Package ‘pamr’

April 22, 2019

Title Pam: Prediction Analysis for Microarrays
Version 1.56.1
Author T. Hastie, R. Tibshirani, Balasubramanian Narasimhan, Gil Chu
Description Some functions for sample classification in microarrays.
Maintainer Rob Tibshirani <tibs@stanford.edu>
Depends R (>= 2.10), cluster, survival
License GPL-2
Repository CRAN
NeedsCompilation yes
Date/Publication 2019-04-22 06:58:41 UTC

R topics documented:

khan ........................................... 2
pamr.adaptthresh .................................. 2
pamr.batchadjust .................................. 3
pamr.confusion .................................... 4
pamr.confusion.survival ......................... 5
pamr.cv ........................................... 6
pamr.decorrelate .................................. 7
pamr.fdr ......................................... 9
pamr.geneplot .................................... 10
pamr.indeterminate ................................. 11
pamr.listgenes ................................... 12
pamr.makeclasses ................................. 13
pamr.menu ....................................... 14
pamr.plotcen .................................... 15
pamr.plotcv ..................................... 16
pamr.plotcvprob .................................. 17
pamr.plotfdr ..................................... 18
pamr.plotstrata .................................. 19
pamr.plotstrata .................................. 20
pamr.predict ..................................... 21
khan

Khan microarray data

Description
The khan data frame has 2309 rows and 65 columns. These are one of the datasets data used in the Tibshirani et al paper in PNAS on nearest shrunken centroids.

Details
The first row contains the sample labels. The first two columns of gene ids and names. The remaining values of the matrix are gene expression values.

pamr.adaptthresh

A function to adaptive choose threshold scales, for use in pamr.train

Description
A function to adaptive choose threshold scales, for use in pamr.train

Usage
pamr.adaptthresh(object, ntries = 10, reduction.factor = 0.9, full.out = FALSE)

Arguments

object The result of a call to pamr.train
ntries Number of iterations to use in algorithm
reduction.factor Amount by which a scaling is reduced in one step of the algorithm
full.out Should full output be returned? Default FALSE

Details
pamr.adaptthresh Adapively searches for set of good threshold scales. The baseline (default) scale is 1 for each class. The idea is that for easy to classify classes, the threshold scale can be increased without increasing the error rate for that class, and resulting in fewer genes needed for the classification rule. The scalings from pamr.adaptthresh are then used in pamr.train, and pamr.cv. The results may be better than those obtained with the default values of threshold.scale.
Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

References


Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu (2002). Class prediction by nearest shrunken centroids, with applications to DNA microarrays. Stanford tech report.

Examples

suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)
mydata <- list(x=x,y=y)
mytrain <- pamr.train(mydata)
new.scales <- pamr.adaptthresh(mytrain)

mytrain2 <- pamr.train(mydata, threshold.scale=new.scales)
myresults2 <- pamr.cv(mytrain2, mydata)

pamr.batchadjust

A function to mean-adjust microarray data by batches

Description

A function to mean-adjust microarray data by batches

Usage

pamr.batchadjust(data)

Arguments

data The input data. A list with components: x- an expression genes in the rows, samples in the columns, and y- a vector of the class labels for each sample, and batchlabels- a vector of batch labels for each sample. This object if the same form as that produced by pamr.from.excel.
Details

pamr\textunderscore batchadjust does a genewise one-way ANOVA adjustment for expression values. Let \(x(i,j)\) be the expression for gene \(i\) in sample \(j\). Suppose sample \(j\) is in batch \(b\), and let \(B\) be the set of all samples in batch \(b\). Then \texttt{pamr.batchadjust}\ adjusts \(x(i,j)\) to \(x(i,j) - \text{mean}[x(i,j)]\) where the mean is taken over all samples \(j\) in \(B\).

Value

A data object of the same form as the input data, with \(x\) replaced by the adjusted \(x\).

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

```r
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
# generate some data
x <- matrix(rnorm(1000*20),nrow=20)
y <- sample(c(1:10),size=20,replace=TRUE)
batchlabels <- sample(c(1:5),size=20,replace=TRUE)
mydata <- list(x=x,y=factor(y),batchlabels=factor(batchlabels))

mydata2 <- pamr.batchadjust(mydata)
```

\texttt{pamr.confusion} \hspace{1cm} \textit{A function giving a table of true versus predicted values, from a nearest shrunken centroid fit.}

Description

A function giving a table of true versus predicted values, from a nearest shrunken centroid fit.

