GeneExpressionSignature

March 24, 2012

`BMRankMerging` *Merging two or more selected ranked lists into a new one ranked list*

Description

Implements a majority voting system, use the Borda Merging Method to merging two or more ranked lists into single one list. This method is used in function krubor.

Usage

```
BMRankMerging(rankings)
```

Arguments

`rankings`	a matrix or a data.frame, which must be numeric.

References

Lin S. (2010) *Space oriented rank-based data integration*

`FootruleMatrix` *Create a footrule-matrix*

Description

Compute distances between any two ranked lists with the same length, and create a n-n matrix, where n is the length of the ranked lists.

Usage

```
FootruleMatrix(Rankings, n)
```

Arguments

`Rankings`	a numeric matrix or a data.frame to be computed
`n`	a number, a non zero value n means a normalized results matrix should be returned.

Details

This function uses SMfootrule to compute any two ranked lists in the first argument, a column represents a ranked list in the first argument. n is used to indicate whether the distance matrix is normalized.

Value

This function returns a n-n numberic matrix as a distance matrix, where n is the number of column of the first argument. And the m-n elements in the result should be equal to the n-m elements in the result.

See Also

SMfootrule, findclosestrank

GeneExpressionSignature *Introduction to the GeneExpressionSignature Package*

Description

The **GeneExpressionSignature** add-on is an implementation of computing distances among preprocessed gene-expression profiles of samples for R. The distances can be used to detect similarities among the signatures of drugs, diseases, and biological states of interest, and construct connectivity map.

Details

This package contains functions for the distances computation based on gene expression signature. First, list of genes is ranked according to their expression ratios to produce the Prototype Ranked List (PRL). Second, all the PRLs with the same state are aggregated by aggregate function. Finally, all the ranked lists are made as one input of the distances function to compute the pairwise distances.

SMfootrule *Compute distance between two ranked lists.*

Description

Compute the distances between the two input ranked lists.

Usage

```
SMfootrule(R1, R2)
```

Arguments

R1	a ranked lists represented as a array, vector, data.frame, or matrix with one single column.
R2	with the same type of R1.

Details

Return a nonnegative number that represents the distances between the two input ranked lists using Spearman's algorithm. The two input ranked lists must be of the same length, which is used to measure the similarity of two ranked lists. In this package, this function can be used to compute distances between ranked lists which obtained for each gene expression profile by sorting the microarray probe-set identifiers according to the expression ratios (in decreasing order) with respect to the untreated hybridization.

References

Diaconis, R.L Graham. (1977) *Spearman's footrule as a matter of disarray*

See Also

`FootruleMatrix`

`aggregate` *Aggregate each group of ranked lists with the same state into a single list*

Description

Aggregate the assay data according to phenotypic data of the input ExpressionSet. Each group of the ranked lists with the same phenotypic data is aggregated into a single list, return it as an ExpressionSet object.

Usage

```
aggregate(exprSet)
```

Arguments

`exprSet`	an ExpressionSet object, each column of assay data represents a ranked list obtained by preprocessing the corresponding gene expression profile, and phenotypic data represents the short description (characteristics of gene expression profile, such as the drug type, the disease state) about the assay data.

Details

The krubor function is used in the aggregating procedure. And the following methods are used in the implementation: a measure of the distance between two ranked lists (Spearman's Footrule), a method to merge two or more ranked lists the (Borda Merging Method), and a algorithm to obtain a single ranked list from a set of them in a hierarchical way (the Kruskal Algorithm).

See Also

`krubor,aggregate all ranked lists into one list`

Examples

```
library(Biobase)
## load sample ranked list
PRLs=as.matrix(read.table(system.file("extdata/example_PRLs.txt",package="GeneExpressionS
## load sample phenotypic data
states=read.table(system.file("extdata/example_states.txt",package="GeneExpressionSignatu
## create an new ExpressionSet object
rownames(states)=colnames(PRLs)
phenodata=new("AnnotatedDataFrame",data = states)
exprSet=new("ExpressionSet",exprs=PRLs,phenoData=phenodata)
## aggregate each group of the ranked lists in the exprSet with the same phenotypic data
aggregateSet=aggregate(exprSet)
```

`distances` *Compute pairwise distances between samples*

Description

Compute pairwise distances between sample according to their (Prototype Ranked List) PRL, get a n-n distance matrix as the assay data of the result , n is the length of PRL.

Usage

```
distances(aggregateSet, qlen)
```

Arguments

`aggregateSet` an ExpressionSet object. The assay data represents the PRLs of the samples, each column represents one PRL. The number of column of this argument must be greater than 1, oherwise, this function is not meaningful.

`qlen` the length of "gene signature". In order to compute pairwise distances among samples, genes lists are ranked according to the gene expression ratio (fold change). And the "gene signature" includes the most up-regulated genes (near the top of the list) and the most down-regulated genes (near the bottom of the list).

Details

Once the PRL obtained for each sample, the distances between samples are calculated base on gene signature, including the expression of genes that seemed to consistently vary in response to the across different experimental conditions (e.g., different cell lines and different dosages).

