Package ‘propOverlap’

October 14, 2022

Type Package

Title Feature (gene) selection based on the Proportional Overlapping Scores

Version 1.0

Date 2014-09-15

Author
Osama Mahmoud, Andrew Harrison, Aris Perperoglou, Asma Gul, Zardad Khan, Berthold Lausen

Maintainer Osama Mahmoud <ofamah@essex.ac.uk>

Description A package for selecting the most relevant features (genes) in the high-dimensional binary classification problems. The discriminative features are identified using analyzing the overlap between the expression values across both classes. The package includes functions for measuring the proportional overlapping score for each gene avoiding the outliers effect. The used measure for the overlap is the one defined in the ``Proportional Overlapping Score (POS)'' technique for feature selection. A gene mask which represents a gene's classification power can also be produced for each gene (feature). The set size of the selected genes might be set by the user. The minimum set of genes that correctly classify the maximum number of the given tissue samples (observations) can be also produced.

Depends R (>= 2.10), Biobase

LazyLoad yes

License GPL (>= 2)

Repository CRAN

NeedsCompilation no

Date/Publication 2014-09-15 17:06:03

R topics documented:

`propOverlap-package` *Feature (gene) selection based on the Proportional Overlapping Scores.*

Description

A package for selecting the most relevant features (genes) in the high-dimensional binary classification problems. The discriminative features are identified using analyzing the overlap between the expression values across both classes. The package includes functions for measuring the proportional overlapping score for each gene avoiding the outliers effect. The used measure of the overlap is the one defined in the “Proportional Overlapping Score (**POS**)” technique for feature selection, see ‘References’ section below. A gene mask which represents a gene’s classification power can also be produced for each gene (feature). The set size of the selected genes might be set by the user. The minimum set of genes that correctly classify the maximum number of the given tissue samples (observations) can be also produced.

Details

Package:	propOverlap
Type:	Package
Version:	1.0
Date:	2014-09-15
License:	GPL (>= 2)

Author(s)

Osama Mahmoud, Andrew Harrison, Aris Perperoglou, Asma Gul, Zardad Khan, Berthold Lausen
Maintainer: Osama Mahmoud `<ofamah@essex.ac.uk>`

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274

CI.emprical *Computing the Core Intervals for Both Classes.*

Description

CI.emprical is used to compute the core interval boundaries for each class.

Usage

```
CI.emprical(ES, Y)
```

Arguments

ES	gene (feature) matrix: P, number of genes, by N, number of samples(observations).
Y	a vector of length N for samples' class label.

Value

CI.emprical returns an object of class “data.frame” which has P rows and 4 columns. The first two columns represent a1, the minimum boundary of the first class, and b1, the maximum boundary of the first class, respectively. Whereas, the last two columns represent a2, the minimum boundary of the second class, and b2, the maximum boundary of the second class, respectively.

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

Examples

```
data(lung)
GenesExpression <- lung[1:12533,]   #define the features matrix
Class           <- lung[12534,]     #define the observations' class labels
CoreIntervals   <- CI.emprical(GenesExpression, Class)
CoreIntervals[1:10,]              #show classes' core interval for the first 10 features
```

GMask *Producing Gene Masks.*

Description

GMask produces the masks of features (genes). Each gene mask reports the samples that can unambiguously be assigned to their correct target classes by this gene.

Usage

```
GMask(ES, Core, Y)
```

Arguments

ES	gene (feature) matrix: P, number of genes, by N, number of samples(observations).
Core	a data.frame of the core interval boundaries for both classes. It should have the same number of rows as ES and 4 columns (the minimum and the maximum of the first class's core interval followed by the minimum and the maximum of the second class's core interval). See the returned value of the CI.emprical.
Y	a vector of length N for samples' class label.

Details

GMask gives the gene masks that can represent the capability of genes to correctly classify each sample. Such a mask represents a gene's classification power. Each element of a mask is set either to 1 or 0 based on whether the corresponding sample (observation) could be unambiguously assign to its correct target class by the considered gene or not respectively.

Value

It returns a P by N matrix with elements of zeros and ones.