Usage

\texttt{pamr.confusion(fit, threshold, extra=TRUE)}

Arguments

- \texttt{fit} \hspace{1cm} The result of a call to \texttt{pamr.train} or \texttt{pamr.cv}
- \texttt{threshold} \hspace{1cm} The desired threshold value
- \texttt{extra} \hspace{1cm} Should the classwise and overall error rates be returned? Default: \texttt{TRUE}

Details

\texttt{pamr.confusion}\ gives a cross-tabulation of true versus predicted classes for the fit returned by \texttt{pamr.train} or \texttt{pamr.cv}, at the specified threshold.
**pamr.confusion.survival**

**Author(s)**

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

**Examples**

```r
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(100*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)
mydata <- list(x=x,y=y)
mytrain <- pamr.train(mydata)
mycv <- pamr.cv(mytrain,mydata)
pamr.confusion(mytrain, threshold=2)
pamr.confusion(mycv, threshold=2)
```

**pamr.confusion.survival**

*Compute confusion matrix from pamr survival fit*

**Description**

computes confusion matrix for (survival.time, censoring) outcome based on fit object "fit" and class predictions "yhat" soft response probabilities for (survival.time, censoring) are first estimated using Kaplan-Meier method applied to training data

**Usage**

```r
pamr.confusion.survival(fit, survival.time, censoring.status, yhat)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>The result of a call to pamr.train or pamr.cv</td>
</tr>
<tr>
<td>survival.time</td>
<td>Survival time</td>
</tr>
<tr>
<td>censoring.status</td>
<td>censoring status</td>
</tr>
<tr>
<td>yhat</td>
<td>class predictions</td>
</tr>
</tbody>
</table>

**Author(s)**

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu
pamr.cv

A function to cross-validate the nearest shrunken centroid classifier

Description

A function to cross-validate the nearest shrunken centroid classifier produced by pamr.train

Usage

pamr.cv(fit, data, nfold = NULL, folds = NULL, ...)

Arguments

fit The result of a call to pamr.train
data A list with at least two components: x- an expression genes in the rows, samples in the columns), and y- a vector of the class labels for each sample. Same form as data object used by pamr.train.
nfold Number of cross-validation folds. Default is the smallest class size
folds A list with nfold components, each component a vector of indices of the samples in that fold. By default a (random) balanced cross-validation is used.
... Any additional arguments that are to be passed to pamr.train

Details

pamr.cv carries out cross-validation for a nearest shrunken centroid classifier.

Value

A list with components

threshold A vector of the thresholds tried in the shrinkage
errors The number of cross-validation errors for each threshold value
loglik The cross-validated multinomial log-likelihood value for each threshold value
size A vector of the number of genes that survived the thresholding, for each threshold value tried.

yhat A matrix of size n by nthreshold, containing the cross-validated class predictions for each threshold value, in each column
prob A matrix of size n by nthreshold, containing the cross-validated class probabilities for each threshold value, in each column
folds The cross-validation folds used
cv.objects Train objects (output of pamr.train), from each of the CV folds
call The calling sequence used
Author(s)
Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20), ncol=20)
y <- sample(c(1:4), size=20, replace=TRUE)
mydata <- list(x=x, y= factor(y), geneid=as.character(1:nrow(x)),
genenames= paste("g", as.character(1:nrow(x)), sep=""))
mytrain <- pamr.train(mydata)
mycv <- pamr.cv(mytrain, mydata)

pamr.decorrelate
A function to decorrelate (adjust) the feature matrix with respect to some additional predictors

Description
A function to decorrelate (adjust) the feature matrix with respect to some additional predictors

Usage
pamr.decorrelate(x, adjusting.predictors, xtest=NULL, adjusting.predictors.test=NULL)

Arguments
x Matrix of training set feature values, with genes in the rows, samples in the columns
adjusting.predictors List of training set predictors to be used for adjustment
xtest Optional matrix of test set feature values, to be adjusted in the same way as the training set
adjusting.predictors.test Optional list of test set predictors to be used for adjustment

Details
pamr.decorrelate Does a least squares regression of each row of x on the adjusting predictors, and returns the residuals. If xtest is provided, it also returns the adjusted version of xtest, using the training set least squares regression model for adjustment
Value

A list with components

- `x.adj`: Adjusted x matrix
- `xtest.adj`: Adjusted xtest matrix, if xtest we provided

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

References

Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu Diagnosis of multiple cancer types by shrunken centroids of gene expression PNAS 99: 6567-6572. Available at www.pnas.org

Examples

