Package 'BootMRMR'

September 12, 2016

Type Package

Title Bootstrap-MRMR Technique for Informative Gene Selection

Version 0.1

Date 2016-08-30

Author Samarendra Das <samarendra4849@gmail.com>

Maintainer Samarendra Das <samarendra4849@gmail.com>

Depends R (>= 3.3.1)

Description Selection of informative features like genes, transcripts, RNA seq, etc. using Bootstrap Maximum Relevance and Minimum Redundancy technique from a given high dimensional genomic dataset. Informative gene selection involves identification of relevant genes and removal of redundant genes as much as possible from a large gene space. Main applications in high-dimensional expression data analysis (e.g. microarray data, NGS expression data and other genomics and proteomics applications).

LazyLoad Yes

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-12 15:08:51

R topics documented:

`bmrmr.pval.cutoff` *Selection of informative geneset based on statistical significance value using Bootstrap-MRMR technique*

Description

The informative geneset which has maximum relevance with target class/trait and minimum redundancy among genes based on statistical significance values computed from the Bootstrap-MRMR technique.

Usage

```
bmrmr.pval.cutoff(x, y, s, Q, n)
```

Arguments

`x`	x is a N by p data frame of gene expression values where rows represent genes and columns represent samples or subjects or time point. Each cell entry represents the expression level of a gene in a sample/subject (row names of x as gene names or gene ids).
`y`	y is a p by 1 numeric vector with entries 1 or -1 representing sample labels, where, 1/-1 represents the sample label of subjects/ samples for stress/control condition(for two class problems).
`s`	s is a scalar representing the number of bootstrap generated, s must be sufficiently large (i.e. number of times bootstrap samples are generated.
`Q`	Q is a scalar representing the quartile value of the rankscores of genes (lies within 1/N to 1), usually the second quartile, i.e. 0.5 or third quartile i.e. 0.75 may be taken.
`n`	n is a scalar representing the size of the informative geneset to be obtained.

Value

The function returns a list of the genes/informative geneset which are highly relevant to the particular trait/condition under investigation and minimal redundant among themselves.

Author(s)

Samarendra Das

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
s=80
Q=0.5
n=20
bmrmr.pval.cutoff(x, y, s, Q, n)
```

---

bmrmr.weight.cutoff *Selection of informative geneset using gene weights obtained from the Bootstrap-MRMR technique*

Description

The function enables to find set of informative genes which are obtained based on weights computed from Bootstrap-MRMR technique.

Usage

```
bmrmr.weight.cutoff(x, y, s, n)
```

Arguments

x	x is a N by p dataframe of gene expression, where rows are genes and columns are as samples or subjects (gene names are taken as row names). Each cell or entry represents the expression level of a gene for a sample or subject.
y	y is a p by 1 numeric vector having elements as 1 and -1 representing the sample labels of samples or subjects (for two class problems, i.e. stress or control respectively).
s	s is a numeric constant representing the number of bootstrap samples drawn (s must be sufficiently large)
n	n must be a numeric constant representing the number of informative genes to be selected from the large gene space.

Value

The function returns a set of genes, which are highly informative to the trait or condition under consideration based on the computed weights form Bootstrap-MRMR technique.

Author(s)

Samarendra Das

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
s=80
n=20
bmrmr.weight.cutoff(x, y, s, n)
```

`bootmr.weight` *Computation of weights for informative genes/ geneset selection using Bootstrap-MRMR technique*

Description

The function computes the weights associated with each genes for a given dataset using Bootstrap-MRMR technique.

Usage

```
bootmr.weight(x, y, s, plot)
```

Arguments

`x`	x is a N by p dataframe of gene expression, where rows are genes and columns are as samples/subjects (gene names are taken as row names). Each cell/entry represents the expression level of a gene in a sample/subject.
`y`	y is a p by 1 numeric vector having elements as 1/-1 representing the sample labels of samples/subjects (for two class problems, i.e. stress/control)
`s`	s is a numeric constant representing the number of bootstrap samples drawn (s must be sufficiently large)
`plot`	plot is a character string must either take logical value TRUE/FALSE representing whether the plot of the gene weights of all genes in the dataset needs to be constructed or not.

Details

The function returns a vector of weights associated with each genes computed from Bootstrap-MRMR technique for a given dataset.

