

Package ‘MAclinical’

May 29, 2009

Version 1.0-4

Date 2009-05-29

Title Class prediction based on microarray data and clinical parameters

Author Anne-Laure Boulesteix <boulesteix@slcmsr.org>.

Maintainer Anne-Laure Boulesteix <boulesteix@slcmsr.org>

Depends R (>= 2.4.1), party, plsgenomics, st, e1071

Suggests

Description ‘MAclinical’ implements class prediction using both microarray data and clinical parameters. It addresses the question of the additional predictive value of microarray data. Class prediction is performed using a two-step method combining (pre-validated) PLS dimension reduction and random forests.

License GPL (>= 2)

URL <http://cran.r-project.org/web/packages/MAclinical/index.html>

Repository CRAN

Date/Publication 2009-05-29 13:19:42

R topics documented:

generate.learningsets	2
logistic_z	3
plsr_x	5
plsr_xz	7
plsr_xz_pv	9
plsr_x_pv	11
rf_z	13
simulate	14
svm_x	16
testclass	17
testclass_simul	20

`generate.learningsets`*Generating learning sets*

Description

This function generates a design matrix giving the indices of observations forming the learning data set for several iterations.

Usage

```
generate.learningsets(n, method, fold=NULL, niter=NULL, nlearn=NULL)
```

Arguments

<code>n</code>	The total number of observations in the available data set.
<code>method</code>	One of "LOOCV" (leave-one-out cross-validation), "CV" (cross-validation), "MCCV" (Monte-Carlo cross-validation, also called subsampling), "bootstrap" (bootstrap sampling - with replacement).
<code>fold</code>	Gives the number of CV-groups. Used only when <code>method="CV"</code> .
<code>niter</code>	Number of iterations.
<code>nlearn</code>	Number of observations in the learning sets. Used only for <code>method="MCCV"</code> and <code>method="bootstrap"</code> . When <code>method="bootstrap"</code> , the default is <code>nlearn=n</code> .

Details

When `method="CV"`, `niter` gives the number of times the whole CV-procedure is repeated. The output matrix has then `foldxniter` rows. When `method="MCCV"` or `method="bootstrap"`, `niter` is simply the number of considered learning sets. Note that `method="CV"`, `fold=n` is equivalent to `method="LOOCV"`.

Value

A matrix giving the indices (from 1 to `n`) of the observations included in the learning sets. Each row corresponds to a learning set. The order of the columns is not important. The number of rows is equal to `n` when `method="LOOCV"`, `niter` when `method="MCCV"` or `method="bootstrap"`, `fold` when `method="CV"` and `niter` is null, and `fold x niter` when `method="CV"` and `niter` is non-null.

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

See Also

testclass.

Examples

```
# load MAclinical library
# library(MAclinical)

# LOOCV
generate.learningsets(n=40,method="LOOCV")

# CV
generate.learningsets(n=40,method="CV",fold=5)
generate.learningsets(n=40,method="CV",fold=5,niter=3)

# MCCV
generate.learningsets(n=40,method="MCCV",niter=3,nlearn=30)

# bootstrap
generate.learningsets(n=40,method="bootstrap",niter=3)
```

logistic_z	<i>Class prediction based on logistic regression using clinical parameters only</i>
------------	---

Description

This function builds a prediction rule based on the learning data (clinical predictors only) and applies it to the test data. It uses the function `glm`.

Usage

```
logistic_z(Xlearn=NULL,Zlearn,Ylearn,Xtest=NULL,Ztest,...)
```

Arguments

Xlearn	A nlearn x p matrix giving the microarray predictors for the learning data set. This argument is ignored.
Zlearn	A nlearn x q matrix giving the clinical predictors for the learning data set.
Ylearn	A numeric vector of length nlearn giving the class membership of the learning observations, coded as 0,1.
Xtest	A ntest x p matrix giving the microarray predictors for the test data set. This argument is ignored.

`Ztest` A `n`test x `q` matrix giving the clinical predictors for the test data set.
... Other arguments.

Details

See Boulesteix et al (2008).