```r
# generate some data
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)

x <- matrix(rnorm(10000), ncol=20)
y <- c(rep(1, 10), rep(2, 10))
adjusting.predictors <- list(pred1 = rnorm(20),
    pred2 = as.factor(sample(c(1, 2), replace = TRUE, size=20)))
xtest <- matrix(rnorm(10000), ncol=10)
adjusting.predictors.test <- list(pred1 = rnorm(10),
    pred2 = as.factor(sample(c(1, 2), replace = TRUE, size=10)))

# decorrelate training x wrt adjusting predictors
x.adj <- pamr.decorrelate(x, adjusting.predictors)$x.adj

# train classifier with adjusted x

d <- list(x = x.adj, y = y)
a <- pamr.train(d)

# decorrelate training and test x wrt adjusting predictors, then make predictions for test set

temp <- pamr.decorrelate(x, adjusting.predictors, xtest = xtest,
    adjusting.predictors.test = adjusting.predictors.test)

d <- list(x = temp$x.adj, y = y)
a <- pamr.train(d)
aa <- pamr.predict(a, temp$xtest.adj, threshold=.5)
```
pamr.fdr

A function to estimate false discovery rates for the nearest shrunken centroid classifier

Description

A function to estimate false discovery rates for the nearest shrunken centroid classifier

Usage

pamr.fdr(trained.obj, data, nperms=100, 
xl.mode=c("regular","firsttime","onetime","lasttime"),xl.time=NULL, xl.prevfit=NULL)

Arguments

- **trained.obj**: The result of a call to pamr.train
- **data**: Data object; same as the one passed to pamr.train
- **nperms**: Number of permutations for estimation of FDRs. Default is 100
- **xl.mode**: Used by Excel interface
- **xl.time**: Used by Excel interface
- **xl.prevfit**: Used by Excel interface

Details

pamr.fdr estimates false discovery rates for a nearest shrunken centroid classifier

Value

A list with components:

- **results**: Matrix of estimates FDRs for various threshold values. Reported are both the median and 90th percentile of the FDR over permutations
- **pi0**: The estimated proportion of genes that are null, i.e. not significantly different

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu
Examples

```r
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)

mydata <- list(x=x,y=factor(y), geneid=as.character(1:nrow(x)),
genenames=paste("g",as.character(1:nrow(x)),sep=""))

mytrain <- pamr.train(mydata)
myfdr <- pamr.fdr(mytrain, mydata)
```

pamr.geneplot

A function to plot the genes that survive the thresholding from the nearest shrunken centroid classifier

Description

A function to plot the genes that survive the thresholding, from the nearest shrunken centroid classifier produced by pamr.train

Usage

```r
pamr.geneplot(fit, data, threshold)
```

Arguments

- `fit` The result of a call to pamr.train
- `data` The input data. In the same format as the input data for pamr.train
- `threshold` The desired threshold value

Details

pamr.geneplot Plots the raw gene expression for genes that survive the specified threshold. Plot is stratified by class. Plot is set up to display only up to about 20 or 25 genes, otherwise it gets too crowded. Hence threshold should be chosen to yield at most about 20 or 25 genes.

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu
pamr.indeterminate

Examples

```r
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)
mydata <- list(x=x,y=y)
mytrain <- pamr.train(mydata)
pamr.geneplot(mytrain, mydata, threshold=1.6)
```

pamr.indeterminate A function that takes estimate class probabilities and produces a class prediction or indeterminate prediction

Description

A function that takes estimate class probabilities and produces a class prediction or indeterminate prediction

Usage

```r
pamr.indeterminate(prob, mingap=0)
```

Arguments

- `prob`: Estimated class probabilities, from pamr.predict with type="posterior"
- `mingap`: Minimum difference between highest and second highest probability. If difference is < mingap, prediction is set to indeterminate (NA)

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

