Package 'classpredict'

June 27, 2019

Title Class Prediction with Mutiple Methods

Version 0.2

Description Implements the class prediction tool using multiple methods in ArrayTools

Depends R (>= 3.1.0)

License Same as BRB-ArrayTools
(https://brb.nci.nih.gov/BRB-ArrayTools/)

Imports ROC

Author BRB-ArrayTools team <arraytools@emmes.com>

Maintainer BRB-ArrayTools team <arraytools@emmes.com>

RoxygenNote 6.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

`classPredict` *Class prediction*

Description

This function calculates multiple classifiers that are used to predict the class of a new sample. It implements the class prediction tool with multiple methods in BRB-ArrayTools.

Usage

```
classPredict(exprTrain, exprTest = NULL, isPaired = FALSE,
  pairVar.train = NULL, pairVar.test = NULL, geneId, cls,
  pmethod = c("ccp", "bcc", "dlda", "knn", "nc", "svm"),
  geneSelect = "igenes.univAlpha", univAlpha = 0.001, univMcr = 0.2,
  foldDiff = 2, rvm = FALSE, filter = NULL, ngenePairs = 25,
  nfrvm = 10, cvMethod = 1, kfoldValue = 10, bccPrior = 1,
  bccThresh = 0.8, nperm = 0, svmCost = 1, svmWeight = 1,
  fixseed = 1, prevalence = NULL, projectPath,
  outputName = "ClassPrediction", generateHTML = FALSE)
```

Arguments

`exprTrain`	matrix of gene expression data for training samples. Rows are genes and columns are arrays. Its column names must be provided.
`exprTest`	matrix of gene expression data for new samples. Its column names must be provided.
`isPaired`	logical. If `TRUE`, samples are paired.
`pairVar.train`	vector of pairing variables for training samples.
`pairVar.test`	vector of pairing variables for new samples.
`geneId`	matrix/data frame of gene IDs.
`cls`	vector of training sample classes.
`pmethod`	character string vector of prediction methods to be employed. • "ccp": Compound Covariate Predictor • "bcc": Bayesian Compound Covariate Predictor • "dlda": Diagonal Linear Discriminant Analysis • "knn": 1-Nearest Neighbor/ 3-Nearest Neighbor • "nc": Nearest Centroid • "svm": Support Vector Machine
`geneSelect`	character string for gene selection method. • "igenes.univAlpha": select individual genes univariately significantly differentially expressed between the classes at the specified threshold significance level. • "igenes.grid": select individual genes that optimize over the grid of alpha levels. • "igenes.univMcr": select individual genes with univariate misclassification rate below a specified value. • "gpairs": select gene pairs bye the "greedy pairs" method. • "rfe": select genes by recursive feature elimination.
`univAlpha`	numeric for a significance level. Default is 0.001.
`univMcr`	numeric for univariate misclassification rate. Default is 0.2.

`foldDiff` numeric for fold ratio of geometric means between two classes exceeding. 0 means not to enable this option. Default is 2.

`rvm` logical. If `TRUE`, random variance model will be employed. Default is `FALSE`.

`filter` vector of 1/0's of the same length as genes. 1 means to keep the gene while 0 means to exclude genes from class comparison analysis. If `rvm = TRUE`, all genes will be used in random variance model estimation. Default is `FALSE`.

`nfrvm` numeric specifying the number of features selected by the support vector machine recursive feature elimination method. Default is 10.

`cvMethod` numeric for the cross validation method. Default is 1.

- 1: leave-one-out CV,
- 2: k-fold CV,
- 3: 0.632+ bootstrap.

`kfoldValue` numeric specifying the number of folds if K-fold method is selected. Default is 10.

`bccPrior` numeric specifying the prior probability option for the Baysian compound covariate prediction. If `bccPrior == 1`, equal prior probabilities will be applied. If `bccPrior == 2`, prior probabilities based on the proportions in training data are applied. Default is 1.