Value

A list with the element:

`prediction` A numeric vector of length `nrow(Xtest)` giving the predicted class for each observation from the test data set.

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

See Also

[testclass](#), [testclass_simul](#), [simulate](#), [plsr_x_pv](#), [plsr_xz_pv](#), [plsr_x](#), [plsr_xz](#), [rf_z](#), [svm_x](#).

Examples

```
# load MAclinical library
# library(MAclinical)

# Generating zlearn, ylearn, ztest
zlearn<-matrix(rnorm(120),30,4)
ylearn<-sample(0:1,30,replace=TRUE)
ztest<-matrix(rnorm(80),20,4)

my.prediction<-logistic_z(Zlearn=zlearn,Ylearn=ylearn,Ztest=ztest)
my.prediction
```

plsrf_x	<i>Classification based on PLS dimension reduction and random forests using microarray data only</i>
---------	--

Description

This function builds a prediction rule based on the learning data (microarray predictors only) and applies it to the test data. The classifier consists of two steps: PLS dimension reduction (without pre-validation step) for summarizing microarray data, and random forests applied to the obtained PLS components. See Boulesteix et al (2008) for more details.

The function `plsrf_x` uses the functions `cforest` and `varimp` from the package `party` and the function `pls.regression` from the package `plsgenomics`.

Usage

```
plsrf_x(Xlearn, Zlearn=NULL, Ylearn, Xtest, Ztest=NULL, ncomp=0:3, ordered=NULL, nbgene=NU
```

Arguments

<code>Xlearn</code>	A <code>nlearn</code> x <code>p</code> matrix giving the microarray predictors for the learning data set.
<code>Zlearn</code>	A <code>nlearn</code> x <code>q</code> matrix giving the clinical predictors for the learning data set. This argument is ignored.
<code>Ylearn</code>	A numeric vector of length <code>nlearn</code> giving the class membership of the learning observations, coded as 0,...,K-1 (where K is the number of classes).
<code>Xtest</code>	A <code>ntest</code> x <code>p</code> matrix giving the microarray predictors for the test data set.
<code>Ztest</code>	A <code>ntest</code> x <code>q</code> matrix giving the clinical predictors for the test data set. This argument is ignored.
<code>ncomp</code>	A numeric vector giving the candidate numbers of PLS components. All numbers must be >0.
<code>ordered</code>	A vector of length <code>p</code> giving the order of the microarray predictors in terms of relevance for prediction. For instance, if the three first elements of <code>ordered</code> are 30,2,2400, it means that the most relevant genes are the genes in the 30th, 2nd and 2400th columns of the gene expression data matrix <code>Xlearn</code> . Note: if <code>ordered=NULL</code> , the columns of <code>Xlearn</code> and <code>Xtest</code> are assumed to be already ordered.
<code>nbgene</code>	The number of genes to be selected for use in dimension reduction. Default is <code>nbgene=NULL</code> , in which case all genes are used.
<code>...</code>	Other arguments to be passed to the function <code>cforest_control</code> from the <code>party</code> package.

Details

See Boulesteix et al (2008).

Value

A list with the elements:

prediction	A numeric vector of length <code>nrow(Xtest)</code> giving the predicted class for each observation from the test data set.
importance	The variable importance information output by the function <code>varimp</code> from the package <code>party</code> for the corresponding forest.
bestncomp	The best number of PLS components, as obtained using the model selection method based on the out-of-bag error.
OOB	A numeric vector of length <code>ncomp</code> giving the out-of-bag error of the forest constructed with the corresponding number of PLS components.

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

See Also

[testclass](#), [testclass_simul](#), [simulate](#), [plsr_x_pv](#), [plsr_xz](#), [plsr_xz_pv](#), [rf_z](#), [logistic_z](#), [svm_x](#).

