Package ‘netClass’

February 20, 2015

Version 1.2.1
Date 2013-12-03
Title netClass: An R Package for Network-Based Biomarker Discovery
Author Yupeng Cun
Maintainer Yupeng Cun <yupeng.cun@gmail.com>

Description netClass is an R package for network-based feature (gene) selection for biomarkers discovery via integrating biological information. This package adapts the following 5 algorithms for classifying and predicting gene expression data using prior knowledge: 1) average gene expression of pathway (aep); 2) pathway activities classification (PAC); 3) Hub network Classification (hubc); 4) filter via top ranked genes (FrSVM); 5) network smoothed t-statistic (stSVM).

Depends R (>= 2.14), kernlab
Imports AnnotationDbi, Matrix, ROCR, graph, igraph, samr
Suggests parallel, Biobase, KEGG.db, pathClass
License GPL (>= 2)
LazyLoad yes
NeedsCompilation no
Repository CRAN
Date/Publication 2013-12-03 22:44:46

R topics documented:

netClass-package .................................................. 2
ad.matrix .......................................................... 3
calc.diffusionKernelp ........................................... 3
classify.aep ....................................................... 4
classify.frsvm ..................................................... 5
classify.hubc ...................................................... 6
classify.pac ....................................................... 8
An R package for network-Based microarray Classification

Description

We implemented average gene expression of pathway (aep), pathway activitive classification (PAC), Hub network Classsification, filter via top ranked genes(FrSVM), smoothed t-statistic(stSVM) for two classes microarray classification which employed the prior information.

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>netClass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.2</td>
</tr>
<tr>
<td>Date:</td>
<td>2013-09-09</td>
</tr>
<tr>
<td>License:</td>
<td>GPL (&gt;= 2)</td>
</tr>
<tr>
<td>LazyLoad:</td>
<td>yes</td>
</tr>
</tbody>
</table>
Author(s)
Yupeng Cun
Maintainer: Yupeng Cun <yupeng.cun@gmail.com>

References

ad.matrix  
An adjacency matrix of a sample graph...

Description
An adjacency matrix of a sample graph

Details
An adjacency matrix of a random graph with some random Entre ID of Protein for use in example files and the vignette

Authors
Yupeng Cun <yupeng.cun@gmail.com>

calc.diffusionKernelp
Computing the Random Walk Kernel matrix of network

Description
Computing the Random Walk Kernel matrix of network

Usage
calc.diffusionKernelp(L, is.adjacency = TRUE, p = 3, a = 2)

Arguments
L  an adjacency matrix that represents the underlying biological network.
is.adjacency  using adjacency of graph or not
p  #(p) random walk step(s) of random walk kernel
a  constant value of random walk kernel
Return a Random Walk Kernel matrix of given network, L.

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


See Also

See Also as classify.stsvm

Examples

```r
library(netClass)
data(ad.matrix)
#dk = calc.diffusionKernelp(L=ad.matrix, is.adjacency=TRUE, p=2, a=1)
```

classify.aep Training and predicting using aepSVM (aepSVM) classification methods

Description

Training and predicting using aepSVM (aepSVM) classification methods

Usage

```r
classify.aep(fold, cuts, Cs, x, y, cv.repeat, int, DEBUG = DEBUG, Gsub)
```

Arguments

- `fold` number of -folds cross validation (CV)
- `cuts` list for randomly divide the training set in to x-x-folds CV
- `Cs` soft-margin tuning parameter of the SVM. Defaults to $10^{-c(-3:3)}$.
- `x` gene expression data
- `y` class labels
- `cv.repeat` model for one CV training and predicting
- `int` Intersect of genes in network and gene expression profile.
- `DEBUG` show debugging information in screen more or less.
- `Gsub` an adjacency matrix that represents the underlying biological network.
classify.frsvm

Value

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fold</td>
<td>the recored for test fold</td>
</tr>
<tr>
<td>auc</td>
<td>The AUC values of test fold</td>
</tr>
<tr>
<td>train</td>
<td>The tranined models for traning folds</td>
</tr>
<tr>
<td>feat</td>
<td>The feature selected by each by the train</td>
</tr>
</tbody>
</table>