Value

`ES` an ExpressionSet, assay data is the enrichment score matrix

`DS` an ExpressionSet, assay data is the distance matrix, the maximum distance is more sensitive to weak similarities, providing a lower precision but a larger recall.

See Also

`aggregate`

Examples

```
## create an instance ExpressionSet
library(Biobase)
PRLs=as.matrix(read.table(system.file("extdata/example_PRLs.txt",package="GeneExpressionS
states=read.table(system.file("extdata/example_states.txt",package="GeneExpressionSignatu
rownames(states)=colnames(PRLs)
phenodata=new("AnnotatedDataFrame",data = states)
exampleSet=new("ExpressionSet",exprs=PRLs,phenoData=phenodata)

## aggregate the exampleSet
PRL=aggregate(exampleSet)

## compute distances from aggregated matrix
d=distances(PRL,250)
enrichmentscore=d[[1]]
distance=d[[2]]
```

`findclosestrank` *Find the closest ranks.*

Description

Find the two closest ranks among ranks with the same state. Get the NO. of the two closest ranks.

Usage

```
findclosestrank(SMDM)
```

Arguments

`SMDM` a distance matrix or a data.frame represents the distances between any two ranked lists, which must be preprocessed before used (let the lower triangular part of the matrix is Inf).

Details

Get the distance matrix by using `FootruleMatrix` function. This function is used to find the two closest ranked lists to aggregate them into a new list.

See Also

`krubor`, `FootruleMatrix`

`integratePRL` *Updating an existing dataset with new sample.*

Description

Compute new enrichment score and distances among new PRL and previous PRLs in the existing dataset.

Usage

```
integratePRL(ES, PRL, newPRL, qlen)
```

Arguments

`ES`	an ExpressionSet, array data is the existing enrichment score matrix
`PRL`	the existing PRLs correspond to the ES
`newPRL`	the PRL which you want to integrate to the existing PRLs
`qlen`	the length of the gene signature

Details

This function can integrate the new PRL into the previous PRLs to get the new enrichment score and distances matrix.

Value

`newPRLs`	an ExpressionSet, assay data is the PRL which new PRL have been integrated
`newES`	an ExpressionSet, assay data is the integrated new ES matrix
`newdistance`	an ExpressionSet, assay data is the integrated new distance matrix

See Also

`quickenrichmentscore`

Examples

```
## create an instance ExpressionSet
library(Biobase)
PRLs=as.matrix(read.table(system.file("extdata/example_PRLs.txt",package="GeneExpressionS
states=read.table(system.file("extdata/example_states.txt",package="GeneExpressionSignatu
rownames(states)=colnames(PRLs)
phenodata=new("AnnotatedDataFrame",data = states)
exampleSet=new("ExpressionSet",exprs=PRLs,phenoData=phenodata)

## aggregate the exampleSet
PRL=aggregate(exampleSet)

## compute distances and ES from aggregated matrix
d=distances(PRL,250)
ES=d[[1]]
distance=d[[2]]
```

```
## integrate new PRL to get newES and newdistances
newPRL<- PRL[,2]
d <- integratePRL(ES,PRL,newPRL,250)
newES <- d[[2]]
newdistance <- d[[3]]
```

krubor *Aggregate all ranked lists into one list*

Description

Return a matrix with one column representing all the input ranked lists, get a single Prototype Ranked List(PRL)

Usage

```
krubor(...)
```

Arguments

`...`	column vectors,matrices or data.frames. These can be given as named arguments. The mode of arguments must be numeric.

Details

This function is aim to aggregate all ranked lists with the same state into one single ranked list. First, remove the duplicate columns. If there are the same columns in combination, delete the same columns until only one of them left. Second, aggregate the lists with the same state using the Borda Merging Method until only one single list left.

The arguments can be a mix of matrices, vectors and data.frames. The length of the column of the matrices or data.frames and the length of the vectors must be equal.

Value

A matrix with one column as the aggregated list.

See Also

`aggregate`which uses krubor to aggregate ranked lists according to the biological states.

Examples

```
## the inputs are in the same class
krubor(matrix(2,3,3),matrix(3,3,3))

## the inputs are mixed
krubor(matrix(2,3,3),as.data.frame(matrix(3,3,3)))
```

quickenrichmentscore

Compute the Enrichment Score.

Description

Use Gene Set Enrichmentscore Analysis (GSEA) method to compute the Enrichment Score among PRLs .

Usage

```
quickenrichmentscore(S, S1, List)
```

Arguments

`S`	optimal signature,the up-regulated genes of a PRL A
`S1`	optimal signature, the down-regulated genes of a PRL A
`List`	a PRL which distances from A (whose optimal signature is S and S1) will be computed.

Details

Once gene signature is given, we computed the GSEA enrichment score between ranked list A and B based on the gene signature of A, and vice versa. The average value of this two enrichment scores is used to quantify the distance between A and B.

References

Subramanian. (2005) *Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles.*

See Also

`distances`, `integratePRL`

Index