Package ‘maPredictDSC’

January 14, 2017

Version 1.12.0
Date 2013-6-27
Title Phenotype prediction using microarray data: approach of the best overall team in the IMPROVER Diagnostic Signature Challenge
Author Adi Laurentiu Tarca <atarca@med.wayne.edu>
Depends R (>= 2.15.0), MASS, affy, limma, gcrma, ROC, class, e1071, caret, hgu133plus2.db, ROCR, AnnotationDbi, LungCancerACvsSCCGEO
Suggests parallel
Maintainer Adi Laurentiu Tarca <atarca@med.wayne.edu>
Description This package implements the classification pipeline of the best overall team (Team221) in the IMPROVER Diagnostic Signature Challenge. Additional functionality is added to compare 27 combinations of data preprocessing, feature selection and classifier types.
License GPL-2
URL http://bioinformaticsprb.med.wayne.edu/maPredictDSC
biocViews Microarray, Classification
Collate aggregateDSC.R, perfDSC.R, predictDSC.R, maPredictDSC.R
Imports
LazyLoad yes
NeedsCompilation no

R topics documented:

aggregateDSC .......................................................... 2
perfDSC ........................................................... 3
predictDSC .......................................................... 4

Index 7
aggregateDSC

Combine predictions from several fitted models fitted with predictDSC

Description
This function simply adds the posterior probabilities for a given class and sample from several models, and scales the resulting sums so that the sum over the classes is 1.0.

Usage
aggregateDSC(modlist)

Arguments
modlist
An object returned by predictDSC.

Details
See cited documents for more details.

Value
A a data frame with the predicted class membership belief value (posterior probability) for each sample (row) and each class (column).

Author(s)
Adi Laurentiu Tarca <atarca@med.wayne.edu>

References

Tarca AL, Than NG, Romero R, Methodological Approach from the Best Overall Team in the IMPROVER Diagnostic Signature Challenge, Systems Biomedicine, submitted, 2013.

See Also
predictDSC

Examples
#see function predictDSC for example
perfDSC

Area Under the Precision-Recall Curve (AUPR), Belief Confusion Metric (BCM) and Correct Class Enrichment Metric (CCEM).

Description

This function implements the three metrics used in the IMPROVER Diagnostic Signature Challenge.

Usage

perfDSC(pred, gs)

Arguments

pred  A belief matrix, with rows corresponding to samples and columns to classes. The values are between 0 and 1 and sum on each row is 1. It needs to have row names. The belief values are the result of a prediction made by a model.

gs   A matrix, with rows corresponding to samples and columns to classes that give the true (gold standard) class membership of samples.

Details

See cited documents for more details.

Value

A named vector that includes the BCM, CCEM, AUPR_avg and Accuracy.

Author(s)

Adi Laurentiu Tarca <atarca@med.wayne.edu>

References


See Also

predictDSC
Examples

```r
# assume a 3 class classification problem; gs is the gold standard and pred are predictions
gs = cbind(A=c(1,1,1,0,0,0,0,0,0,0,0,0), B=c(0,0,0,1,1,1,0,0,0,0,0,0), C=c(0,0,0,0,0,0,0,0,1,1,1,1))
rownames(gs) <- paste("sample", 1:12, sep="")
pred = cbind(A=c(0.6,0.9,1,0.3,0,0,0,0,0,0,0,0), B=c(0.4,0.1,0,0.7,1,1,0.7,1,0,0,0,0), C=c(0,0,0,0,0,0,0.3,0,1,1,1,1))
rownames(pred) <- paste("sample", 1:12, sep="")
# make sure the sum per row is 1 is both gs and pred
apply(gs, 1, sum)
apply(pred, 1, sum)
# compute performance
perfDSC(pred, gs)
```

predictDSC

Phenotype prediction using microarray data: approach of the best overall team in the IMPROVER Diagnostic Signature Challenge

Description

This function implements the classification pipeline of the best overall team (Team221) in the IMPROVER Diagnostic Signature Challenge. The function offers also exploring other combinations of data preprocessing, feature selection and classifier types.

