Package 'maPredictDSC'

November 6, 2026

Version 1.51.1

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,jsonlite

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 https://tarcalab.med.wayne.edu/software/

biocViews Microarray, Classification

Collate aggregateDSC.R perfDSC.R predictDSC.R maPredictDSC.R getLungCancerData.R

LazyLoad yes

git_url https://git.bioconductor.org/packages/maPredictDSC

git_branch devel

git_last_commit 2a34fd8

git_last_commit_date 2026-10-05

Repository Bioconductor 3.24

Date/Publication 2026-11-05

Contents

aggregateDSC 2
getLungCancerData 3
LungCancerACvsSCCGEO 4
maPredictDSC-internal 4
perfDSC 5
predictDSC 6

Index 9

`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

Adi L. Tarca, Mario Lauria, Michael Unger, Erhan Bilal, Stephanie Boue, Kushal Kumar Dey, Julia Hoeng, Heinz Koeppl, Florian Martin, Pablo Meyer, Preetam Nandy, Raquel Norel, Manuel Peitsch, Jeremy J Rice, Roberto Romero, Gustavo Stolovitzky, Marja Talikka, Yang Xiang, Christoph Zechner, and IMPROVER DSC Collaborators, Strengths and limitations of microarray-based phenotype prediction: Lessons learned from the IMPROVER Diagnostic Signature Challenge. Bioinformatics, submitted 2013.

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
```

`getLungCancerData` *Download the Lung Cancer Example CEL Files for the former LungCancerACvsSCCGEO data package*

Description

This function downloads 30 .CEL file needed to illustrate predictive modeling starting with raw gene expression .CEL files.

Usage

```
getLungCancerData(ano)
```

Arguments

`ano`	A data frame with at least one columns: `files` that should have valid .CEL.gz file names from the zenodo record 'https://zenodo.org/api/records/23108090'

Details

See cited documents for more details.

Value

Returns the path to the location of the .CEL.gz files that were reterieved from zenodo.org. The files are cached locally and can be read directly by `ReadAffy`.

Author(s)

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

References

Tarca AL, Lauria M, Unger M, Bilal E, Boue S, Dey KK, Hoeng J, Koeppl H, Martin F, Meyer P, Nandy P, Norel R, Peitsch M, Rice JJ, Romero R, Stolovitzky G, Talikka M, Xiang Y, Zechner C, and IMPROVER DSC Collaborators. Strengths and limitations of microarray-based phenotype prediction: lessons learned from the IMPROVER Diagnostic Signature Challenge. *Bioinformatics*. 2013;29(22):2892-2899. doi:10.1093/bioinformatics/btt492.

Tarca AL, Than NG, Romero R. Methodological approach from the Best Overall Team in the sbv IMPROVER Diagnostic Signature Challenge.*Systems Biomedicine*. 2013;1(4):217-227. doi:10.4161/sysb.25980.

See Also

`maPredictDSC`

Examples

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

## Not run:
celfile.path <- getLungCancerData(anoLC)

## End(Not run)
```

LungCancerACvsSCCGEO *Annotation of a small set of training and test set samples (30 total) used by team 221 in the IMPROVER DSC for the lung cancer sub-challenge.*

Description

The LungCancerACvsSCCGEO dataset consists: i) a data frame anoLC giving the file names of the affy cel files used in the training phase and their corresponding phenotype (AC or SCC) and ii) gsLC the gold standard, i.e. the class membership of each test sample apearing in anoLC.

Usage

```
data(LungCancerACvsSCCGEO)
```

Source

GEO for the training data, while the test data comes from the citation below: Tarca AL, Lauria M, Unger M, Bilal E, Boue S, Dey KK, Hoeng J, Koeppl H, Martin F, Meyer P, Nandy P, Norel R, Peitsch M, Rice JJ, Romero R, Stolovitzky G, Talikka M, Xiang Y, Zechner C, and IMPROVER DSC Collaborators. Strengths and limitations of microarray-based phenotype prediction: lessons learned from the IMPROVER Diagnostic Signature Challenge. *Bioinformatics*. 2013;29(22):2892-2899. doi:10.1093/bioinformatics/btt492.