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


See Also

See Also as cv.aep

Examples

#See cv.aep

classify.frsvm

Training and predicting using FrSVM

Description

Training and predicting using FrSVM

Usage

classify.frsvm(fold, cuts, x, y, cv.repeat, DEBUG = DEBUG, Gsub = Gsub, d = d, op = op, aa = aa, Cs = Cs)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fold</td>
<td>number of folds to perform</td>
</tr>
<tr>
<td>cuts</td>
<td>list for randomly divide the training set in to x-x-CV</td>
</tr>
<tr>
<td>x</td>
<td>expression data</td>
</tr>
<tr>
<td>y</td>
<td>a factor of length p comprising the class labels.</td>
</tr>
<tr>
<td>cv.repeat</td>
<td>model for one CV training and predicting</td>
</tr>
<tr>
<td>DEBUG</td>
<td>show debugging information in screen more or less.</td>
</tr>
<tr>
<td>Gsub</td>
<td>an adjacency matrix that represents the underlying biological network.</td>
</tr>
</tbody>
</table>
classify.hubc

d damping factor for GeneRank, defaults value is 0.5
op the upper bound of top ranked genes
aa the lower bound of top ranked genes
Cs soft-margin tuning parameter of the SVM. Defaults to $10^{-3\cdot3}$.

Value
fold the recorded for test fold
auc The AUC values of test fold
train The trained models for training folds
feat The feature selected by each by the train

Author(s)
Yupeng Cun <yupeng.cun@gmail.com>

References

See Also
See Also as cv.frsvm

Examples
#see cv.frsvm

classify.hubc Training and predicting using hub nodes classification methods

Description
Training and predicting using hub nodes classification methods

Usage
classify.hubc(fold, r, cuts, x, y, cv.repeat, Gsub = Gsub, DEBUG = DEBUG, gHub = gHub, hubs = hubs, nperm = nperm, node.ct = node.ct, Cs = Cs)
classify.hubc

Arguments

fold  number of -fold cross validation (CV)
cuts  list for randomly divide the training set in to x-x-fold CV
Gsub  an adjacency matrix that represents the underlying biological network.
x    gene expression data.
y    a factor of length p comprising the class labels.
cv.repeat model for one CV training and predicting
DEBUG show debugging information in screen more or less.
r    repeat order for CV
gHub Subgraph of hubs of graph Gs
hubs Hubs in graph Gs
nperm number of permutation test steps
node.ct cut off value for select highly quantile nodes in a network. Defaults to 0.98).
Cs  Soft-margin tuning parameter of the SVM. Defaults to 10^c(-3:3).

Value

fold    the recored for test fold
auc     The AUC values of test fold
train   The tranined models for traning folds
feat    The feature selected by each by the train

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


See Also

See cv.hubc

Examples

#See cv.hubc
classify.pac

Training and predicting using PAC classification methods

Description

Training and predicting using PAC classification methods

Usage

classify.pac(fold, cuts, x, y, cv.repeat, Gsub, int, DEBUG = FALSE)

Arguments

fold number of -folds cross validation (CV)
cuts list for randomly divide the training set in to x-x-folds CV
Gsub an adjacency matrix that represents the underlying biological network.
x gene expression data
y a factor of length p comprising the class labels.
cv.repeat model for one CV training and predicting
int Intersect of genes in network and gene expression profile.
DEBUG show debugging information in screen or not.

Value

fold the recored for test fold
auc The AUC values of test fold
train The tranined models for traning folds
feat The feature selected by each by the train

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


See Also

See Also as cv.pac

Examples

#see cv.pac
Training and predicting using stSVM classification methods

Usage

classify.stsvm(fold, cuts, ex.sum, x, p, a, y, cv.repeat, DEBUG = DEBUG, Gsub=Gsub,  op.method=op.method, op = op, aa = aa, dk = dk, dk.tf = dk.tf, seed = seed, Cs = Cs)

Arguments

fold number of folds to perform
cuts list for randomly divide the training set in to x-x-folds CV
ex.sum expression data
x expression data
a constant value of random walk kernel
p random walk step(s) of random walk kernel
y a factor of length p comprising the class labels.
cv.repeat model for one CV training and predicting
DEBUG show debugging information in screen more or less.
Gsub an adjacency matrix that represents the underlying biological network.
op.method Method for selecte optimal feature subgoups: pt is permutation test, sp is span bound.
op optimal on top op
aa permutation test steps
dk Random Walk Kernel matrix of network
dk.tf cut off p-value of permutation test
seed seed for random sampling.
Cs Soft-margin tuning parameter of the SVM. Defaults to 10^c(-3:3).