```r
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)
mydata <- list(x=x,y=y)
mytrain <- pamr.train(mydata)
prob<- pamr.predict(mytrain, mydata$x , threshold=1, type="posterior")
pamr.indeterminate(prob,mingap=.75)
```
pamr.listgenes  A function to list the genes that survive the thresholding, from the nearest shrunken centroid classifier

Description

A function to list the genes that survive the thresholding, from the nearest shrunken centroid classifier produced by pamr.train

Usage

pamr.listgenes(fit, data, threshold, fitcv=NULL, genenames=FALSE)

Arguments

fit The result of a call to pamr.train
data The input data. In the same format as the input data for pamr.train
threshold The desired threshold value
fitcv Optional object, result of a call to pamr.cv
genenames Include genenames in the list? If yes, they are taken from "data". Default is false (geneid is always included in the list).

Details

pamr.listgenes List the geneids, and standardized centroids for each class, for genes surviving at the given threshold. If fitcv is provided, the function also reports the average rank of the gene in the cross-validation folds, and the proportion of times that the gene is chosen (at the given threshold) in the cross-validation folds.

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

#generate some data
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20), ncol=20)
y <- sample(c(1:4), size=20, replace=TRUE)
mydata <- list(x=x, y=factor(y), geneid=as.character(1:nrow(x)),
genenames=paste("g", as.character(1:nrow(x)), sep=""))

#train classifier
mytrain<- pamr.train(mydata)

pamr.listgenes(mytrain, mydata, threshold=1.6)

---

pamr.makeclasses  
*A function to interactively define classes from a clustering tree*

**Description**

function to interactively define classes from a clustering tree

**Usage**

pamr.makeclasses(data, sort.by.class=FALSE,...)

**Arguments**

- **data**  
The input data. A list with components: x- an expression genes in the rows, samples in the columns, and y- a vector of the class labels for each sample, and batchlabels- a vector of batch labels for each sample. This object is the same form as that produced by pamr.from.excel.

- **sort.by.class**  
Optional argument. If true, the clustering tree is forced to put all samples in the same class (as defined by the class labels y in ‘data’) together in the tree. This is useful if a regrouping of classes is desired. Eg: given classes 1,2,3,4 you want to define new classes (1,3) vs (2,4) or 2 vs (1,3)

- **...**  
Any additional arguments to be passed to hclust

**Details**

pamr.makeclasses  
Using this function the user interactively defines a new set of classes, to be used in pamr.train, pamr.cv etc. After invoking pamr.makeclasses, a clustering tree is drawn. This calls the R function hclust, and any arguments for hclust can be passed to it. Using the left button, the user clicks at the junction point defining the subgroup 1. More groups can be added to class 1 by clicking on further junction points. The user ends the definition of class 1 by clicking on the rightmost button [in Windows, an additional menu appears and he chooses Stop]. This process is continued for classes 2,3 etc. Note that some sample may be left out of the new classes. Two consecutive clicks of the right button ends the definition for all classes.

At the end, the clustering is redrawn, with the new class labels shown.

Note: this function is "fragile". The user must click close to the junction point, to avoid confusion with other junction points. Classes 1,2,3.. cannot have samples in common (if they do, an Error message will appear). If the function is confused about the desired choices, it will complain and ask the user to rerun pamr.makeclasses. The user should also check that the labels on the final redrawn cluster tree agrees with the desired classes.
pamr.menu

A function that interactively leads the user through a PAM analysis

Description

A function that interactively leads the user through a PAM analysis

Usage

pamr.menu(data)

Arguments

data A list with at least two components: x- an expression genes in the rows, samples in the columns), and y- a vector of the class labels for each sample. Same form as data object used by pamr.train.

Details

pamr.menu provides a menu for training, cross-validating and plotting a nearest shrunken centroid analysis.
Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20), ncol=20)
y <- sample(c(1:4), size=20, replace=TRUE)
mydata <- list(x=x, y=y)
# pamr.menu(mydata)

Description

A function to plot the shrunken class centroids, from the nearest shrunken centroid classifier produced by pamr.train

Usage

pamr.plotcen(fit, data, threshold)

Arguments

fit The result of a call to pamr.train
data The input data, in the same form as that used by pamr.train,
threshold The desired threshold value

Details

pamr.plotcen plots the shrunken class centroids for each class, for genes surviving the threshold for at least once class. If genenames are included in "data", they are added to the plot. Note: for many classes and long gene names, this plot may need some manual prettying.

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu
Examples

```r
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)
mydata <- list(x=x,y=y,genenames=as.character(1:1000))
mytrain <- pamr.train(mydata)
mycv <- pamr.cv(mytrain,mydata)
pamr.plotcen(mytrain, mydata,threshold=1.6)
```

---

table

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pamr.plotcv</code></td>
<td>A function to plot the cross-validated error curves from the nearest shrunken centroid classifier</td>
</tr>
</tbody>
</table>

Description

A function to plot the cross-validated error curves the nearest shrunken centroid classifier

Usage