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

See Also

CI.emprical for the core interval boundaries.

Examples

```
data(leukaemia)
GenesExpression <- leukaemia[1:7129,] #define the features matrix
Class           <- leukaemia[7130,]   #define the observations' class labels
Gene.Masks      <- GMask(GenesExpression, CI.emprical(GenesExpression, Class), Class)
Gene.Masks[1:100,]                    #show the masks of the first 100 features
```

`leukaemia` *Leukaemia data set.*

Description

The leukemia dataset was taken from a collection of leukemia patient samples reported by Golub et. al., (1999). This dataset often serves as a benchmark for microarray analysis methods. It contains gene expressions corresponding to acute lymphoblast leukemia (ALL) and acute myeloid leukemia (AML) samples from bone marrow and peripheral blood. The dataset consisted of 72 samples: 49 samples of ALL; 23 samples of AML. Each sample is measured over 7,129 genes.

Usage

```
data(leukaemia)
```

Format

A matrix with 7130 rows (7129 rows show the gene expressions while the last row reports the corresponding sample's class label), and 72 columns represent the samples. The samples class's label coded as follows:

`1` acute lymphoblast leukemia sample (ALL).

`2` acute myeloid leukemia sample (AML).

Source

http://cilab.ujn.edu.cn/datasets.htm

References

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES. (1999) *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring*. Science: 286 (5439), 531-537.

Examples

```
data(leukaemia)
str(leukaemia)
```

`lung` *Lung cancer data set.*

Description

Gene expression data for lung cancer classification between two classes: adenocarcinoma (ADCA); malignant pleural mesothe-lioma (MPM). The lung data set contains 181 tissue samples (150 ADCA and 31 MPM). Each sample is described by 12533 genes.

Usage

```
data(lung)
```

Format

A matrix with 12534 rows (12533 rows show the gene expressions for 181 tissue samples, reported in columns, while the last row reports the corresponding sample's class label). The samples class's label coded as follows:

`1` adenocarcinoma sample (ADCA).

`2` malignant pleural mesothe-lioma sample (MPM).

Source

http://cilab.ujn.edu.cn/datasets.htm

References

Gordon GJ, Jensen RV, Hsiao L-L, Gullans SR, Blumenstock JE, Ramaswamy S, Richards WG, Sugarbaker DJ, Bueno R. (2002) *Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma.* Cancer research: 62(17), 4963-4967.

Examples

```
data(lung)
str(lung)
```

POS *Calculating the proportional Overlapping Scores.*

Description

POS computes the proportional overlapping scores of the given genes (features). This score measures the overlap degree between gene expression values across various classes. It produces a value lies in the interval [0,1]. A lower score denotes gene with higher discriminative power for the considered classification problem.

Usage

```
POS(ES, Core, Y)
```

Arguments

ES	gene (feature) matrix: P, number of genes, by N, number of samples(observations).
Core	a data.frame of the core interval boundaries for both classes. It should have the same number of rows as ES and 4 columns (the minimum and the maximum of the first class's core interval followed by the minimum and the maximum of the second class's core interval). See the returned value of the CI.emprical.
Y	a vector of length N for samples' class label.

Details

For each gene, POS computes a measure that estimates the overlapping degree between the expression intervals of different classes. For estimating the overlap, POS measure takes into account three factors: the length of the overlapping region; number of the overlapped samples (observations); the proportion of each class's overlapped samples to the total number of overlapping samples.

Value

It returns a vector of length P for 'POS' measures of all genes (features).

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

See Also

CI.emprical for the core interval boundaries and GMask for the gene masks.

Examples

```
data(leukaemia)
Score <- POS(leukaemia[1:7129,], CI.emprical(leukaemia[1:7129,],
leukaemia[7130,]), leukaemia[7130,])
Score[1:5]      #show the proportional overlapping scores for the first 5 features
summary(Score)  #show the the summary of the scores of all features.
```

RDC *Assiging the Relative Dominant Class.*

Description

RDC associates genes (features) with the class which it is more able to distingish. For each gene, a class that has the highest proportion, relative to classes' size, of correctly assigned samples (observations) is reported as the relative dominant class for the considered gene.