Value

fold the recored for test fold
auc The AUC values of test fold
train The tranined models for traning folds
feat The feature selected by each by the train
Author(s)
Yupeng Cun <yupeng.cun@gmail.com>

References

See Also
see cv.stsvm

Examples
#see cv.stsvm

---

cv.aep

Cross validation for aepSVM (aepSVM)

Description
Cross validation for aepSVM (aepSVM) using SAM to select significant differential expressed genes

Usage
cv.aep(x, y, folds = 10, repeats = 5, parallel = FALSE, cores = 2, DEBUG = TRUE, Gsub = matrix(1, 100, 100), Cs = 10^(-3:3), seed = 1234)

Arguments
- `x`: a p x n matrix of expression measurements with p samples and n genes.
- `y`: a factor of length p comprising the class labels.
- `folds`: number of -folds cross validation (CV)
- `repeats`: number of CV repeat times
- `parallel`: parallel computing or not
- `cores`: cores used in parallel computing
- `DEBUG`: show more results or not
- `Gsub`: Adjacency matrix of Protein-protein interaction network
- `Cs`: soft-margin tuning parameter of the SVM. Defaults to 10^(-3:3).
- `seed`: seed for random sampling.
Value

- **value**
  - A LIST for Cross-Validation results
  - **auc**
    - The AUC values of each test fold
  - **fits**
    - The trained models for training folds
  - **feat**
    - The feature selected by each by the fits
  - **labels**
    - The original labels for training

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


Examples

```r
library(netClass)
data(expr)
data(ad.matrix)
x <- expr$genes
y <- expr$y

library(KEGG.db)
#r.aep <- cv.aep(x[,1:500], y, folds=3, repeats=1, parallel=FALSE, cores=2,
# Gsub=ad.matrix, Cs=10^(-3:3), seed=1234, DEBUG=TRUE)
```

---

**cv.frsvm**

Cross validation for FrSVM

Description

Cross validation for FrSVM, an R algorithm, which integrates protein-protein interaction network information into gene selection for microarray classification.

Usage

```r
cv.frsvm(x, y, folds = 10, Gsub = matrix(1, 100, 100), repeats = 5, parallel = FALSE, cores = 2, DEBUG = FALSE, d = 0.85, top.upper = 10, top.lower = 50, seed = 1234, Cs = 10^c(-3:3))
```
Arguments

- **x**: gene expression data
- **y**: class labels
- **folds**: number of -folds cross validation (CV)
- **Gsub**: Adjacency matrix of Protein-protein intersection network
- **repeats**: number of CV repeat times
- **parallel**: parallel computing or not
- **cores**: cores used in parallel computing
- **DEBUG**: show more results or not
- **d**: damping factor for GeneRank, defaults value is 0.5
- **top.upper**: the upper bound of top ranked genes
- **top.lower**: the lower bound of top ranked genes
- **seed**: Seed for random sampling.
- **Cs**: soft-margin tuning parameter of the SVM. Defaults to $10^c(-3:3)$.

Value

- a LIST for Cross-Validation results
- **auc**: The AUC values of each test fold
- **fits**: The trained models for training folds
- **feat**: The feature selected by each by the fits
- **labels**: the original labels for training

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


Examples

library(netClass)
data(expr)
data(ad.matrix)
x <- expr$genes
y <- expr$y
###
cv.hubc <- cv.frsvm(x[,1:200], y, folds=3, Gsub=ad.matrix, repeats=1, parallel=FALSE, cores=2, DEBUG=TRUE, d=.85, top.upper=5, top.lower=15, seed=1234, Cs=10^c(-3:3))

---

**cv.hubc**

*Cross validation for hub nodes classification*

**Description**

Cross validation for hub nodes classification, which described in Taylor et al.(2009).

