Package 'rocc'

May 9, 2026

Title ROC Based Classification

Version 1.3

Depends ROCR

Imports methods

Author Martin Lauss

Description Functions for a classification method based on receiver operating characteristics (ROC). Briefly, features are selected according to their ranked AUC value in the training set. The selected features are merged by the mean value to form a meta-gene. The samples are ranked by their meta-gene value and the meta-gene threshold that has the highest accuracy in splitting the training samples is determined. A new sample is classified by its meta-gene value relative to the threshold. In the first place, the package is aimed at two class problems in gene expression data, but might also apply to other problems.

Maintainer Martin Lauss <martin.lauss@med.lu.se>

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-12-06 15:00:02 UTC

Contents

o.rocc *LOOCV using the ROC based classifier*

Description

The function performs classification by leave-one-out-cross-validation (LOOCV) using the ROC based classifier: Features are combined to a metagene by the mean expression and samples are ranked according to the metagene expression. The metagene threshold that yields optimal accuracy in the training samples is then used to classify new samples according to their metagene expression values.

Usage

```
o.rocc(g, out, xgenes = 200)
```

Arguments

g	the input data in form of a matrix with genes as rows and samples as columns. rownames(g) and colnames (g) must be specified.
out	describes the phenotype of the samples. a factor vector with levels 0 and 1 (in this order) with as many values as there are samples.
xgenes	number of genes in the classifier. numeric vector with length of at least 1.

Details

For feature selection the genes are ranked by AUC values and the top ranking AUC genes according to xgenes are picked (all AUCs below 0.5 are mirrored). To obtain AUC values for a given gene signature an arithmetic mean is computed by summing up the expression values after multiplying expression values for genes negatively associated with the feature (AUC below 0.5) by -1. The resulting expression values for the thus formed metagenes are then used in ROC analysis. The optimal split of positive (i.e., 1) and negative (i.e., 0) samples is determined as the metagene expression threshold which produces the highest accuracy, correct class assignments in respect to the real class, in the training set. The split yielding optimal accuracy in the ROC curve is determined using the package ROCR. The threshold is computed as the mean metagene expression value of the two samples at the boarder of the split. A new sample to be classified has its metagene expression value determined with the same genes to be multiplied by -1. The new sample is classified according to which side of the threshold the sample falls in, with a sample having higher metagene expression being classified as positive (i.e., 1) and with lower expression as negative (i.e., 0). Performance of the classifier is estimated in the dataset by leave-one-out cross validation with feature selection and classifier specification repeated in each loop to ensure that the remaining sample has not seen the classifier.

Value

a list (orocc object) with components

confusion	a matrix containing the classifier performance in leave-one-out cross validation for each gene size determined by xgenes. The following measures are returned: accuracy, lower and upper 95 percent confidence interval, the largest prior class (accuracy null), the p value from the binomial test that the accuracy is different from accuracy null, accuracy obtained by random assignment (using prior distributions), the p value from the binomial test that the accuracy is different from random assignment, sensitivity, specificity, positive predictive value, negative predictive value, prevalence, contingency table values of predicted versus true class, and the balanced accuracy that is calculated as (sensitivity+specificity)/2.
concordance	a matrix that contains the predicted classification obtained by leave-one-out cross validation. Additionally the true classes (out) are shown.
method	the classification method used: ROC.based.predictor.

Note

depends on the package ROCR

Author(s)

Martin Lauss

References

Lauss M, Frigyesi A, Ryden T, Hoglund M. Robust assignment of cancer subtypes from expression data using a uni-variate gene expression average as classifier. BMC Cancer 2010 (in print)

See Also

tr.rocc(), p.rocc()

Examples

```
## Random dataset and phenotype (small dataset for demonstration)
## Dataset should be a matrix
set.seed(100)
g <- matrix(rnorm(1000*25),ncol=25)
rownames(g) <- paste("Gene",1:1000,sep="_")
colnames(g) <- paste("Sample",1:25,sep="_")
## Phenotype should be a factor with levels 0 and 1:
out <- as.factor(sample(c(0:1),size=25,replace=TRUE))
## Set the size of the Gene Signature
xgenes=c(50,500)

####### o.rocc

results<-o.rocc (g,out,xgenes)
results



####### performance of a given gene set by LOOCV in independent data
```

```
## load given genes (or take $genes from tr.rocc output)
genes<-paste("Gene",1:50,sep="_")
## load validation data
set.seed(101)
f <- matrix(rnorm(1000*25),ncol=25)
rownames(f) <- paste("Gene",1:1000,sep="_")
colnames(f) <- paste("Sample",1:25,sep="_")
outf <- as.factor(sample(c(0:1),size=25,replace=TRUE))
## reduce validation set to gene signature genes
f<-f[genes,]
## use all genes of reduced dataset for LOOCV
xgenes<-length(genes)

resultval<-o.rocc (f,outf,xgenes)
resultval


######### o.rocc results can be redone as a LOOCV with tr.rocc und p.rocc functions
#
#results$concordance[,"50"]
#
### now with a LOOCV loop of tr.rocc and p.rocc
#pr<-as.numeric(rep(NA,length(colnames(g))))
#pr<-factor(pr,level=c(0,1))
#
#for (i in 1:length(colnames(g))){
#e<-g[,-i]
#oute<-out[-i]
#train<-tr.rocc(e,oute,xgenes=50)
#procc<-p.rocc(train,g[,i]) ## ignore warnings, they dont apply here
#pr[i]<-procc
#}
#
#all.equal(results$concordance[,"50"],pr)
## TRUE
```

---

p.rocc *Making predictions using the ROC based classifier*

Description

Class predictions of new samples using a ROC based classifier obtained by tr.rocc()

Usage

```
p.rocc(trocc, newsample)
```

Arguments

trocc	a ROC based classifier (containing the classifier specifications). This object is generated in training data using tr.rocc()
newsample	a matrix containing the new samples, with genes as rows and samples as columns. rownames(g) and colnames (g) must be specified. All features of the classifier (trocc$genes) have to be present in the rownames of the matrix.

