $G(V, E)$.

For the graph $G(V, E)$ where V is the set of vertices and E is the set of edges, the first MST is defined as the acyclic subset $T_1 \subseteq E$ that connects all vertices in V and whose total length $\sum_{i,j \in T_1} d(v_i, v_j)$ is minimal (Rahmatallah et. al. 2014). The second MST is defined as the MST of the reduced graph $G(V, E - T_1)$. The union of the first and second MSTs is denoted as MST2.

It was shown in Rahmatallah et. al. 2014 that MST2 can be used as a graphical visualization tool to highlight the most highly correlated genes in the correlation network. A gene that is highly correlated with all the other genes tends to occupy a central position and has a relatively high degree in the MST2 because the shortest paths connecting the vertices of the first and second MSTs tend to pass through the vertex corresponding to this gene. In contrast, a gene with low intergene correlations most likely occupies a non-central position in the MST2 and has a degree of 2.

In rare cases, a feature may have a constant or nearly constant level across the samples. This results in a zero or a tiny standard deviation. Such case produces an error in command `cor` used to compute the correlations between features. To avoid this situation, standard deviations are checked in advance and if any is found below the minimum limit min.sd (default is 1e-3), the execution stops and an error message is returned indicating the the number of feature causing the problem (if only one the index of that feature is given too).

Value

If return.MST2only is TRUE, findMST2 returns an object of class igraph containing the MST2. If return.MST2only is FALSE (default), findMST2 returns a list with the following components:

MST2	an object of class igraph containing the union of the first MST and second MST.
first.mst	an object of class igraph containing the first MST.
second.mst	an object of class igraph containing the second MST.

Author(s)

Yasir Rahmatallah and Galina Glazko

References

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2014) Gene sets net correlations analysis (GSNCA): a multivariate differential coexpression test for gene sets. Bioinformatics **30**, 360–368.

See Also

GSNCAtest, plotMST2.pathway.

Examples

```
## generate a dataset of 20 features and 20 samples
## use multivariate normal distribution with different covariance matrices
library(MASS)
ngenes <- 20
nsamples <- 20
zero_vector <- array(0,c(1,ngenes))
## create a covariance matrix with high off-diagonal elements
## for the first 5 features and low for the remaining 15 features
cov_mtrx <- diag(ngenes)
cov_mtrx[!diag(ngenes)] <- 0.1
mask <- diag(ngenes/4)
mask[!diag(ngenes/4)] <- 0.6
cov_mtrx[1:(ngenes/4),1:(ngenes/4)] <- mask
gp <- mvrnorm(nsamples, zero_vector, cov_mtrx)
dataset <- aperm(gp, c(2,1))
## findMST2 returns a list of length 3
## trees[[1]] is an object of class igraph containing the MST2
trees <- findMST2(dataset)
```

GSNCAtest *Gene Sets Net Correlations Analysis*

Description

Performs Gene Sets Net Correlation Analysis (GSNCA) test to detect differentially coexpressed gene sets.

Usage

```
GSNCAtest(object, group, nperm=1000, cor.method="pearson", check.sd=TRUE,
min.sd=1e-3, max.skip=10)
```

Arguments

`object` a numeric matrix with columns and rows respectively corresponding to samples and features.

`group` a numeric vector indicating group associations for samples. Possible values are 1 and 2.

`nperm` number of permutations used to estimate the null distribution of the test statistic. If not given, a default value 1000 is used.

`cor.method` a character string indicating which correlation coefficient is to be computed. Possible values are "`pearson`" (default), "`spearman`" and "`kendall`".

`check.sd` logical. Should the standard deviations of features checked for small values before the intergene correlations are computed? Default is TRUE (recommended).

`min.sd` the minimum allowed standard deviation for any feature. If any feature has a standard deviation smaller than `min.sd` the execution stops and an error message is returned.

`max.skip` maximum number of skipped random permutations which yield any feature with a standard deviation less than `min.sd`.