**Usage**

```r
cv.hubc(x, y, folds = 10, repeats = 5, parallel = TRUE, cores = NULL, DEBUG = TRUE, nperm = 500, node.ct = 0.98, Gsub = matrix(1, 100, 100), Gs = Gs, seed = 1234, Cs = 10^c(-3:3))
```

**Arguments**

- `x`: a p x n matrix of expression measurements with p samples and n genes.
- `y`: a factor of length p comprising the class labels.
- `folds`: number of -folds cross validation (CV)
- `repeats`: number of CV repeat times
- `parallel`: parallel computing or not
- `cores`: cores used in parallel computing
- `DEBUG`: show more results or not
- `nperm`: number of permutation test steps
- `node.ct`: cut off value for select highly quantile nodes in a nwtwork. Defaults to 0.98).
- `Gsub`: an adjacency matrix that represents the underlying biological network.
- `Gs`: Undirected of graph with adjacency matrix Gsub.
- `seed`: Seed for random sampling.
- `Cs`: Soft-margin tuning parameter of the SVM. Defaults to 10^c(-3:3).

**Value**

- `auc`: The AUC values of each test fold
- `fits`: The trained models for training folds
- `feat`: The selected features of each training folds
- `labels`: the original labels for training

**Author(s)**

Yupeng Cun <yupeng.cun@gmail.com>
References


Examples

```r
data(ad.matrix)
# data(Gs2)
library(netClass)
data(expr)
x <- expr$genes
y <- expr$y

# r.hubC <- cv.hubc(x=x, y=y, folds=3, repeats=1, parallel=FALSE, cores=2, DEBUG=TRUE,
# nperm=2, Gsub=ad.matrix,Gs=Gs2,node.ct=0.5,Cs=10*(-3:3))
```

---

**cv.pac**

*Cross validation for Pathway Activities Classification (PAC)*

**Description**

Cross validation for Pathway Activities Classification (PAC) using Logistic regression model for classification. Implementation of the Pathway Activities Classification by CROG algorithm.

**Usage**

```r
cv.pac(x=x, y=y, folds=10, repeats=5, parallel = TRUE, cores = NULL,
DEBUG=TRUE, Gsub=matrix(1,100,100), seed=1234)
```

**Arguments**

- **x**: a p x n matrix of expression measurements with p samples and n genes.
- **y**: a factor of length p comprising the class labels.
- **folds**: number of -folds cross validation (CV)
- **repeats**: number of CV repeat times
- **parallel**: parallel computing or not
- **cores**: cores used in parallel computing
- **DEBUG**: show debugging information in screen or not.
- **Gsub**: Adjacency matrix of Protein-protein interaction network
- **seed**: seed for random sampling.
Value

- a LIST for Cross-Validation results
  - auc: The AUC values of each test fold
  - fits: The trained models for training folds
  - feat: The feature selected by each by the fits
  - labels: the original labels for training

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


Examples

```r
library(netClass)

data(expr)
data(ad.matrix)
x <- expr$genes
y <- expr$y

library(KEGG.db)
r.pac <- cv.stsvm(x=x, y=y, folds=3, repeats=1, parallel=FALSE, cores=2, DEBUG=TRUE, Gsub=ad.matrix, seed=1234)
```

---

## cv.stsvm

*Cross validation for smoothed t-statistic to select significant top ranked differential expressed genes*

Description

Cross validation for smoothed t-statistic to select significant top ranked differential expressed genes

Usage

```r
cv.stsvm(x=x, x.mi=NULL, y=y, folds=5, Gsub=matrix(1,100,100), op.method=c("pt","spb"), repeats=3, parallel=FALSE, cores=2,DEBUG=TRUE, pt.pvalue=0.05, op=0.85, aa=1000,a=1,p=2,allF=TRUE, seed=1234,Cs=10^c(-3:3))
```
Arguments

x A p x n matrix of expression measurements with p samples and n genes.
x.mi A p x m matrix of expression measurements with p samples and m miRNAs.
y A factor of length p comprising the class labels.
folds Folds number of folds to perform
Gsub An adjacency matrix that represents the underlying biological network.
opNmethod Method for selecting optimal feature subgroups: pt is permutation test, sp is span bound.
repeats Number of how often to repeat the x-fold cross-validation
parallel Use parallel computing or not
cores Number of cores will be used when parallel is TRUE
DEBUG Show debugging information in screen more or less.
pt.pvalue Cut off p-value of permutation test
op Optimal on top op
aa permutation test steps for permutation test (pt); low bounds top op
a constant value of random walk kernel
p random walk step(s) of random walk kernel
allF Using all features (TRUE) or only these genes mapped to prior information (FALSE).
seed seed for random sampling.
Cs Soft-margin tuning parameter of the SVM. Defaults to $10^c(-3:3)$.