`bccThresh` numeric specifying the uncertainty threshold for the Bayesian compound covariate prediction. Default is 0.8.

`nperm` numeric specifying the number of permutations for the significance test of cross-validated mis-classification rate. It should be equal to zero or greater than 50. Default is 0.

`svmCost` numeric specifying the cost values for SVM. Default is 1.

`svmWeight` numeric specifying the weight values for SVM. Default is 1.

`fixseed` numeric. `fixseed == 1` if a fixed seed is used; otherwise, `fixseed == 0`. Default is 1.

`prevalence` vector for class prevalences. When prevalence is `NULL`, the proportional of samples in each class will be the estimate of class prevalence. Default is `NULL`. Names of vector should be provided and consistent with classes in `cls`.

`projectPath` character string specifying the full project path.

`outputName` character string specifying the output folder name. Default is "ClassPrediction".

`generateHTML` logical. If `TRUE`, an HTML page will be generated with detailed class prediction results saved in <projectPath>/Output/<outputName>/<outputName>.html.

`ngenePairs:` numeric specifying the number of gene pairs selected by the greedy pairs method. Default is 25.

Details

Please see the BRB-ArrayTools manual (https://brb.nci.nih.gov/BRB-ArrayTools/Documentation.html) for details.

Value

A list that may include the following objects:

- `performClass`: a data frame with the performance of classifiers during cross-validation:
- `percentCorrectClass`: a data frame with the mean percent of correct classification for each sample using different prediction methods.
- `predNewSamples`:s a data frame with predicted class for each new sample. 'NC' means that a sample is not classified. In this example, there are four new samples.
- `probNew`: a data frame with the predicted probability of each new sample belonginG to the class (BRCA1) from the the Bayesian Compound Covariate method.
- `classifierTable`: a data frame with composition of classifiers such as geometric means of values in each class, p-values and Gene IDs.
- `probInClass`: a data frame with predicted probability of each training sample belonging to aclass during cross-validation from the Bayesian Compound Covariate
- `CCPSenSpec`: a data frame with performance (i.e., sensitivity, specificity, positive prediction value, negative prediction value) of the Compound Covariate Predictor Classifier.
- `LDASenSpec`: a data frame with performance (i.e., sensitivity, specificity, positive prediction value, negative prediction value) of the Diagonal Linear Discriminant Analysis Classifier.
- `K1NNSenSpec`: a data frame with performance (i.e., sensitivity, specificity, positive prediction value, negative prediction value) of the 1-Nearest Neighbor Classifier.
- `K3NNSenSpec`: a data frame with performance (i.e., sensitivity, specificity, positive prediction value, negative prediction value) of the 3-Nearest Neighbor Classifier.
- `CentroidSenSpec`: a data frame with performance (i.e., sensitivity, specificity, positive prediction value, negative prediction value) of the Nearest Centroid Classifierr.
- `SVMSenSpec`: a data frame with performance (i.e., sensitivity, specificity, positive prediction value, negative prediction value) of the Support Vector Machine Classifier.
- `BCPPSenSpec`: a data frame with performance (i.e., sensitivity, specificity, positive prediction value, negative prediction value) of the Bayesian Compound Covariate Classifier.
- `weightLinearPred`: a data frame with gene weights for linear predictors such as Compound Covariate Predictor, Diagonal Linear Discriminat Analysis and Support Vector Machine.
- `thresholdLinearPred`: a numeric vector of the thresholds for the linear prediction rules related with `weightLinearPred`. Each prediction rule is defined by the inner sum of the weights (w_i) and log expression values (x_i) of significant genes. In this case, a sample is classified to the class BRCA1 if the sum is greater than the threshold; that is, $\sum_i w_i x_i > threshold$.
- `GRPCentroid`: a data frame with centroid of each class for each predictor gene.
- `ppval`: a vector of permutation p-values of statistical significance tests of cross-validated estimate of misclassification rate from specified #' prediction methods.
- `pmethod`: a vector of prediction methods that are specified.
- `workPath`: the path for fortran and other intermediate outputs.