Details

This function performs the Gene Sets Net Correlations Analysis (GSNCA), a two-sample nonparametric multivariate differential coexpression test that accounts for the correlation structure between features (genes). The test assigns weight factors to features under one condition and adjust these weights simultaneously such that equality is achieved between each feature's weight and the sum of its weighted absolute correlations with other features in the feature set. The problem is solved as an eigenvector problem with a unique solution (see Rahmatallah et. al. 2014 for details). The test statistic w_{GSNCA} is given by the first norm between the scaled weight vectors $w^{(1)}$ and $w^{(2)}$ (each vector is multiplied by its norm) between two conditions

$$w_{GSNCA} = \sum_{i=1}^{p} |w_i^{(1)} - w_i^{(2)}|$$

This test statistic tests the null hypothesis that $w_{GSNCA} = 0$ against the alternative that w_{GSNCA} does not equal to zero. The performance of this test was thoroughly examind in Rahmatallah et. al. (2014). The null distribution of the test statistic is estimated by permuting sample labels `nperm` times and calculating the test statistic for each. P-value is calculated as `p.value`=(`b` + 1)/(`nperm` + 1) where `b` is the number of permutations giving a more extreme statistic than the observed test statistic.

In the case of RNA-seq count data, some non-expressed genes may have zero counts across the samples under one or two conditions. Such situation results in zero or tiny standard deviation for one or more features. Such case produces an error in command `cor` used to compute the correlation coefficients between features. To avoid this situation, standard deviations are checked in advance when `check.sd` is `TRUE` (default) and if any is found below the minimum limit `min.sd` (default is `1e-3`), the execution stops and an error message is returned indicating the number of feature causing the problem (if only one the index of that feature is given too). If a feature has nearly a constant level for some samples under both conditions, permuting sample labels may group such samples under one condition and produce a standard deviation smaller than `min.sd`. To allow the test to skip such permutations without causing excessive delay, we set an upper limit for the number of allowed skips by the argument `max.skip` (default is 10). If the upper limit is exceeded, an error message is returned. Allowing this skipping may or may not solve the issue depending on the proportion of samples causing the problem in the feature set.

If the user is certain that the tested feature sets contain no feature with nearly equal levels over many samples (such as the case with microarrays), the checking stage for tiny standard deviations can be skipped by setting `check.sd` to `FALSE` in order to reduce the execution time.

Value

GSNCAtest produces a list with the following components:

`statistic` the value of the observed test statistic.

`perm.stat` numeric vector of the resulting test statistic for `nperm` random permutations of sample labels.

`p.value` p-value indicating the attained significance level.

Author(s)

Yasir Rahmatallah and Galina Glazko

References

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2014) Gene sets net correlations analysis (GSNCA): a multivariate differential coexpression test for gene sets. Bioinformatics **30**, 360–368.

See Also

`findMST2`, `plotMST2.pathway`.

Examples

```
## generate a feature set of length 20 in two conditions
## each condition has 20 samples
## use multivariate normal distribution with different covariance matrices
library(MASS)
ngenes <- 20
nsamples <- 40
zero_vector <- array(0,c(1,ngenes))
## create a covariance matrix with low off-diagonal elements
cov_mtrx1 <- diag(ngenes)
cov_mtrx1[!diag(ngenes)] <- 0.1
## create a covariance matrix with high off-diagonal elements
## for the first 5 features and low for the rest 15 features
cov_mtrx2 <- diag(ngenes)
cov_mtrx2[!diag(ngenes)] <- 0.1
mask <- diag(ngenes/4)
mask[!diag(ngenes/4)] <- 0.6
cov_mtrx2[1:(ngenes/4),1:(ngenes/4)] <- mask
gp1 <- mvrnorm((nsamples/2), zero_vector, cov_mtrx1)
gp2 <- mvrnorm((nsamples/2), zero_vector, cov_mtrx2)
gp <- rbind(gp1,gp2)
dataset <- aperm(gp, c(2,1))
## first 20 samples belong to group 1
## second 20 samples belong to group 2
result <- GSNCAtest(object=dataset, group=c(rep(1,20),rep(2,20)))
pvalue <- result$p.value
```

HDP.ranking *High Directed Preorder Ranking of MST*

Description

Rank nodes in an object of class igraph (see package igraph for the definition of class igraph) containing a minimum spanning tree (MST) according to the High Directed Preorder traversal of the tree.