Value

a LIST for Cross-Validation results

auc The AUC values of each test fold
fits The trained models for training folds
feat The feature selected by each by the fits
labels the original labels for training

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References

Examples

```r
library(netClass)
data(expr)
data(ad.matrix)
x <- expr$genes
y <- expr$y

r.stsvm <- cv.stsvm(x=x[1:500], x.mi=NULL, y=y, folds=3, Gsub=ad.matrix, op.method="pt",
repeats=1, parallel=FALSE, cores=2, DEBUG=TRUE, pt.pvalue=0.05, op=0.9,
aa=5, a=1, p=2, allF=TRUE, seed=1234, Cs=10^(-3:3))
```

---

**EN2SY**

An list for mapping gene entre ids to symbol ids

**Description**

An list for mapping gene entre ids to symbol ids

**Details**

An list for mapping gene Entre ID of Symbol ID

**Author(s)**

Yupeng Cun <yupeng.cun@gmail.com>

---

**expr**

Two expression profile matrixs and their labels

**Description**

Two expression profile matrixs and their labels

**Details**

Two expression profile matrixs and thei labels of random samples. expr$genes is the expression profile with Entrez ID of genes; expr$y is labels of the expression profile.

**Author(s)**

Yupeng Cun <yupeng.cun@gmail.com>
### getGeneRanking

*Get gene ranking based on geneRank algorithm.*

**Description**

Get the ranking of differential expression of genes on graph using geneRank algorithm.

**Usage**

```r
getGeneRanking(x = x, y = y, Gsub = Gsub, d = d)
```

**Arguments**

- `x` gene expression data
- `y` class labels
- `Gsub` Adjacency matrix of Protein-protein interaction network
- `d` damping factor for GeneRank, defaults value is 0.5

**Value**

- `r` ranking of each genes on graph

**Author(s)**

Yupeng Cun <yupeng.cun@gmail.com>

**See Also**

See Also as pGeneRank

**Examples**

```r
library(netClass)
data(expr)
data(ad.matrix)
ex.sum <- expr$genes
y <- expr$y

#r= getGeneRanking(x = ex.sum, y = y, Gsub = ad.matrix, d = 0.5)
```
**getGraphRank**  

**Random walk kernel matrix smoothing t-statistic**

**Description**

Using Random walk kernel matrix of network to smooth t-statistic of each gene

**Usage**

```r
getGraphRank(x = x, y = y, Gsub = Gsub, sca = TRUE)
```

**Arguments**

- `x`: a matrix of expression measurements with p samples and n genes.
- `y`: a factor of length p comprising the class labels.
- `Gsub`: Random Walk Kernel matrix of network
- `sca`: Scaling data or not

**Value**

- `r`: return a smoothed t-statistic of each gene

**Author(s)**

Yupeng Cun <yupeng.cun@gmail.com>

**References**

Yupeng Cun, Holger Frohlich (2013) Network and Data Integration for Biomarker Signature Discovery via Network Smoothed T-Statistics

**See Also**

See Also as `getGraphRank`

**Examples**

```r
#See also \code{classfy.stsvm}
```
**pGeneRANK**

---

**Gs2**

An subgraph of hub nodes

---

**Description**

An subgraph of hub nodes, which using igraph to generate from hubs

**Details**

An adjacency matrix of hubs of a random graph was used to constructed a sub-graph of hubs using igraph

**Author(s)**

Yupeng Cun <yupeng.cun@gmail.com>

---

**pGeneRANK**

**GeneRANK**

---

**Description**

Ranking gene based on Googles’s PageRank algorithm

**Usage**

pGeneRANK(W, ex, d, max.degree = Inf)

