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

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
pGeneRANK . . . 20
pOfHubs . . . 21
predictAep . . . 22
predictFrsvm . . . 23
predictHubc . . . 23
predictPac . . . 24
predictStsvm . . . 25
probeset2pathway . . . 26
probeset2pathwayTrain . . . 27
probeset2pathwayTst . . . 28
train.aep . . . 29
train.frsvm . . . 30
train.hubc . . . 31
train.pac . . . 32
train.stsvm . . . 33

Index **34**

`netClass-package` *An R package for network-Based microarray Classification*

Description

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

Details

Package:	netClass
Type:	Package
Version:	1.2
Date:	2013-09-09
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Yupeng Cun

Maintainer: Yupeng Cun <yupeng.cun@gmail.com>

References

Yupeng Cun, Holger Frohlich (2013) netClass: An R-package for network based, integrative biomarker signature discovery.

`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

Author(s)

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

Value

R — Return a Random Walk Kernel matrix of given network, L.

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References

Kondor, R. I., & Lafferty, J. (2002, July). Diffusion kernels on graphs and other discrete input spaces. In MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE- (pp. 315-322).

See Also

See Also as `classify.stsvm`

Examples

```
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

```
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.

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

Guo et al., Towards precise classification of cancers based on robust gene functional expression profiles. BMC Bioinformatics 2005, 6:58.

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

fold	number of folds to perform
cuts	list for randomly divide the training set in to x-x-CV
x	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.
Gsub	an adjacency matrix that represents the underlying biological network.

d	damping factor for GeneRank, defaults value is 0.5
op	the uper bound of top ranked genes
aa	the lower bound of top ranked genes
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

Yupeng Cun, Holger Frohlich (2012) Integrating Prior Knowledge Into Prognostic Biomarker Discovery Based on Network Structure.arXiv:1212.3214
Winter C, Kristiansen G, Kersting S, Roy J, Aust D, et al. (2012) Google Goes Cancer: Improving Outcome Prediction for Cancer Patients by Network-Based Ranking of Marker Genes. PLoS Comput Biol 8(5): e1002511. doi:10.1371/journal.pcbi.1002511

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)
```

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 nwtwork. 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

Taylor et al.(2009)Dynamic modularity in protein interaction networks predicts breast cancer outcome, Nat. Biotech.: doi: 10.1038/nbt.1522

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

Lee E, Chuang H-Y, Kim J-W, Ideker T, Lee D (2008) Inferring Pathway Activity toward Precise Disease Classification. PLoS Comput Biol 4(11): e1000217. doi:10.1371/journal.pcbi.1000217

See Also

See Also as cv.pac

Examples

```
#see cv.pac
```

`classify.stsvm` *Training and predicting using stSVM classification methods*

Description

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 selecet 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

Yupeng Cun, Holger Frohlich (2013) Network and Data Integration for Biomarker Signature Discovery via Network Smoothed T-Statistics. PLoS ONE 8(9): e73074. doi:10.1371/journal.pone.0073074

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	paralle 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^c(-3:3)`.
seed	seed for random sampling.

Value

a LIST for Cross-Validation results

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

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References

Guo et al., Towards precise classification of cancers based on robust gene functional expression profiles. BMC Bioinformatics 2005, 6:58.

Examples

```
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 microarry classification

Usage

```
cv.frsvm(x, y, folds = 10, Gsub = matrix(1, 100, 100), repeats
                 = 5, parallel = FALSE, cores = 2, DEBUG = FALSE, d =
                 0.85, top.uper = 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 intersction network
repeats	number of CV repeat times
parallel	paralle 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.uper	the uper 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 tranined models for traning folds
feat	The feature selected by each by the fits
labels	the original lables for training

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References

Yupeng Cun, Holger Frohlich (2012) Integrating Prior Knowledge Into Prognostic Biomarker Discovery Based on Network Structure, arXiv:1212.3214
Winter C, Kristiansen G, Kersting S, Roy J, Aust D, et al. (2012) Google Goes Cancer: Improving Outcome Prediction for Cancer Patients by Network-Based Ranking of Marker Genes. PLoS Comput Biol 8(5): e1002511.

Examples

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

```
r.frsvm <-cv.frsvm(x[,1:200], y, folds=3,Gsub=ad.matrix, repeats=1, parallel=FALSE,
cores=2, DEBUG=TRUE,d=.85,top.uper=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

```
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	paralle 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 tranined models for traning folds
feat	The selected features of each training folds
labels	the original lables for training

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References

Taylor et al.(2009)Dynamic modularity in protein interaction networks predicts breast cancer outcome, Nat. Biotech.: doi: 10.1038/nbt.1522

Examples