Usage

```
HDP.ranking(mst)
```

Arguments

mst	object of class igraph containing the minimum spanning tree.

Details

Rank nodes in an object of class igraph (see package igraph) containing a minimum spanning tree (MST). The MST is rooted at a node with the largest geodesic distance and the rest of the nodes are ranked according to the high directed preorder (HDP) traversal of the tree (Friedman and Rafsky 1979).

Value

Numeric vector giving the node ranks according to HDP traversal of the MST.

Note

This function does not work properly if there is any node in the MST with more than 26 links. However, this situation is almost impossible for a dataset composed of a few hundreds or less of samples.

Author(s)

Yasir Rahmatallah and Galina Glazko

References

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2012) Gene set analysis for self-contained tests: complex null and specific alternative hypotheses. Bioinformatics **28**, 3073–3080.

Friedman J. and Rafsky L. (1979) Multivariate generalization of the Wald-Wolfowitz and Smirnov two-sample tests. Ann. Stat. **7**, 697–717.

See Also

KStest.

Examples

```
## generate random data using normal distribution
## generate 20 features in 20 samples
object <- matrix(rnorm(400),20,20)
objt <- aperm(object, c(2,1))
## calculate the weight matrix
Wmat <- as.matrix(dist(objt, method = "euclidean", diag = TRUE, upper = TRUE, p = 2))
## create a weighted undirectional graph from the weight matrix
gr <- graph.adjacency(Wmat, weighted = TRUE, mode = "undirected")
## find the minimum spanning tree
mst <- minimum.spanning.tree(gr)
ranks <- HDP.ranking(mst)
plot(mst)
```

KStest *Multivariate Kolmogorov-Smirnov Test of Means*

Description

Performs two-sample nonparametric multivariate test of means based on the minimum spanning tree (MST) and Kolmogorov-Smirnov statistic. It tests the null hypothesis that a set of features has the same mean in two conditions versus different means.

Usage

```
KStest(object, group, nperm=1000)
```

Arguments

`object`	a numeric matrix with columns and rows respectively corresponding to samples and features.
`group`	a numeric vector indicating group associations for samples. Possible values are 1 and 2.
`nperm`	number of permutations used to estimate the null distribution of the test statistic. If not given, a default value 1000 is used.

Details

This function tests the null hypothesis that a set of features has no shift between two conditions. It performs a two-sample nonparametric multivariate test based on the minimum spanning tree (MST) and Kolmogorov-Smirnov statistic as proposed by Friedman and Rafsky (1979). The MST of the weighted undirectional graph created from the samples is found. The nodes of the MST are ranked based on their position in the MST. The MST is rooted at the node with largest geodisic distance (rank 1) and then nodes are ranked in the High Directed Preorder (HDP) traversal of the tree (Rahmatallah et. al. 2012). The quantity $d_i = (r_i/n_1) - (s_i/n_2)$ is calculated where $r_i(s_i)$ is the number of nodes (samples) from condition 1(2) which ranked lower than i, $1 \leq i \leq N$ and N is the total number of samples. The Kolmogorov-Smirnov statistic is given by the maximum absolute difference $D = max|d_i|$. The performance of this test under different alternative hypotheses was thoroughly examind in Rahmatallah et. al. (2012). The null distribution of the test statistic is estimated by permuting sample labels `nperm` times and calculating the test statistic for each. P-value is calculated as `p.value`=(`b` + 1)/(`nperm` + 1) where `b` is the number of permutations giving a more extreme statistic than the observed test statistic.

Value

KStest produces a list with the following components:

`statistic`	the value of the observed test statistic.
`perm.stat`	numeric vector of the resulting test statistic for `nperm` random permutations of sample labels.
`p.value`	p-value indicating the attained significance level.

Note

This function invokes function `HDP.ranking` which does not work properly if there is any node in the MST with more than 26 links. However, this situation is almost impossible for a dataset composed of a few hundreds or less of samples.