**Arguments**

- **w**: adjacency matrix of graph
- **ex**: the fold change/ differential expression of genes
- **d**: damping factor for GeneRank, defaults value is 0.5
- **max.degree**: Max degree of graph

**Value**

- **r**: ranking of each genes on graph

**Author(s)**

Yupeng Cun <yupeng.cun@gmail.com>
References

See Also
See Also as classify.frsvm

Examples

```r
# See Also as {classify.frsvm}

pOfHubs
```

Computing p value of hubs using the permutation test

Description
Computing p value of hubs using the permutation test

Usage
```
pOfHubs(x = x, y = y, gHub = gHub, hubs = hubs, nperm = nperm)
```

Arguments
- `x`: gene expression data
- `y`: a factor of length p comprising the class labels.
- `gHub`: Subgraph of hubs of graph Gs
- `hubs`: Hubs in graph Gs
- `nperm`: number of permutation test steps

Value
- `pVal`: Permutation test Pvalues of each hub
- `hub`: name of hubs

Author(s)
Yupeng Cun <yupeng.cun@gmail.com>

Examples

```
# see \code{pOfHubs}
```
predictAep

Predicting the test data using aep trained model

Description

Predicting the test data using aep trained model

Usage

predictAep(train = train, x, y, DEBUG = FALSE, Gsub = Gsub)

Arguments

train trained model
x gene expression data for testing
y class labels
DEBUG show debugging information in screen more or less.
Gsub an adjacency matrix that represents the underlying biological network.

Value

The value returned

auc The AUC values of test fold

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

See Also

See Also as cv.aep

Examples

#see cv.aep
# predictFrsvm

**Description**

Predicting the test data using frsvm trained model

**Usage**

```r
predictFrsvm(train = train, x = x, y = y, DEBUG = FALSE)
```

**Arguments**

- `train`: trained model
- `x`: expression data for testing
- `y`: class labels
- `DEBUG`: show debugging information in screen more or less.

**Value**

- `auc`: The AUC values of test fold

**Author(s)**

Yupeng Cun <yupeng.cun@gmail.com>

**See Also**

See Also as `cv.frsvm`

**Examples**

```r
#see cv.frsvm
```

# predictHubc

**Description**

Predicting the test data using hubc trained model

**Usage**

```r
predictHubc(train = train, x = x, y = y, DEBUG = FALSE)
```

**See Also**

See Also as `cv.hubc`
predictPac

Arguments

train: trained model bases on hub nodes.

x: gene expression data for predicting.

y: Class labels

DEBUG: show debugging information in screen more or less.

Value

The value returned

auc: The AUC values of test fold

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

See Also

See Also as cv.hubc

Examples

#See cv.hubc

predictPac(train = train, x = x, y = y, int = int, DEBUG = FALSE)

Description

Predicting the test data using pac trained model

Usage

predictPac(train = train, x = x, y = y, int = int, DEBUG = FALSE)

Arguments

train

x: gene expression data for the testing data

y: a factor of length p comprising the class labels.

int: Intersect of genes in network and gene expression profile.

DEBUG: show debugging information in screen or not.
predictStsvm

Value

The value returned

auc  The AUC values of test fold

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

See Also

See Also as cv.pac

Examples

#see cv.pac

predictStsvm  Predicting the test data using stsvm trained model

Description

Predicting the test data using stsvm trained model

Usage

predictStsvm(train = train, x = x, y = y, DEBUG = DEBUG)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>train</td>
<td>trained model</td>
</tr>
<tr>
<td>x</td>
<td>expression data for testing</td>
</tr>
<tr>
<td>y</td>
<td>Class labels</td>
</tr>
<tr>
<td>DEBUG</td>
<td>show debugging information in screen more or less.</td>
</tr>
</tbody>
</table>

Value

The value returned

auc  The AUC values of test fold

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

See Also

See Also as cv.stsvm
probeset2pathway

Examples

# see cv.stsvm

probeset2pathway  Generates a mean gene expression of genes of each pathway matrix

Description

Generates a mean gene expression of genes of each pathway matrix

Usage

probeset2pathway(x = x, int = int, sigGens = sigGens)

Arguments

x  gene expression data
int  common genes between pathway genes and genes in gene expression profile
sigGens  significant gene expression using SAM methods

Value

kse  an matrix with n pathways and p samples

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


See Also

See Also as classify.aep
probeset2pathwayTrain  \textit{Search CROG in training data}

\textbf{Description}

Search CROG in training data, and using these CROP set to make a matrix for pathways.