Examples

```
dataset<-"Brca"
# gene IDs
geneId <- read.delim(system.file("extdata", paste0(dataset, "_GENEID.txt"),
                     package = "classpredict"), as.is = TRUE, colClasses = "character")
# expression data
x <- read.delim(system.file("extdata", paste0(dataset, "_LOGRAT.TXT"),
                 package = "classpredict"), header = FALSE)
# filter information, 1 - pass the filter, 0 - filtered
filter <- scan(system.file("extdata", paste0(dataset, "_FILTER.TXT"),
                package = "classpredict"), quiet = TRUE)
# class information
expdesign <- read.delim(system.file("extdata", paste0(dataset, "_EXPDESIGN.txt"),
                        package = "classpredict"), as.is = TRUE)
# training/test information
testSet <- expdesign[, 10]
trainingInd <- which(testSet == "training")
predictInd <- which(testSet == "predict")
ind1 <- which(expdesign[trainingInd, 4] == "BRCA1")
ind2 <- which(expdesign[trainingInd, 4] == "BRCA2")
ind <- c(ind1, ind2)
exprTrain <- x[, ind]
colnames(exprTrain) <- expdesign[ind, 1]
exprTest <- x[, predictInd]
colnames(exprTest) <- expdesign[predictInd, 1]
projectPath <- file.path(Sys.getenv("HOME"),"Brca")
outputName <- "ClassPrediction"
generateHTML <- TRUE
resList <- classPredict(exprTrain = exprTrain, exprTest = exprTest, isPaired = FALSE,
                        pairVar.train = NULL, pairVar.test = NULL, geneId,
                        cls = c(rep("BRCA1", length(ind1)), rep("BRCA2", length(ind2))),
                        pmethod = c("ccp", "bcc", "dlda", "knn", "nc", "svm"),
                        geneSelect = "igenes.univAlpha",
                        univAlpha = 0.001, univMcr = 0, foldDiff = 0, rvm = TRUE,
                        filter = filter, ngenePairs = 25, nfrvm = 10, cvMethod = 1,
                        kfoldValue = 10, bccPrior = 1, bccThresh = 0.8, nperm = 0,
                        svmCost = 1, svmWeight =1, fixseed = 1, prevalence = NULL,
                        projectPath = projectPath, outputName = outputName, generateHTML)
if (generateHTML)
  browseURL(file.path(projectPath, "Output", outputName,
            paste0(outputName, ".html")))
```

`plotROCCurve` *Plot ROC curves*

Description

This function plots ROC curves for compound covariate predictor, diagonal linear discriminant analysis and Bayesian compound covariate predictor methods.

Usage

```
plotROCCurve(list, method)
```

Arguments

list	list returned by function classPredic.
method	character string of a prediction method. • "ccp": compound covariate predictor • "dlda": diagonal linear discriminant analysis • "bcc": Bayesian compound covariate predictor

Examples

```
res <- test.classPredict("Brca")
plotROCCurve(res, "ccp")
plotROCCurve(res, "dlda")
plotROCCurve(res, "bcc")
```

---

test.classPredict *Test classpredict() function*

Description

This function will load a test dataset to run classPredict function.

Usage

```
test.classPredict(dataset = c("Brca", "Perou", "Pomeroy"), projectPath,
  outputName = "ClassPrediction", generateHTML = FALSE)
```

Arguments

dataset	character string specifying one of "Brca", "Perou" or "Pomeroy" datasets.
projectPath	characteer string specifying the project path. Default is C:/Users/UserName/Documents/$dataset.
outputName	character string for the output folder name.
generateHTML	logical. If TRUE, an html page will be generated with detailed prediction results.

Details

If the random variance model is enabled, all genes will be used for the model estimation.

Value

A list as returned by classPredict.

See Also

classPredict

Examples

```
test.classPredict("Pomeroy")
```

Index