Details

The classifier specifications of the trocc object from classifier training are used to classify new samples. The metagene value of the new sample is calculated using the information from trocc$positiv and trocc$negativ. If the metagene value is higher than the threshold value (obtained from trocc$cutoffvalue) the new sample is predicted to be of class 1, else to be of class 0.

Value

a named factor vector with levels 0 and 1 containing the predictions.

Note

p.rocc() requires a trocc object generated by the tr.rocc() function

Author(s)

Martin Lauss

References

Lauss M, Frigyesi A, Ryden T, Hoglund M. Robust assignment of cancer subtypes from expression data using a uni-variate gene expression average as classifier. BMC Cancer 2010 (in print)

See Also

tr.rocc, o.rocc

Examples

```
#### tr.rocc

### Random Dataset and phenotype
set.seed(100)
## Dataset should be a matrix
g <- matrix(rnorm(1000*25),ncol=25)
rownames(g) <- paste("Gene",1:1000,sep="_")
colnames(g) <- paste("Sample",1:25,sep="_")
## Phenotype should be a factor with levels 0 and 1:
out <- as.factor(sample(c(0:1),size=25,replace=TRUE))

predictor <- tr.rocc (g,out,xgenes=50)
```

```
## find classifier specification:
predictor$positiv
predictor$negativ
predictor$cutoffvalue




#### p.rocc

### just an example: classification of the training samples
p.rocc(trocc=predictor,newsample=g)
predictions<-p.rocc(trocc=predictor,newsample=g)
table(predictions,out)
## all correctly classified because newsample is the training set
## (try UNSEEN validation data instead)
```

`tr.rocc` *Training of a ROC based classifier*

Description

The function establishes the ROC based classifier, returning the classifier specifications.

Usage

```
tr.rocc(g, out, xgenes = 200)
```

Arguments

`g`	the input data in form of a matrix with genes as rows and samples as columns. rownames(g) and colnames (g) must be specified.
`out`	describes the phenotype of the samples. a factor vector with levels 0 and 1 (in this order) with as many values as there are samples.
`xgenes`	numeric (vector of length 1), determines the number of features to be selected in feature selection.

Details

For feature selection the function picks the given number of xgenes with highest AUC (AUC below 0.5 are mirrored). Features negatively associated (AUC below 0.5) are multiplied by -1. The selected features are merged by the mean values to form a metagene. Samples are ranked according to the metagene expression. The optimal split of positive (i.e., 1) and negative (i.e., 0) samples is determined as the split yielding the highest accuracy, i.e. correct class assignments in respect to the real class. The split yielding optimal accuracy in the ROC curve is determined using the package

ROCR. The metagene threshold is computed as the mean metagene expression value of the two samples that build the boarder of the split. The final classifier specifications consist of a) the selected genes b) positive (AUC above 0.5) or negative (AUC below 0.5) association of these genes to the true class, and c) the metagene threshold. A new sample can be classified using the o.rocc() function.

Value

a list as a trocc object with components

`AUCs`	a matrix containing the selected features with corresponding AUC (aucv), positiv or negativ association (posneg), and mirrored AUC (allpos).
`genes`	character vector containing the genes selected in the feature selection.
`positiv`	character vector containing all positively associated genes (AUC above 0.5) selected in the feature selection.
`negativ`	character vector containing all negatively associated genes (AUC below 0.5) selected in the feature selection.
`metagene.expression`	numeric vector containing the metagene values of the training samples.
`metagene.expression.ranked`	numeric vector containing the samples ranked by metagene expression values.
`cutoffvalue`	the metagene threshold obtained from the best split of training samples.
`method`	the classification method used: ROC.based.predictor.

Note

depends on the package ROCR

Author(s)

Martin Lauss

References

Lauss M, Frigyesi A, Ryden T, Hoglund M. Robust assignment of cancer subtypes from expression data using a uni-variate gene expression average as classifier. BMC Cancer 2010 (in print)

See Also

p.rocc,o.rocc

Examples

```
### Random Dataset and phenotype
set.seed(100)
## Dataset should be a matrix
g <- matrix(rnorm(1000*25),ncol=25)
rownames(g) <- paste("Gene",1:1000,sep="_")
colnames(g) <- paste("Sample",1:25,sep="_")
```

```
## Phenotype should be a factor with levels 0 and 1:
out <- as.factor(sample(c(0:1),size=25,replace=TRUE))

predictor <- tr.rocc (g,out,xgenes=50)

## find classifier specification:
predictor$positiv
predictor$negativ
predictor$cutoffvalue
```

Index