\textbf{Usage}

\begin{verbatim}
probeset2pathwayTrain(x = x, y = y, int = int)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \textit{x}  \hspace{1cm} gene expression data
  \item \textit{y}  \hspace{1cm} a factor of length \(p\) comprising the class labels.
  \item \textit{int}  \hspace{1cm} Common genes between gene expression data and interaction network.
\end{itemize}

\textbf{Value}

\begin{itemize}
  \item \textit{ap}  \hspace{1cm} top ranked pathways
  \item \textit{selectedGenes}  \hspace{1cm} CROG genes
  \item \ldots
\end{itemize}

\textbf{Author(s)}

Yupeng Cun <yupeng.cun@gmail.com>

\textbf{References}


\textbf{See Also}

See Also as \texttt{pac.cv}

\textbf{Examples}

\begin{verbatim}
#See Also as \name{pac.cv}
\end{verbatim}
probeset2pathwayTst

Description

Applied CROG and pathways activities lists to make a matrix for pathways for test data.

Usage

```r
probeset2pathwayTst(x = x, apTrain = apTrain)
```

Arguments

- `x` gene expression data
- `apTrain` PAC object which contain CROG and pathways activities lists of training data.

Value

- `ap` top ranked pathways

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


See Also

See Also as `pac.cv`, `probeset2pathwayTrain`

Examples

```r
# See Also as \
```
train.aep  Training the data using aep methods

Description

Training the data using aep methods

Usage

train.aep(x = x, y = y, DEBUG = FALSE, int = int, Gsub = Gsub, Cs = 10^c(-3:3))

Arguments

x  
expression data for training

y  
a factor of length p comprising the class labels.

DEBUG  
show debugging information in screen more or less.

int  
Intersect of genes in network and gene expression profile.

Gsub  
an adjacency matrix that represents the underlying biological network.

Cs  
soft-margin tuning parameter of the SVM. Defaults to 10^c(-3:3).

Value

The returned lists

trained  
The tranined models for traning folds

sig.genes  
The differential expressed feature

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References


See Also

See Also as cv.aep

Examples

#see cv.aep
train.frsvm  

*Training the data using frsvm method*

**Description**

Training the data using frsvm methods

**Usage**

```r
train.frsvm(x = x, y = y, DEBUG = FALSE, Gsub = Gsub, d = 0.85, op
            = 10, aa = 50, Cs = 10^(-3:3))
```

**Arguments**

- `x`: Expression data for training
- `y`: Class labels
- `DEBUG`: show debugging information in screen more or less.
- `Gsub`: an adjacency matrix that represents the underlying biological network.
- `d`: damping factor for GeneRank, defaults value is 0.5
- `op`: the upper bound of top ranked genes
- `aa`: the lower bound of top ranked genes
- `Cs`: soft-margin tuning parameter of the SVM. Defaults to `10^(-3:3)`.

**Value**

The value list returned

- `train`: The trained models for training folds
- `feat`: The feature selected by each by the train

**Author(s)**

Yupeng Cun <yupeng.cun@gmail.com>

**See Also**

See Also as `cv.frsvm`

**Examples**

```
#see cv.frsvm
```
train.hubc

Predicting the data using hub nodes classification model

Description
Predicting the data using hub nodes classification model

Usage
```r
train.hubc(x = x, y = y, DEBUG = FALSE, Gsub = Gsub, gHub = gHub,
           hubs = hubs, nperm = 500, node.ct = 0.95, Cs = 10^(-3:3))
```

Arguments
- `x`: gene expression data for training.
- `y`: Class labels
- `DEBUG`: show debugging information in screen more or less.
- `Gsub`: an adjacency matrix that represents the underlying biological network.
- `gHub`: Subgraph of hubs of graph Gs
- `hubs`: Hubs in graph Gs
- `nperm`: number of permutation test steps
- `node.ct`: cut off value for select highly quantile nodes in a network. Defaults to 0.98.
- `Cs`: Soft-margin tuning parameter of the SVM. Defaults to 10^(-3:3).