Usage

```
RDC(GMask, Y)
```

Arguments

GMask	gene (feature) mask matrix: P, number of genes, by N, number of samples(observations) with elements of zeros and ones. See the returned value of the GMask.
Y	a vector of length N for samples' class label.

Value

RDC returns a vector of length P. Each element's value is either 1 or 2 indicating which class label is reported as the relative dominant class for the corresponding gene (feature).

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

See Also

GMask for gene (feature) mask matrix.

Examples

```
data(lung)
Class          <- lung[12534,]   #define the observations' class labels
Gene.Masks     <- GMask(lung[1:12533,], CI.emprical(lung[1:12533,], Class), Class)
RelativeDC     <- RDC(Gene.Masks, Class)
RelativeDC[1:10]            #show the relative dominant classes for the first 10 features
table(RelativeDC)                #show the number of assignments for each class
```

Sel.Features *Gene (Feature) Selection.*

Description

Sel.Feature selects the most discriminative genes (features) among the given ones.

Usage

```
Sel.Features(ES, Y, K = "Min", Verbose = FALSE)
```

Arguments

ES	gene (feature) matrix: P, number of genes, by N, number of samples (observations).
Y	a vector of length N for samples' class label.
K	the number of genes to be selected. The default is to give the minimum subset of genes that correctly classify the maximum number of the given tissue samples (observations). Alternatively, K should be a positive integer.
Verbose	logical. If TRUE, more information about the selected genes are returned.

Details

Sel.Feature selects the most relevant genes (features) in the high-dimensional binary classification problems. The discriminative genes are identified using analyzing the overlap between the expression values across both classes. The "**POS**" technique has been applied to produce the selected set of genes. A proportional overlapping score measures the overlapping degree avoiding the outliers effect for each gene. Each gene is described by a robust mask that represents its discriminative power. The constructed masks along with the gene scores are exploited to produce the selected subset of genes.

Value

If K is specified as 'Min' (the default), a list containing the following components is returned:

Features	A matrix of the indices of selected genes with their POS measures. See POS.
Covered.Obs	A vector showing the indices of the observations that have been covered by the returned minimum subset of genes.

If K is specified as a positive integer, a list containing the following components is returned:

features	A vector of the indices of the selected genes.
nMin.Features	The number of genes included in the minimum subset.
Measures	Available only when Verbose is TRUE. It is an object with class "data.frame" which contains 3 columns: the indices of the selected genes; the POS measures of the selected genes (see POS); the status that reports on which basis a gene is selected ("Min.Set": the gene is a member of the selected minimum subset, 1: the gene has a low POS score and its relative dominant class is the first class or 2: the gene has a low POS score and its relative dominant class is the second class), see RDC.

Note

Verbose is only needed when K is specified. If K is set to "Min" (default), all information are automatically returned.

Author(s)

Osama Mahmoud <ofamah@essex.ac.uk>

References

Mahmoud O., Harrison A., Perperoglou A., Gul A., Khan Z., Metodiev M. and Lausen B. (2014) *A feature selection method for classification within functional genomics experiments based on the proportional overlapping score*. BMC Bioinformatics, 2014, 15:274.

See Also

POS for calculating the proportional overlapping scores and RDC for assigning the relative dominant class.

Examples

```
data(leukaemia)
GenesExpression <- leukaemia[1:7129,] #define the features matrix
Class           <- leukaemia[7130,]   #define the observations' class labels
## select the minimum subset of features
Selection       <- Sel.Features(GenesExpression, Class)
attributes(Selection)
(Candidates      <- Selection$Features)   #return the selected features
(Covered.observations <- Selection$Covered.Obs) #return the covered observations by the selection
## select a specific number of features
Selection.k      <- Sel.Features(GenesExpression, Class, K=10, Verbose=TRUE)
Selection.k$Features
Selection.k$nMin.Features   #return the size of the minimum subset of genes
Selection.k$Measures        #return the selected features' information
```

Index