Author(s)

Samarendra Das

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
s=80
bootmr.weight(x, y, s, plot=FALSE)
```

`geneslect.f` *Informative gene set selection using F-score*

Description

The function returns geneset which is informative for a particular trait/condition using F-score as the gene selection criterion.

Usage

```
geneslect.f(x, y, s)
```

Arguments

x	x is a N by p dataframe of gene expression, where, rows represent as genes and columns as samples/subjects (with row names as gene names/ids).
y	y is a p by 1 numeric vector of 1 and -1, where 1/-1 indicates the class label of the samples/subjects either of two classes (e.g. stress and control).
s	s is a numeric constant (< N) representing the number of genes to be selected from the large gene space.

Details

This function identifies the genes/ geneset which is informative for the particular trait/condition using F-score as a criterion.

Author(s)

Samarendra Das

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
s=20
geneslect.f (x, y, s)
```

`mbmr.pval.cutoff` *Selection of informative geneset based on statistical significance value using Modified Bootstrap MRMR technique*

Description

The informative geneset which has maximum relevance with target class/trait and minimum redundancy among genes are selected based on p-values obtained from Modified Bootstrap MRMR technique.

Usage

```
mbmr.pval.cutoff(x, y, m, s, Q, n)
```

Arguments

`x`	x is a N by p data frame of gene expression values where rows represent genes and columns represent samples/subject/time point. Each cell entry represents the expression level of a gene in a sample/subject (row names of x as gene names/gene ids).
`y`	y is a p by 1 numeric vector with entries 1/-1 representing sample labels, where 1/-1 represents the sample label of subjects/ samples for stress/control condition (for two class problems).
`m`	m is a scalar representing the size of the Modified Bootstrap Sample (i.e. Out of p samples/subjects, m samples/subjects are randomly drawn with replacement, which constitutes one Modified Bootstrap Sample).
`s`	s is a scalar representing the number of Modified Bootstrap samples (i.e. number of times each of the m samples/subjects will be resampled from p samples/subjects).
`Q`	Q is a scalar representing the quartile value of the gene rankscores (lies within 1/N to 1), usually the second quartile, i.e. 0.5 or third quartile i.e. 0.75
`n`	n is a scalar representing the size of the informative gene set to be obtained.

Value

The function returns a list of the genes/ geneset which are highly informative to the particular trait/condition under investigation using Modified Bootstrap MRMR technique.

Author(s)

Samarendra Das

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
m=36
s=80
Q=0.5
n=20
mbmr.pval.cutoff(x, y, m, s, Q, n)
```

`mbmr.weight.cutoff` *Identification of informative geneset based on weights obtained from Modified Bootstrap-MRMR technique*

Description

The function enables to find set of informative genes based on weights which are obtained by maximising the relevancy of genes with classes/condition/trait and minimising the redundancy among genes using Modified Bootstrap-MRMR technique

Usage

```
mbmr.weight.cutoff(x, y, m, s, n)
```

Arguments

`x`	x is a N by p data frame of gene expression values where rows represent genes and columns represent samples/subject/time point. Each cell entry represents the expression level of a gene in a sample/subject (row names of x as gene names/gene ids).
`y`	y is a p by 1 numeric vector with entries 1/-1 representing sample labels, where 1/-1 represents the sample label of subjects/ samples for stress/control condition (for two class problems).
`m`	m is a scalar representing the size of the Modified Bootstrap Sample (i.e. Out of p samples/subjects, m samples/subjects are randomly drawn with replacement, which constitutes one Modified Bootstrap Sample).
`s`	s is a scalar representing the number of Modified Bootstrap samples (i.e. number of times each of the m samples/subjects will be resampled from p samples/subjects).
`n`	n is a numeric constant representing the number of informative genes to be selected from the large gene space.

Value

The function returns a set of genes, which are highly informative to the trait or condition under consideration based Modified Bootstrap-MRMR weights.

Author(s)

Samarendra Das

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
m=36
s=80
n=20
mbmr.weight.cutoff(x, y, m, s, n)
```

`mrmr.cutoff` *Informative geneset selection using MRMR weights*

Description

The function returns the informative genes/ geneset for the particular trait/condition under investigation using Maximum Relevance and Minimum Redundancy (MRMR) technique.