Author(s)

Yasir Rahmatallah and Galina Glazko

References

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2012) Gene set analysis for self-contained tests: complex null and specific alternative hypotheses. Bioinformatics **28**, 3073–3080.

Friedman J. and Rafsky L. (1979) Multivariate generalization of the Wald-Wolfowitz and Smirnov two-sample tests. Ann. Stat. **7**, 697–717.

See Also

RKStest, WWtest, HDP.ranking.

Examples

```
## generate a feature set of length 20 in two conditions
## each condition has 20 samples
## use multivariate normal distribution
library(MASS)
ngenes <- 20
nsamples <- 40
## let the mean vector have zeros of length 20 both conditions
zero_vector <- array(0,c(1,ngenes))
## set the covariance matrix to be an identity matrix for both conditions
cov_mtrx <- diag(ngenes)
gp <- mvrnorm(nsamples, zero_vector, cov_mtrx)
## apply a mean shift of 3 to half of the features under condition 2
gp[1:20,1:10] <- gp[1:20,1:10] + 3
dataset <- aperm(gp, c(2,1))
## first 20 samples belong to group 1
## second 20 samples belong to group 2
result <- KStest(object=dataset, group=c(rep(1,20),rep(2,20)))
pvalue <- result$p.value
```

p53DataSet *p53 Dataset of the NCI-60 Cell Lines*

Description

A matrix of gene expression profiles for a processed version of the p53 dataset obtained from the NCI-60 cell lines using the hgu95av2 microarray platform.

Usage

```
data(p53DataSet)
```

Format

A matrix of 8655 rows and 50 columns where rows correspond to genes and columns correspond to samples. Gene symbol identifiers are used for rows. Column names indicate the class of the samples (wild type p53 or mutated p53) with the first 17 column names starting with `WT1` and ending with `WT17` and next 33 column names starting with `MUT1` and ending with `MUT33`.

Details

p53 is a major tumor suppressor protein. The p53 dataset comprises 50 samples of NCI-60 cell lines differentiated based on the status of the TP53 gene: 17 cell lines carrying wild type (WT) TP53 and 33 cell lines carrying mutated (MUT) TP53 (Olivier et. al. 2002, Subramanian et. al. 2005). Transcriptional profiles obtained from microarrays of platform hgu95av2 were obtained from the available datasets at the GSEA Broad Institute's website.

Probe level intensities were quantile normalized and transformed to the log scale using log2(1 + intensity). Probes originally had Affymetrix identifiers which are mapped to unique gene symbol identifiers. Probes without mapping to entrez and gene symbol identifiers were discarded. Probes with duplicate intensities were assessed and the probe with the largest absolute value of t-statistic between WT and MUT conditions was selected as the gene match. Genes were assigned gene symbol identifiers and columns were assigned names indicating weither they belong to WT or MUT condition. The columns were sorted such that the first 17 columns are WT samples and the next 33 columns are the MUT samples. `p53DataSet` was used in the analysis presented in Rahmatallah et. al. 2014.

Source

Broad Institute (http://www.broadinstitute.org/gsea/datasets.jsp)

References

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2014) Gene sets net correlations analysis (GSNCA): a multivariate differential coexpression test for gene sets. Bioinformatics **30**, 360–368.

Subramanian A., Tamayo P., Mootha V., Mukherjee S., Ebert B., Gillette M., Paulovich A., Pomeroy S., Golub T., Lander E. and Mesirov J. (2005) Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. **102**, 15545–15550.

Olivier M., Eeles R., Hollstein M., Khan M., Harris C. and Hainaut P. (2002) The IARC TP53 database: new online mutation analysis and recommendations to users. Hum. Mutat. **19**, 607–614.

Examples

```
data(p53DataSet)
dim(p53DataSet)
```

`plotMST2.pathway` *Plot MST2 for a pathway in two conditions*

Description

This is a wrapper function which uses function `findMST2` to find the union of the first and second minimum spanning trees (or MST2) of the correlation network for a feature set (pathway) under two conditions. It plots the MST2 of the correlation network of the feature set under both conditions side-by-side and highlights hub nodes to facilitate a visual comparison.