Value
The list returned
- `trained`: The trained models for training folds
- `feat`: The feature selected by each by the train

Author(s)
Yupeng Cun <yupeng.cun@gmail.com>

See Also
See Also as cv.hubc

Examples
```r
#See cv.hubc
```
Training the data using pac methods

Description

Training the data using pac methods

Usage

train.pac(x = x, y = y, int = int, DEBUG = FALSE, Gsub = Gsub)

Arguments

- **x**: gene expression data for the training data
- **y**: a factor of length p comprising the class labels.
- **int**: Intersect of genes in network and gene expression profile.
- **DEBUG**: show debugging information in screen or not.
- **Gsub**: an adjacency matrix that represents the underlying biological network.

Value

- **trained**: The trained models for training folds

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

See Also

See Also as cv.pac

Examples

#see cv.pac
train.stsvm

Training the data using stsvm methods

Description

Training the data using stsvm methods

Usage

train.stsvm(x=x, y=y, DEBUG=FALSE, Gsub=Gsub, op.method="sp", op=10, aa=100, dk=dk, dk.tf=0.05, seed = 1234, Cs=10^(-3:3), EN2SY=NULL)

Arguments

x expression data for training
y Class labels
DEBUG show debugging information in screen more or less.
Gsub an adjacency matrix that represents the underlying biological network.
op.method Method for select optimal feature subgroups: pt is permutation test, sp is span bound.
op optimal on top op
aa permutation test steps
dk Random Walk Kernel matrix of network
dk.tf cut off p-value of permutation test
seed seed for random sampling.
Cs Soft-margin tuning parameter of the SVM. Defaults to 10^c(-3:3).
EN2SY A list for mapping gene symbol ids or entrez ids.

Value

The list returned

trained The trained models for training folds
feat The feature selected by each by the train

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

See Also

See cv.stsvm

Examples

# see cv.stsvm
Index

*Topic FrSVM
  getGeneRanking, 18
  pGeneRank, 20
*Topic GeneRank
  train.frsvm, 30
*Topic aep
  classify.aep, 4
  cv.aep, 10
  predictAep, 22
  probeset2pathway, 26
  train.aep, 29
*Topic biomarker discovery, microarray classification, interaction network
  netClass-package, 2
*Topic data
  ad.matrix, 3
  EN2SY, 17
  expr, 17
  Gs2, 20
*Topic frsvm
  classify.frsvm, 5
  cv.frsvm, 11
  predictFrsvm, 23
  train.frsvm, 30
*Topic hubc
  classify.hubc, 6
  cv.hubc, 13
  pOfHubs, 21
  predictHubc, 23
  train.hubc, 31
*Topic pac
  classify.pac, 8
  cv.pac, 14
  predictPac, 24
  probeset2pathwayTrain, 27
  probeset2pathwayTst, 28
  train.pac, 32
*Topic stSVM
  calc.diffusionKernelp, 3
  cv.stsvm, 15
  getGraphRank, 19
*Topic stsvm
  classify.stsvm, 9
  predictStsvm, 25
  train.stsvm, 33
  ad.matrix, 3
  calc.diffusionKernelp, 3
  classify.aep, 4
  classify.frsvm, 5
  classify.hubc, 6
  classify.pac, 8
  classify.stsvm, 9
  cv.aep, 10
  cv.frsvm, 11
  cv.hubc, 13
  cv.pac, 14
  cv.stsvm, 15
  EN2SY, 17
  expr, 17
  getGeneRanking, 18
  getGraphRank, 19
  Gs2, 20
  netClass (netClass-package), 2
  netClass-package, 2
  pGeneRank, 20
  pOfHubs, 21
  predictAep, 22
  predictFrsvm, 23
  predictHubc, 23
  predictPac, 24
  predictStsvm, 25
  probeset2pathway, 26
  probeset2pathwayTrain, 27
INDEX

probeset2pathwayTst, 28
train.aep, 29
train.frsvm, 30
train.hubc, 31
train.pac, 32
train.stsvm, 33