Usage

```
mrmr.cutoff(x, y, n)
```

Arguments

x	x is a N by p data frame of gene expression values where rows represent genes and columns represent samples/subject/time point. Each cell entry represents the expression level of a gene in a sample/subject (row names of x as gene names/gene ids).
y	y is a p by 1 numeric vector with entries 1 and -1 representing sample labels, where 1 and -1 represents the sample label of subjects/ samples for stress and control condition respectively.
n	n is a numeric constant represents the number of informative genes to be selected.

Value

An informative geneset is obtained, which is relevanit to the particular trait/condition and the genes within the selected geneset are minimum redundant using MRMR technique.

Author(s)

Samarendra Das

References

Ding, C and Peng, H (2005). Minimum redundancy feature selection from microarray gene expression data. J. Bioinformatics Comput Biol 3(2):185-205.

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
n=20
mrmr.cutoff(x, y, n)
```

`pval.bmrmr` *Compuation of statistical significance values for genes using Bootstrap-MRMR technique*

Description

The function computes ths statisical significance values for the genes from the non-parametric test "H0: i-th gene is not informative against H1: i-th gene is informative" for selection of informative genes using Bootstrap-MRMR technique

Usage

```
pval.bmrmr(x, y, s, Q, plot)
```

Arguments

`x`	x is a N by p data frame of gene expression values where rows represent genes and columns represent samples/subject/time point. Each cell entry represents the expression level of a gene in a sample/subject (row names of x as gene names/gene ids).
`y`	y is a p by 1 numeric vector with entries 1 and -1 representing sample labels, where 1 and -1 represents the sample label of subjects/ samples for stress and control condition respectively.
`s`	s is a scalar representing the number of bootstraps generated, s must be sufficiently large (i.e. number of times bootstrap samples are generated
`Q`	Q is a scalar representing the quartile value of the gene rankscores (lies within 1/N to 1), usually the second quartile (Q2), i.e. 0.5 or third quartile (Q3) i.e. 0.75 is taken.
`plot`	plot is a character string must either take logical value TRUE/FALSE representing whether to plot the statistical significance values of genes in the dataset.

Value

The function returns a vector of p-values for all the genes from the given statistical test in the dataset using Bootstrap-MRMR technique.

Author(s)

Samarendra Das

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
s=80
Q=0.5
pval.bmrmr(x, y, s, Q, plot=FALSE)
```

`pval.mbmr` *Computation of statistical significance values for genes using Modified Bootstrap MRMR technique for a particular trait/condition*

Description

The statisical significance values (p-values) will be computed for all the genes in the dataset from the non-parametric test "H0: i-th gene is not informative against H1: i-th gene is informative" for selection of informative genes using Modified Bootstrap MRMR technique.

Usage

```
pval.mbmr(x, y, m, s, Q, plot)
```

Arguments

`x`	x is a N by p data frame of gene expression values where rows represent genes and columns represent samples/subject/time point. Each cell entry represents the expression level of a gene in a sample/subject (row names of x as gene names/gene ids).
`y`	y is a p by 1 numeric vector with entries 1 and -1 representing sample labels, where 1 and -1 represents the sample label of subjects/ samples for stress and control condition respectively.
`m`	m is a scalar representing the size of the Modified Bootstrap Sample (i.e. Out of p samples/subjects, m samples/subjects are randomly drawn with replacement, which constitutes one Modified Bootstrap Sample).
`s`	s is a scalar representing the number of Modified Bootstrap samples (i.e. number of times each of the m samples/subjects will be resampled from p samples/subjects).
`Q`	Q is a scalar representing the quartile value of the gene rankscores (lies within 1/N to 1), usually the second quartile, i.e. 0.5 or third quartile i.e. 0.75.
`plot`	plot is a character string must either take logical value TRUE/FALSE representing whether to plot the statistical significance values of genes in the dataset.

Value

The function returns a vector of p-values for all the genes from the given statistical test in the gene space/dataset using Modified Bootstrap MRMR technique.