Usage

```
plotMST2.pathway(object, group, name=NULL, legend.size=1, label.size=1,
cor.method="pearson", min.sd=1e-3, return.weights=FALSE)
```

Arguments

`object` a numeric matrix with columns and rows respectively corresponding to samples and features. Gene names are provided to this function as the rownames of this matrix.

`group` a numeric vector indicating group associations for samples. Possible values are 1 and 2.

`name` an optional character string giving the name of the feature set (pathway). If given, the name will be displayed at the top of the plot.

`legend.size` an optional numeric value controlling the relative font size of the legend to the default font size. Default is 1.

`label.size` an optional numeric value controlling the relative font size of vertex labels to the default font size. Default is 1.

`cor.method` a character string indicating which correlation coefficient is to be computed. Possible values are “`pearson`” (default), “`spearman`” and “`kendall`”.

`min.sd` the minimum allowed standard deviation for any feature. If any feature has a standard deviation smaller than `min.sd` the execution stops and an error message is returned.

`return.weights` logical. Default value is FALSE. If the weight factors aasigned to the genes by the GSNCA method are desired, setting this parameter to TRUE returns the weight factors in a matrix with 2 columns (for class 1 and class 2) and number of rows equal to the number of genes in the gene set. If the `rownames` of `object` are provided, then they will be used as `rownames` for the returned matrix. If the `rownames` of `object` are abscent, node labels will be set to `as.character(c(1:nrow(object)))`.

Details

This is a wrapper plotting function for the convenience of users. It uses function `findMST2` to find the union of the first and second minimum spanning trees (or MST2) of the correlation network for a feature set (pathway) under two conditions and plots them side-by-side. It also lists the hub nodes and their weight factors (w) under each condition (see Rahmatallah et. al. 2014 for details). The range in which weight factors fall is indicated by the node colors defined in the legend. Weight factor have values mostly ranging between 0.5 (low coexpression) and 1.5 (high coexpression). To allow the users more control over plotting parameters and to present different feature sets appropriately, two optional arguments were introduced: `legend.size` and `label.size`. Node lables will be the names of the features in the set, i.e. `rownames(object)`. If the `rownames` attribute is not set for `object`, node labels will be set to `as.character(c(1:nrow(object)))`.

The weight factors, inferred from the Gene Sets Net Correlations Analysis (GSNCA) method (see `GSNCAtest`), correlate to some extent with genes centralities in the MST2: genes with large weights are placed near the center of the MST2, and genes with small weights are placed on the periphery (Rahmatallah et. al. 2014). Adopting network terminology, a gene with the largest weight is a hub gene, coexpressed with most of the other genes in a pathway (see findMST2). Therefore, MST2 is a convenient graphical visualization tool to examine the pathways tested by the GSNCA method (see GSNCAtest).

The correlation (coexpression) network is obtained using the weight matrix W with elements $w_{ij} = 1 - |r_{ij}|$ where r_{ij} is the correlation between features i and j and w_{ij} is the weight of the link between vertices (nodes) i and j in the network. The correlation coefficient used is indicated by the argument `cor.method` with three possible values: “`pearson`” (default), “`spearman`” and “`kendall`”.

In some cases (especially for RNA-Seq count data), a feature (or more) may have a constant or nearly constant level across the samples in one or both conditions. This results in a zero or a tiny standard deviation. Such case produces an error in command `cor` used to compute the correlation coefficients between features. To avoid this situation, standard deviations are checked in advance and if any is found below the minimum limit `min.sd` (default is `1e-3`), the execution stops and an error message is returned indicating the number of feature causing the problem (if only one the index of that feature is given too).

Note

This function is suitable for a feature set of roughly 80 features or less. It works for feature sets with larger number of features but the placements of nodes and their labels in the plot will be too crowded for a useful visual presentation.

Author(s)

Yasir Rahmatallah and Galina Glazko

References

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2014) Gene sets net correlations analysis (GSNCA): a multivariate differential coexpression test for gene sets. Bioinformatics **30**, 360–368.

See Also

findMST2, GSNCAtest.