---

maPredictDSC-internal *Internal maPredictDSC functions*

Description

Internal maPredictDSC functions. maPredictDSC is the main function of the package called via predictDSC

Usage

```
maPredictDSC(ano,celfile.path,annotation,preproc.m="rma",
filter.m="mttest",FCT=1.0,classifier.m="LDA", otherCovariates=NULL,CVP=4,NF=20,by=ifelse(NF>10,2
```

Details

These are not to be called directly by the user.

`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 coresponding 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 coresponding 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

Adi L. Tarca, Mario Lauria, Michael Unger, Erhan Bilal, Stephanie Boue, Kushal Kumar Dey, Julia Hoeng, Heinz Koeppl, Florian Martin, Pablo Meyer, Preetam Nandy, Raquel Norel, Manuel Peitsch, Jeremy J Rice, Roberto Romero, Gustavo Stolovitzky, Marja Talikka, Yang Xiang, Christoph Zechner, and IMPROVER DSC Collaborators, Strengths and limitations of microarray-based phenotype prediction: Lessons learned from the IMPROVER Diagnostic Signature Challenge. Bioinformatics, submitted 2013.

See Also

`predictDSC`

Examples

```
#asume a 3 class classification problem; gs is the gold standard and pred are predictions
gs=cbind(A=c(1,1,1,1,0,0,0,0,0,0,0,0),B=c(0,0,0,0,1,1,1,1,0,0,0,0),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
rownames(pred)<-paste("sample",1:12,sep="")
#male sure the sum per row is 1 is both gs and pred
apply(gs,1,sum)
apply(pred,1,sum)
#compute perfromance
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 ofers also eploring other combinations of data preprocessing, feature selection and classifier types.

Usage

```
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 pre gene. E.g.`hgu133plues2.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".

FCT — A numeric value giving the fold change threshokd 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 probestes 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 smalest group are kept for testing at each fold. E.g. If you have 10 samples in the smalest 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 predcitors 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 th esearch 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 accross 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, Lauria M, Unger M, Bilal E, Boue S, Dey KK, Hoeng J, Koeppl H, Martin F, Meyer P, Nandy P, Norel R, Peitsch M, Rice JJ, Romero R, Stolovitzky G, Talikka M, Xiang Y, Zechner C, and IMPROVER DSC Collaborators. Strengths and limitations of microarray-based phenotype prediction: lessons learned from the IMPROVER Diagnostic Signature Challenge. *Bioinformatics*. 2013;29(22):2892-2899. doi:10.1093/bioinformatics/btt492.

Tarca AL, Than NG, Romero R. Methodological approach from the Best Overall Team in the sbv IMPROVER Diagnostic Signature Challenge.*Systems Biomedicine*. 2013;1(4):217-227. doi:10.4161/sysb.25980.

See Also

aggregateDSC

Examples

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

## Not run:
celfile.path <- getLungCancerData(anoLC)

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

#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=testPerf[order(testPerf[,"AUC"],decreasing=TRUE),]
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)

## End(Not run)
```

Index

∗ **datasets**
LungCancerACvsSCCGEO, 4
∗ **internal**
maPredictDSC-internal, 4
∗ **methods**
aggregateDSC, 2
getLungCancerData, 3
perfDSC, 5
predictDSC, 6
∗ **parametric**
aggregateDSC, 2
getLungCancerData, 3
perfDSC, 5
predictDSC, 6

aggregateDSC, 2, *7*
anoLC *(*LungCancerACvsSCCGEO*)*, 4

getLungCancerData, 3
gsLC *(*LungCancerACvsSCCGEO*)*, 4

LungCancerACvsSCCGEO, 4

maPredictDSC, *3*
maPredictDSC *(*maPredictDSC-internal*)*, 4
maPredictDSC-internal, 4

perfDSC, 5
predictDSC, *2*, *5*, 6

ReadAffy, *3*