Author(s)

Samarendra Das

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
m=36
s=80
Q=0.5
pval.mbmr(x, y, m, s, Q, plot=FALSE)
```

---

rice_salt *A gene expression dataset of rice under salinity stress*

Description

This data has gene expression values of 200 genes over 40 samples/subjects for a salinity vs. control study in rice. These 40 samples belong to either of salinity stress or control condition (two class problem). This gene expression data is balanced type as the first 20 samples are under salinity stress and the later 20 samples are under control condition. The first row of the data contains the samples/subjects labels with entries are 1 and -1, where the label '1' and '-1' represent samples generated under salinity stress and control condition respectively.

Usage

```
data("rice_salt")
```

Format

A data frame with 200 rows as genes with 40 columns as samples/subjects. Each column (sample) represent the gene expression values of genes. Each column as microarray samples with labels -1 or 1 represents control or salinity stress respectively.

Details

The data is created by taking 200 genes from the large number of genes from NCBI GEO database. The rows are the genes and columns are the samples/subjects. The first half of the samples/subjects are generated under salinity stress condition and other half under control condition.The first row of the data contains the samples/subjects albels with entries are 1 and -1, where th label '1' and '-1' represent samples generated under salinity stress and control condition respectively.

Source

Gene Expression Omnibus: NCBI gene expression and hybridization array data repository.ncbi.nlm.nih.gov/geo/.

Examples

```
data(rice_salt)
```

`topsis.meth` *Selection of optimal gene selection method(s)/method(s) through multi-criteria decision analysis*

Description

The function enables to rank gene selection methods/method(s) under a multi-criteria decision making set up and further selection of optimum gene selection method using Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) approach.

Usage

```
topsis.meth(x)
```

Arguments

`x`	x is a M by C data frame representing the performance value of the methods under different criteria, where rows are the methods and columns are the criteria. The performance of the methods are adjudged based on mangitude of the criteria (i.e. higher the criteria value better is the method).

Value

The function returns a data frame consisting rows as method names and different columns with distance score for positive, negative ideal solution, TOPSIS score and ranks for respective methods.

Author(s)

Samarendra Das

References

Ahn BS (2011) Compatible weighting method with rank order centroid: Maximum entropy ordered weighted averaging approach. Eur J Oper Res 212: 552-559.

Examples

```
x=matrix(runif(150), 10, 15)
rownames(x)=paste("Method",1:nrow(x), sep="")
colnames(x)=paste("C",1:ncol(x), sep="")
x=as.data.frame(x)
topsis.meth(x)
```

`weight.mbmr` *Computation of weights for informative gene selection using Modified Bootstrap MRMR technique*

Description

Weights associated with genes in a dataset computed from the Modified Bootstrap MRMR technique will provide a reliable measure for informative gene selection.

Usage

```
weight.mbmr(x, y, m, s, plot)
```

Arguments

`x`	x is a N by p dataframe of gene expression, where rows are genes and columns are as samples/subjects (gene names are taken as row names).
`y`	y is a p by 1 numeric vector with entries 1 and -1 representing sample labels, where 1 and -1 represents the sample label of subjects/ samples for stress and control condition respectively.
`m`	m is a scalar representing the size of the Modified Bootstrap Sample (i.e. Out of p samples/subjects, m samples/subjects are randomly drawn with replacement, which constitutes one Modified Bootstrap Sample).
`s`	s is a scalar representing the number of Modified Bootstrap samples (i.e. number of times each of the m samples/subjects will be resampled from p samples/subjects).
`plot`	plot is a character string must either take logical value TRUE/FALSE representing whether to plot the weights of genes in the dataset.

Details

The function returns a vector of weights associated with each genes in the dataset using Modified Bootstrap MRMR technique.

Author(s)

Samarendra Das

References

Wang J, Chen L, Wang Y, Zhang J, Liang Y, Xu D (2013) A Computational systems biology study for understanding salt tolerance mechanism in Rice. PLoS one 8(6): e64929.

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
m=36
s=80
weight.mbmr(x, y, m, s, plot=FALSE)
```

`Weights.mrmr` *Computation of MRMR weights for gene selection*

Description

The MRMR weights associated with each gene in the dataset are computed by using the MRMR technique for informative gene selection.

Usage

```
Weights.mrmr(x, y)
```

Arguments

`x`	x is a N by p dataframe of gene expression, where, rows as genes and columns as samples (with row names as gene names/ids)
`y`	y is a p by 1 numeric vector with entries 1 and -1 representing sample labels, where 1 and -1 represents the sample label of subjects/ samples for stress and control condition respectively.

Details

This function returns a vector of MRMR weights for all genes in the dataset.

Author(s)

Samarendra Das

References

Ding, C and Peng, H (2005). Minimum redundancy feature selection from microarray gene expression data. J. Bioinformatics Comput Biol 3(2):185-205.

Examples

```
data(rice_salt)
x=as.data.frame(rice_salt[-1,])
y=as.numeric(rice_salt[1,])
Weights.mrmr(x, y)
```

Index