Examples

```
## generate a feature set of length 20 in two conditions
## each condition has 20 samples
## use multivariate normal distribution with different covariance matrices
library(MASS)
ngenes <- 20
nsamples <- 40
zero_vector <- array(0,c(1,ngenes))
## create a covariance matrix with low off-diagonal elements
cov_mtrx1 <- diag(ngenes)
cov_mtrx1[!diag(ngenes)] <- 0.1
## create a covariance matrix with high off-diagonal elements
## for the first 5 features and low for the rest 15 features
cov_mtrx2 <- diag(ngenes)
```

```
cov_mtrx2[!diag(ngenes)] <- 0.1
mask <- diag(ngenes/4)
mask[!diag(ngenes/4)] <- 0.6
cov_mtrx2[1:(ngenes/4),1:(ngenes/4)] <- mask
gp1 <- mvrnorm((nsamples/2), zero_vector, cov_mtrx1)
gp2 <- mvrnorm((nsamples/2), zero_vector, cov_mtrx2)
gp <- rbind(gp1,gp2)
dataset <- aperm(gp, c(2,1))
## first 20 samples belong to group 1
## second 20 samples belong to group 2
## since rowname(object)=NULL, node labels will be automatically
## set to as.character(c(1:nrow(object)))
plotMST2.pathway(object=dataset, group=c(rep(1,20),rep(2,20)),
name="Example Pathway")
```

RKStest *Multivariate Radial Kolmogorov-Smirnov Test of Variance*

Description

Performs two-sample nonparametric multivariate test of variance based on the minimum spanning tree (MST) and Kolmogorov-Smirnov statistic. It tests the null hypothesis that a set of features has the same scale in two conditions versus different scales.

Usage

```
RKStest(object, group, nperm=1000)
```

Arguments

`object`	a numeric matrix with columns and rows respectively corresponding to samples and features.
`group`	a numeric vector indicating group associations for samples. Possible values are 1 and 2.
`nperm`	number of permutations used to estimate the null distribution of the test statistic. If not given, a default value 1000 is used.

Details

This function tests the null hypothesis that a set of features has the same scale in two conditions. It performs a two-sample nonparametric multivariate test based on the minimum spanning tree (MST) and Kolmogorov-Smirnov statistic as proposed by Friedman and Rafsky (1979). The MST of the weighted undirectional graph created from the samples is found. The nodes of the MST are ranked based on their position in the MST. The MST is rooted at the node with smallest geodisic distance (rank 1) and nodes with higher depths from the root are assigned higher ranks. The quantity $d_i = (r_i/n_1) - (s_i/n_2)$ is calculated where $r_i(s_i)$ is the number of nodes (samples) from condition 1(2) which ranked lower than i, $1 \leq i \leq N$ and N is the total number of samples. The Kolmogorov-Smirnov statistic is given by the maximum absolute difference $D = max|d_i|$. The performance of this test under different alternative hypotheses was thoroughly examind in Rahmatallah et. al. (2012). The null distribution of the test statistic is estimated by permuting sample labels nperm times and calculating the test statistic for each. P-value is calculated as p.value=(b + 1)/(nperm + 1) where b is the number of permutations giving a more extreme statistic than the observed test statistic.

Value

RKStest produces a list with the following components:

statistic	the value of the observed test statistic.
perm.stat	numeric vector of the resulting test statistic for nperm random permutations of sample labels.
p.value	p-value indicating the attained significance level.

Note

The variance of both the Poisson and negative Bionomial distributions, used to model count data, is a function of their mean. Therefore, using the radial Kolmogorov-Smirnov test (RKStest) to detect pathways with differential variance for RNA-Seq counts is not possible. RKS test still works well for microarray data.

Author(s)

Yasir Rahmatallah and Galina Glazko

References

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2012) Gene set analysis for self-contained tests: complex null and specific alternative hypotheses. Bioinformatics **28**, 3073–3080.

Friedman J. and Rafsky L. (1979) Multivariate generalization of the Wald-Wolfowitz and Smirnov two-sample tests. Ann. Stat. **7**, 697–717.