```r
pamr.plotcv(fit)
```

Arguments

- `fit` The result of a call to `pamr.cv`

Details

`pamr.plotcv` plots the cross-validated misclassification error curves, from nearest shrunken centroid classifier. An overall plot, and a plot by class, are produced.

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

```r
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)
mydata <- list(x=x,y=y)
mytrain <- pamr.train(mydata)
mycv <- pamr.cv(mytrain,mydata)
pamr.plotcv(mycv)
```
pamr.plotcvprob

pamr.plotcvprob  A function to plot the cross-validated sample probabilities from the nearest shrunken centroid classifier

Description

A function to plot the cross-validated sample probabilities from the nearest shrunken centroid classifier

Usage

pamr.plotcvprob(fit, data, threshold)

Arguments

fit       The result of a call to pamr.cv
data      A list with at least two components: x - an expression genes in the rows, samples in the columns), and y - a vector of the class labels for each sample. Same form as data object used by pamr.train.
threshold Threshold value to be used

Details

pamr.plotcvprob plots the cross-validated sample probabilities the from nearest shrunken centroid classifier, stratified by the true classes.

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20), ncol=20)
y <- sample(c(1:4), size=20, replace=TRUE)
mydata <- list(x=x, y=y)
mytrain <- pamr.train(mydata)
mycv <- pamr.cv(mytrain, mydata)
pamr.plotcvprob(mycv, mydata, threshold=1.6)
pamr.plotfdr

A function to plot the FDR curve from the nearest shrunken centroid classifier

Description

A function to plot the FDR curve the nearest shrunken centroid classifier

Usage

pamr.plotfdr(fdrfit, call.win.metafile = FALSE)

Arguments

fdrfit The result of a call to pamr.fdr
call.win.metafile Used by Excel interface

Details

pamr.plotfdr plots the FDR curves from nearest shrunken centroid classifier. The median FDR (solid line) and upper 90 percentile (broken line) are shown

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000x20), ncol=20)
y <- sample(c(1:2), size=20, replace=TRUE)
x[,1:50,y==2]=x[,1:50,y==2]+3
mydata <- list(x=x, y=y)
mytrain <- pamr.train(mydata)
myfdr <- pamr.fdr(mytrain, mydata)
pamr.plotfdr(myfdr)
pamr.plotstrata

A function to plot the survival curves in each Kaplan Meier stratum

Description

A function to plot the survival curves in each Kaplan Meier stratum

Usage

pamr.plotstrata(fit, survival.time, censoring.status)

Arguments

fit
The result of a call to pamr.train

survival.time
Vector of survival times

censoring.status
Vector of censoring status values

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

gendata<-function(n=100, p=2000){
  tim <- 3*abs(rnorm(n))
  u<-runif(n, min(tim), max(tim))
  y<-pmin(tim,u)
  ic<-1*(tim<u)
  m <- median(tim)
  x<-matrix(rnorm(p*n),ncol=n)
  x[1:100, tim>m] <- x[1:100, tim>m]+3
  return(list(x=x, y=y, ic=ic))
}

# generate training data; 2000 genes, 100 samples
junk<-gendata(n=100)
y<-junk$y
ic<-junk$ic
x<-junk$x
d <- list(x=x, survival.time=y, censoring.status=ic,
  geneid=as.character(1:nrow(x)), genenames=paste("g",as.character(1:nrow(x)), sep=" "))

# train model
a3<- pamr.train(d, ngroup.survival=2)
pamr.plotstrata(a3, d$survival.time, d$censoring.status)

Description

A function to plots Kaplan-Meier curves stratified by a group variable

Usage

pamr.plotstrata(group, survival.time, censoring.status)

Arguments

group A grouping factor

survival.time Vector of survival times
censoring.status Vector of censoring status values: 1=died, 0=censored

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

gendata<-function(n=100, p=2000){
  tim <- 3*abs(rnorm(n))
  u<-runif(n,min(tim),max(tim))
  y<-pmin(tim,u)
  ic<1*(tim<u)
  m <- median(tim)
  x<-matrix(rnorm(p*n),nrow=n)
  x[1:100, tim>m] <- x[1:100, tim>m]+3
  return(list(x=x,y=y,ic=ic))
}

# generate training data; 2000 genes, 100 samples
junk<-gendata(n=100)
y<-junk$y
ic<-junk$ic
x<-junk$x
d <- list(x=x, survival.time=y, censoring.status=ic,
genenameas.character(1:nrow(x)), genenames=paste("g",as.character(1:nrow(x)), sep=""))
# train model
`a3 <- pamr.train(d, ngroup.survival=2)`

# make class predictions
`yhat <- pamr.predict(a3, d$x, threshold=1.0)`

`pamr.plotSurvival(yhat, d$survival.time, d$censoring.status)`

---

**pamr.predict**

*Function giving prediction information, from a nearest shrunken centroid fit.*

### Description

A function giving prediction information, from a nearest shrunken centroid fit.

### Usage