Usage

```r
predictDSC(ano, celfile.path, annotation, preprocs=c("rma","gcrma","mas5"), filters=c("mttest","ttest","wilcox"), classifiers=c("LDA","kNN","svm"), FCT=1.0, CVP=4, NF=10, by=ifelse(NF>10,2,1), NR=5)
```

Arguments

- **ano**: A data frame with two columns: `files` and `group` giving the names of the Affymetrix .cel files (no full path) and their corresponding groups. Only two groups are allowed as well as a third group called "Test". The samples corresponding to these will not be used in training but will be used to normalize the training data with.
- **celfile.path**: The location of the directory where the .cel files are located.
- **annotation**: The names of a package that can be used to map the probesets to the ENTREZ gene IDs in order to deal with duplicate probesets per gene. E.g. `hgu133plus2.db`
- **preprocs**: A character vector giving the names of the normalization methods to try. Supported options are "rma", "gcrma", "mas5"
- **filters**: A character vector giving the names of the methods to use to rank features. Supported options are "mttest" for moderated t-test using limma package, "ttest" for regular t-test, and "wilcox" for wilcoxon test.
- **classifiers**: A character vector giving the names of the classifier types to use for learning the relation between expression levels and phenotype. Supported options are "LDA", "kNN", "svm".
predictDSC

**FCT**
A numeric value giving the fold change threshold to be used to filter out non-relevant features. Note, setting it to a too large value can produce an error as there need to be at least NF probesets with a fold change larger than FCT in each fold of the cross-validation.

**CVP**
The number of cross-validation partitions to create (minimum is 2). Do use a CVP value which ensures that at least two samples from the smallest group are kept for testing at each fold. E.g. If you have 10 samples in the smallest of the 2 groups a CVP of 4 would be maximum.

**NF**
The maximum number of features that would make sense to consider using as predictors in the models. NF should be less than the number of training samples.

**by**
The size of the step when searching for the number of features to include. By default the search starts with the top 2 features, and a number of "by" features are added up to NF.

**NR**
An integer number between 1 and Inf giving the number of times the cross-validation should be repeated to ensure a robust solution to the question: how many features to use as predictors in the model?.

**Details**
See cited documents for more details.

**Value**
A list object containing one item for each possible combination between the elements of preprocs, filters, and classifiers. Each item of the list contains the following information: predictions - a data frame with the predicted class membership belief value (posterior probability) for each sample (row) and each class (column). features - Names of the Affy probesets used as predictors by the model. A letter "F" is added as suffix to the probeset names. model - A fitted model object as produced by the lda, svm and kNN functions. performanceTr - A matrix giving the number of features tested (NN) mean AUC over all folds and repetitions (meanAUC), and the standard deviation of AUC values across folds and repeats of the cross-validation. bestAUC - The value of mean AUC corresponding to the optimal number of features chosen.

**Author(s)**
Adi Laurentiu Tarca <atarca@med.wayne.edu>

**References**

Tarca AL, Than NG, Romero R, Methodological Approach from the Best Overall Team in the IMPROVER Diagnostic Signature Challenge, Systems Biomedicine, submitted, 2013.

**See Also**
aggregateDSC
Examples

```r
library(maPredictDSC)
library(LungCancerACvsSCCGEO)
data(LungCancerACvsSCCGEO)
anoLC
gsLC
table(anoLC$group)

# run a series of methods combinations
modlist=predictDSC(ano=anoLC,celfile.path=system.file("extdata/lungcancer",package="LungCancerACvsSCCGEO"),
annotation="hgu133plus2.db",
preprocs=c("rma"),filters=c("mttest","wilcox"),FCT=1.0,classifiers=c("LDA","kNN"),
CVP=2,NF=4,NR=1)

# rank combinations by the performance on training data (AUC)
trainingAUC=sort(unlist(lapply(modlist,"[[","best_AUC")),decreasing=TRUE)
trainingAUC

# optional step; since we know the class of the test samples, let's see how the
# methods combinations perform on the test data
perfTest=function(out){
  perfDSC(pred=out$predictions,gs=gsLC)
}
testPerf=t(data.frame(lapply(modlist,perfTest)))
testPerf=as.data.frame(testPerf)
testPerf
testPerf=as.data.frame(testPerf)
testPerf

# aggregate predictions from top 3 combinations of methods
best3=names(trainingAUC)[1:3]
aggpred=aggregateDSC(modlist[best3])

# test the aggregated model on the test data
perfDSC(aggpred,gsLC)
```
Index

*Topic methods
  aggregateDSC, 2
  perfDSC, 3
  predictDSC, 4

*Topic parametric
  aggregateDSC, 2
  perfDSC, 3
  predictDSC, 4

aggregateDSC, 2, 5
perfDSC, 3
predictDSC, 2, 3, 4