See Also

KStest, WWtest.

Examples

```
## generate a feature set of length 20 in two conditions
## each condition has 20 samples
## use multivariate normal distribution
library(MASS)
ngenes <- 20
nsamples <- 40
## let the mean vector have zeros of length 20 for both conditions
zero_vector <- array(0,c(1,ngenes))
## set the covariance matrix to be an identity matrix for condition 1
cov_mtrx <- diag(ngenes)
gp1 <- mvrnorm((nsamples/2), zero_vector, cov_mtrx)
```

```
## set some scale difference in the covariance matrix for condition 2
cov_mtrx <- cov_mtrx*3
gp2 <- mvrnorm((nsamples/2), zero_vector, cov_mtrx)
## combine the data of two conditions into one dataset
gp <- rbind(gp1,gp2)
dataset <- aperm(gp, c(2,1))
## first 20 samples belong to group 1
## second 20 samples belong to group 2
result <- RKStest(object=dataset, group=c(rep(1,20),rep(2,20)))
pvalue <- result$p.value
```

WWtest *Multivariate Generalization of the Wald-Wolfowitz Runs Test*

Description

Performs two-sample nonparametric multivariate generalization of the Wald-Wolfowitz runs test based on the minimum spanning tree (MST). It tests the alternative hypothesis that a set of features has different distributions in two conditions against the null hypothesis of having the same distribution.

Usage

```
WWtest(object, group, nperm=1000)
```

Arguments

`object`	a numeric matrix with columns and rows respectively corresponding to samples and features.
`group`	a numeric vector indicating group associations for samples. Possible values are 1 and 2.
`nperm`	number of permutations used to estimate the null distribution of the test statistic. If not given, a default value 1000 is used.

Details

This function tests the alternative hypothesis that a set of features has different distributions in two conditions against the null hypothesis of having the same distribution. It performs the two-sample nonparametric multivariate generalization of the Wald-Wolfowitz runs test based on the minimum spanning tree (MST) as proposed by Friedman and Rafsky (1979). The performance of this test under different alternative hypotheses was thoroughly examind in Rahmatallah et. al. (2012). The null distribution of the test statistic is estimated by permuting sample labels `nperm` times and calculating the test statistic for each. P-value is calculated as `p.value=(b + 1)/(nperm + 1)` where `b` is the number of permutations giving a more extreme statistic than the observed test statistic.

Value

WWtest produces a list with the following components:

`statistic` the value of the observed test statistic.

`perm.stat` numeric vector of the resulting test statistic for `nperm` random permutations of sample labels.

`p.value` p-value indicating the attained significance level.

Author(s)

Yasir Rahmatallah and Galina Glazko

References

Rahmatallah Y., Emmert-Streib F. and Glazko G. (2012) Gene set analysis for self-contained tests: complex null and specific alternative hypotheses. Bioinformatics **28**, 3073–3080.

Friedman J. and Rafsky L. (1979) Multivariate generalization of the Wald-Wolfowitz and Smirnov two-sample tests. Ann. Stat. **7**, 697–717.

See Also

`KStest`, `RKStest`.

Examples

```
## generate a feature set of length 20 in two conditions
## each condition has 20 samples
## use multivariate normal distribution
library(MASS)
ngenes <- 20
nsamples <- 40
## let the mean vector have zeros of length 20 for condition 1
zero_vector <- array(0,c(1,ngenes))
## let the mean vector have 2s of length 20 for condition 2
mu_vector <- array(2,c(1,ngenes))
## set the covariance matrix to be an identity matrix
cov_mtrx <- diag(ngenes)
gp1 <- mvrnorm((nsamples/2), zero_vector, cov_mtrx)
gp2 <- mvrnorm((nsamples/2), mu_vector, cov_mtrx)
## combine the data of two conditions into one dataset
gp <- rbind(gp1,gp2)
dataset <- aperm(gp, c(2,1))
## first 20 samples belong to group 1
## second 20 samples belong to group 2
result <- WWtest(object=dataset, group=c(rep(1,20),rep(2,20)))
pvalue <- result$p.value
```

Index