```r
pamr.predict(fit, newx, threshold, 
  type = c("class", "posterior", "centroid", "nonzero"), 
  prior = fit$prior, threshold.scale = fit$threshold.scale)
```

### Arguments

- `fit` : The result of a call to `pamr.train`
- `newx` : Matrix of features at which predictions are to be made
- `threshold` : The desired threshold value
- `type` : Type of prediction desired: class predictions, posterior probabilities, (unshrunken) class centroids, vector of genes surviving the threshold
- `prior` : Prior probabilities for each class. Default is that specified in "fit"
- `threshold.scale` : Additional scaling factors to be applied to the thresholds. Vector of length equal to the number of classes. Default is that specified in "fit".

### Details

`pamr.predict` Give a cross-tabulation of true versus predicted classes for the fit returned by `pamr.train` or `pamr.cv`, at the specified threshold.

### Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu
pamr.predictmany

Examples

suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)
mydata <- list(x=x,y=y)
mytrain <- pamr.train(mydata)
mycv <- pamr.cv(mytrain,mydata)
pamr.predict(mytrain, mydata$x , threshold=1)

pamr.predictmany

A function giving prediction information for many threshold values, from a nearest shrunken centroid fit.

Description

A function giving prediction information for many threshold values, from a nearest shrunken centroid fit

Usage

pamr.predictmany(fit, newx, threshold = fit$threshold, prior = fit$prior,
threshold.scale = fit$threshold.scale, ...)

Arguments

fit The result of a call to pamr.train
newx Matrix of features at which predictions are to be made
threshold The desired threshold values
prior Prior probabilities for each class. Default is that specified in "fit"
threshold.scale Additional scaling factors to be applied to the thresholds. Vector of length equal to the number of classes. Default is that specified in "fit".
... Additional arguments to be passed to pamr.predict

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu
Examples

```r
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20), ncol=20)
y <- sample(c(1:4), size=20, replace=TRUE)
mydata <- list(x=x, y=y)
mytrain <- pamr.train(mydata)
pamr.predictmany(mytrain, mydata$x)
```

**pamr.surv.to.class2**  
A function to assign observations to categories, based on their survival times.

**Description**

A function to assign observations to categories, based on their survival times.

**Usage**