```
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

```
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	paralle computing or not
cores	cores used in parallel computing
DEBUG	show debugging information in screen or not.
Gsub	Adjacency matrix of Protein-protein intersction network
seed	seed for random sampling.

Value

a LIST for Cross-Validation results

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

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References

Lee E, Chuang H-Y, Kim J-W, Ideker T, Lee D (2008) Inferring Pathway Activity toward Precise Disease Classification. PLoS Comput Biol 4(11): e1000217.

Examples

```
library(netClass)

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

library(KEGG.db)
r.pac <- cv.pac(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

```
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.
op.method	Method for selecet optimal feature subgoups: 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 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 tranined models for traning folds
feat	The feature selected by each by the fits
labels	the original lables for training

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. PLoS ONE 8(9): e73074.

Examples

```
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 thei 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

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

Arguments

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

Value

r	ranking of each gebes on graph

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

See Also

See Also as pGeneRank

Examples

```
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

```
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	Sacling 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 Disvovery via Network Smoothed T-Statistics

See Also

See Also as getGraphRank

Examples

```
#See also \code{classfy.stsvm}
```

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/ diffiencicial expression of genes
d	damping factor for GeneRank, defaults value is 0.5
max.degree	Max degree of graph

Value

r	ranking of each gebes on graph

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References

Morrison, Julie L., et al. "GeneRank: using search engine technology for the analysis of microarray experiments." BMC bioinformatics 6.1 (2005): 233.
Page, Lawrence, et al. "The PageRank citation ranking: bringing order to the web." (1999).

See Also

See Also as `classify.frsvm`

Examples

```
#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 tdata 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 *Predicting the test data using frsvm trained model*

Description

Predicting the test data using frsvm trained model

Usage

```
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

```
#see cv.frsvm
```

predictHubc *Predicting the test data using hubc trained model*

Description

Predicting the test data using hubc trained model

Usage

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

Arguments

train	trainied 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 *Predicting the test data using pac trained model*

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.

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

train	trained model
x	expression data for testing
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.stsvm

Examples

```
#see cv.stsvm
```

`probeset2pathway` *Generae a mean gene expression of genes of each pathway matrix*

Description

Generae 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

Guo et al., Towards precise classification of cancers based on robust gene functional expression profiles. BMC Bioinformatics 2005, 6:58.

See Also

See Also as `classify.aep`

probeset2pathwayTrain *Search CROG in training data*

Description

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

Usage

```
probeset2pathwayTrain(x = x, y = y, int = int)
```

Arguments

x	gene expression data
y	a factor of length p comprising the class labels.
int	Common genes between gene expression data and interaction network.

Value

ap	top ranked pathays
selectedGenes	CROG genes
....	

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References

Lee E, Chuang H-Y, Kim J-W, Ideker T, Lee D (2008) Inferring Pathway Activity toward Precise Disease Classification. PLoS Comput Biol 4(11): e1000217. doi:10.1371/journal.pcbi.1000217

See Also

See Also as pac.cv

Examples

```
#See Also as \name{pac.cv}
```

probeset2pathwayTst *Applied CROG to testing data*

Description

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

Usage

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

Arguments

x	gene expression data
apTrain	PAC objext which contain CORG and pathways activities lists of training data.

Value

ap	top ranked pathays

Author(s)

Yupeng Cun <yupeng.cun@gmail.com>

References

Lee E, Chuang H-Y, Kim J-W, Ideker T, Lee D (2008) Inferring Pathway Activity toward Precise Disease Classification. PLoS Comput Biol 4(11): e1000217. doi:10.1371/journal.pcbi.1000217

See Also

See Also as pac.cv, probeset2pathwayTrain

Examples

```
#See Also as \code{pac.cv, probeset2pathwayTrain}
```

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^(-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

Guo et al., Towards precise classification of cancers based on robust gene functional expression profiles. BMC Bioinformatics 2005, 6:58.

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

```
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 uper bound of top ranked genes
`aa`	the lower bound of top ranked genes
`Cs`	soft-margin tuning parameter of the SVM. Defaults to `10^c(-3:3)`.

Value

The value list returned

`train`	The tranined models for traning 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

```
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 nwtwork. Defaults to 0.98).
Cs	Soft-margin tuning parameter of the SVM. Defaults to 10^c(-3:3).

Value

The list returned

trained	The tranined models for traning 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

```
#See cv.hubc
```

`train.pac` *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

the value returned

`trained`	The tranined models for traning 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 selecet 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)`.
`EN2SY`	A list for mapping gene sybol ids or entez ids.

Value

The list returned

`trained`	The tranined models for traning 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

probeset2pathwayTst, 28

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