```r
pamr.surv.to.class2(y, icens, cutoffs=NULL, n.class=NULL, class.names=NULL, 
newy=y, newic=icens)
```

**Arguments**

- `y` vector of survival times
- `icens` Vector of censoring status values: 1=died, 0=censored
- `cutoffs` Survival time cutoffs for categories. Default NULL
- `n.class` Number of classes to create: if cutoffs is NULL, n.class equal classes are created.
- `class.names` Character names for classes
- `newy` New set of survival times, for which probabilities are computed (see below). Default is y
- `newic` New set of censoring statuses, for which probabilities are computed (see below). Default is icens

**Details**

`pamr.surv.to.class2` splits observations into categories based on their survival times and the Kaplan-Meier estimates. For example if `n.class`=2, it makes two categories, one below the median survival, the other above. For each observation (newy, ic), it then computes the probability of that observation falling in each category. For an uncensored observation that probability is just 1 or 0 depending on when the death occurred. For a censored observation, the probabilities are based on the Kaplan Meier and are typically between 0 and 1.
Value

class The category labels
prob The estimates class probabilities
cutoffs The cutoffs used

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

gendata<-function(n=100, p=2000){
  tim <- 3*abs(rnorm(n))
  u<-runif(n,min(tim),max(tim))
  y<-pmin(tim,u)
  ic<-1*(tim<u)
  m <- median(tim)
  x<-matrix(rnorm(p*n),ncol=n)
  x[1:100, tim>m] <- x[1:100, tim>m]+3
  return(list(x=x,y=y,ic=ic))
}
# generate training data; 2000 genes, 100 samples
junk<-gendata(n=100)
y<-junk$y
ic<-junk$ic
x<-junk$x
d <- list(x=x, survival.time=y, censoring.status=ic,
geneid=as.character(1:nrow(x)), genenames=paste("g", as.character(1:nrow(x)), sep= ""))
# train model
aS<- pamr.train(d, ngroup=2)
# generate test data
junkk<- gendata(n=500)

dd <- list(x=junkk$x, survival.time=junkk$y, censoring.status=junkk$ic)
# compute soft labels
proby <- pamr.surv.to.class2(dd$survival.time, dd$censoring.status,
n.class=aS$ngroup.survival)$prob
pamr.test.errors.surv.compute

A function giving a table of true versus predicted values, from a nearest shrunken centroid fit from survival data.

Description

A function giving a table of true versus predicted values, from a nearest shrunken centroid fit from survival data.

Usage

pamr.test.errors.surv.compute(proby, yhat)

Arguments

proby
Survival class probabilities, from pamr.surv.to.class2

yhat
Estimated class labels, from pamr.predict

Details

pamr.test.errors.surv.compute computes the errors between the true 'soft' class labels proby and the estimated ones "yhat"

Author(s)

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

Examples

gendata<-function(n=100, p=2000){
  tim <- 3*abs(rnorm(n))
  u<-runif(n,min(tim),max(tim))
  y<-pmin(tim,u)
  ic<-1*(tim<u)
  m <- median(tim)
  x<-matrix(rnorm(p*n),ncol=n)
  x[1:100, tim>m] <- x[1:100, tim>m]+3
  return(list(x=x,y=y,ic=ic))
}

# generate training data; 2000 genes, 100 samples
junk<-gendata(n=100)
y<-junk$y
ic<-junk$ic
x<-junk$x
d <- list(x=x,survival.time=y, censoring.status=ic,
pamr.train

A function to train a nearest shrunken centroid classifier

Description

A function that computes a nearest shrunken centroid for gene expression (microarray) data

Usage

pamr.train(data, gene.subset=NULL, sample.subset=NULL, 
threshold = NULL, n.threshold = 30, 
scale.sd = TRUE, threshold.scale = NULL, se.scale = NULL, offset.percent = 50, 
hetero=NULL, prior = NULL, remove.zeros = TRUE, sign.contrast="both", 
ngroup.survival = 2)

Arguments

data The input data. A list with components: x- an expression genes in the rows, samples in the columns), and y- a vector of the class labels for each sample. Optional components- genenames, a vector of gene names, and geneid- a vector of gene identifiers.

gene.subset Subset of genes to be used. Can be either a logical vector of length total number of genes, or a list of integers of the row numbers of the genes to be used
**pamr.train**

- **sample.subset**: Subset of samples to be used. Can be either a logical vector of length total number of samples, or a list of integers of the column numbers of the samples to be used.

- **threshold**: A vector of threshold values for the centroid shrinkage. Default is a set of 30 values chosen by the software.

- **n.threshold**: Number of threshold values desired (default 30).

- **scale.sd**: Scale each threshold by the within class standard deviations? Default: true.

- **threshold.scale**: Additional scaling factors to be applied to the thresholds. Vector of length equal to the number of classes. Default: a vector of ones.

- **se.scale**: Vector of scaling factors for the within class standard errors. Default is \(\sqrt{\frac{1}{n.\text{class}} - \frac{1}{n}}\), where \(n\) is the overall sample size and \(n.\text{class}\) is the sample sizes in each class. This default adjusts for different class sizes.

- **offset.percent**: Fudge factor added to the denominator of each t-statistic, expressed as a percentile of the gene standard deviation values. This is a small positive quantity to penalize genes with expression values near zero, which can result in very large ratios. This factor is especially important for Affy data. Default is the median of the standard deviations of each gene.

- **hetero**: Should a heterogeneity transformation be done? If yes, hetero must be set to one of the class labels (see Details below). Default is no (hetero=NULL).

- **prior**: Vector of length the number of classes, representing prior probabilities for each of the classes. The prior is used in Bayes rule for making class prediction. Default is NULL, and prior is then taken to be \(n.\text{class}/n\), where \(n\) is the overall sample size and \(n.\text{class}\) is the sample sizes in each class.

- **remove.zeros**: Remove threshold values yielding zero genes? Default TRUE.

- **sign.contrast**: Directions of allowed deviations of class-wise average gene expression from the overall average gene expression. Default is “both” (positive or negative). Can also be set to “positive” or “negative”.

- **ngroup.survival**: Number of groups formed for survival data. Default 2

**Details**

**pamr.train** fits a nearest shrunken centroid classifier to gene expression data. Details may be found in the PNAS paper referenced below. One feature not described there is "heterogeneity analysis". Suppose there are two classes labelled "A" and "B". Class "A" is considered a normal class, and "B" an abnormal class. Setting hetero="A" transforms expression values \(x[i,j]\) to \(|x[i,j] - \text{mean}(x[i,j])|\) where the mean is taken only over samples in class "A". The transformed feature values are then used in Pam. This is useful when the abnormal class "B" is heterogeneous, i.e. a given gene might have higher expression than normal for some class "B" samples, and lower for others. With more than 2 classes, each class is centered on the class specified by hetero.

**Value**

A list with components
The outcome classes.

A matrix of predicted classes, each column representing the results from one threshold.

A array of predicted class probabilities. of dimension n by nclass by n.threshold. n is the number samples, nclass is the number of classes, n.threshold is the number of thresholds tried

A matrix of (unshrunk) class centroids, n by nclass

Value of hetero used in call to pamr.train

Centroid of "normal" group, if hetero was specified

A vector containing the (unshrunk) overall centroid (all classes together)

A vector of the standard deviations for each gene

A vector of the threshold tried in the shrinkage

A vector of the number of genes that survived the thresholding, for each threshold value tried

A vector of threshold scale factors that were used

A vector of standard error scale factors that were used

The calling sequence used

The prior probabilities used

The number of trainin errors for each threshold value

**Author(s)**

Trevor Hastie, Robert Tibshirani, Balasubramanian Narasimhan, and Gilbert Chu

**References**

Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu Diagnosis of multiple cancer types by shrunkcenroids of gene expression PNAS 99: 6567-6572. Available at www.pnas.org

**Examples**

#generate some data
suppressWarnings(RNGversion("3.5.0"))
set.seed(120)
x <- matrix(rnorm(1000*20),ncol=20)
y <- sample(c(1:4),size=20,replace=TRUE)
mydata <- list(x=x,y=factor(y))

#train classifier
results<- pamr.train(mydata)
# train classifier on all data except class 4
results2 <- pamr.train(mydata,sample.subset=(mydata$y!=4))

# train classifier on only the first 500 genes
results3 <- pamr.train(mydata,gene.subset=1:500)
## Index

*Topic **datasets**

- khan, 2

khan, 2

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>pamr.adaptthresh</td>
<td>2</td>
</tr>
<tr>
<td>pamr.batchadjust</td>
<td>3</td>
</tr>
<tr>
<td>pamr.confusion</td>
<td>4</td>
</tr>
<tr>
<td>pamr.confusion.survival</td>
<td>5</td>
</tr>
<tr>
<td>pamr.cv</td>
<td>6</td>
</tr>
<tr>
<td>pamr.decorrelate</td>
<td>7</td>
</tr>
<tr>
<td>pamr.fdr</td>
<td>9</td>
</tr>
<tr>
<td>pamr.geneplot</td>
<td>10</td>
</tr>
<tr>
<td>pamr.indeterminate</td>
<td>11</td>
</tr>
<tr>
<td>pamr.listgenes</td>
<td>12</td>
</tr>
<tr>
<td>pamr.makeclasses</td>
<td>13</td>
</tr>
<tr>
<td>pamr.menu</td>
<td>14</td>
</tr>
<tr>
<td>pamr.plotcencen</td>
<td>15</td>
</tr>
<tr>
<td>pamr.plotcv</td>
<td>16</td>
</tr>
<tr>
<td>pamr.plotcvprob</td>
<td>17</td>
</tr>
<tr>
<td>pamr.plotfdr</td>
<td>18</td>
</tr>
<tr>
<td>pamr.plotstrata</td>
<td>19</td>
</tr>
<tr>
<td>pamr.plotssurvival</td>
<td>20</td>
</tr>
<tr>
<td>pamr.predict</td>
<td>21</td>
</tr>
<tr>
<td>pamr.predictmany</td>
<td>22</td>
</tr>
<tr>
<td>pamr.surv.to.class2</td>
<td>23</td>
</tr>
<tr>
<td>pamr.test.errors.surv.compute</td>
<td>25</td>
</tr>
<tr>
<td>pamr.train</td>
<td>26</td>
</tr>
</tbody>
</table>