Examples

```
# load MAclinical library
# library(MAclinical)

# Generating xlearn, zlearn, ylearn, xtest, ztest
xlearn<-matrix(rnorm(3000),30,100)
zlearn<-matrix(rnorm(120),30,4)
ylearn<-sample(0:1,30,replace=TRUE)
xtest<-matrix(rnorm(2000),20,100)
ztest<-matrix(rnorm(80),20,4)

my.prediction1<-plsr_x(Xlearn=xlearn,Ylearn=ylearn,Xtest=xtest)

ordered<-sample(100)
my.prediction2<-plsr_x(Xlearn=xlearn,Ylearn=ylearn,Xtest=xtest,ordered=ordered,nbgene=20)
```

plsrf_xz	<i>Classification based on PLS dimension reduction and random forests using both clinical and microarray predictors</i>
----------	---

Description

This function builds a prediction rule based on the learning data (both clinical and microarray predictors) and applies it to the test data. The classifier consists of two steps: PLS dimension reduction (without pre-validation step) for summarizing microarray data, and random forests applied to both PLS components and clinical predictors. See Boulesteix et al (2008) for more details.

The function `plsrf_xz` uses the functions `cforest` and `varimp` from the package `party` and the function `pls.regression` from the package `plsgenomics`.

Usage

```
plsrf_xz(Xlearn, Zlearn, Ylearn, Xtest, Ztest, ncomp=0:3, ordered=NULL, nbgene=NULL, ...)
```

Arguments

<code>Xlearn</code>	A <code>nlearn</code> x <code>p</code> matrix giving the microarray predictors for the learning data set.
<code>Zlearn</code>	A <code>nlearn</code> x <code>q</code> matrix giving the clinical predictors for the learning data set.
<code>Ylearn</code>	A numeric vector of length <code>nlearn</code> giving the class membership of the learning observations, coded as 0,...,K-1 (where K is the number of classes).
<code>Xtest</code>	A <code>ntest</code> x <code>p</code> matrix giving the microarray predictors for the test data set.
<code>Ztest</code>	A <code>ntest</code> x <code>q</code> matrix giving the clinical predictors for the test data set.
<code>ncomp</code>	A numeric vector giving the candidate numbers of PLS components. All numbers must be ≥ 0 . The number 0 corresponds to prediction based on clinical parameters only.
<code>ordered</code>	A vector of length <code>p</code> giving the order of the microarray predictors in terms of relevance for prediction. For instance, if the three first elements of <code>ordered</code> are 30,2,2400, it means that the most relevant genes are the genes in the 30th, 2nd and 2400th columns of the gene expression data matrix <code>Xlearn</code> . Note: if <code>ordered=NULL</code> , the columns of <code>Xlearn</code> and <code>Xtest</code> are assumed to be already ordered.
<code>nbgene</code>	The number of genes to be selected for use in dimension reduction. Default is <code>nbgene=NULL</code> , in which case all genes are used.
<code>...</code>	Other arguments to be passed to the function <code>cforest_control</code> from the <code>party</code> package.

Details

See Boulesteix et al (2008).

Value

A list with the elements:

prediction	A numeric vector of length <code>nrow(Xtest)</code> giving the predicted class for each observation from the test data set.
importance	The variable importance information output by the function <code>varimp</code> from the package <code>party</code> for the corresponding forest.
bestncomp	The best number of PLS components, as obtained using the model selection method based on the out-of-bag error.
OOB	A numeric vector of length <code>ncomp</code> giving the out-of-bag error of the forest constructed with the corresponding number of PLS components.

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

See Also

`testclass`, `testclass_simul`, `simulate`, `plsr_x_pv`, `plsr_xz_pv`, `plsr_x`, `rf_z`, `logistic_z`, `svm_x`.

Examples

```
# load MAclinical library
# library(MAclinical)

# Generating xlearn, zlearn, ylearn, xtest, ztest
xlearn<-matrix(rnorm(3000), 30, 100)
zlearn<-matrix(rnorm(120), 30, 4)
ylearn<-sample(0:1, 30, replace=TRUE)
xtest<-matrix(rnorm(2000), 20, 100)
ztest<-matrix(rnorm(80), 20, 4)

my.prediction1<-plsr_xz(Xlearn=xlearn, Zlearn=zlearn, Ylearn=ylearn, Xtest=xtest, Ztest=ztest)

ordered<-sample(100)
my.prediction2<-plsr_xz(Xlearn=xlearn, Zlearn=zlearn, Ylearn=ylearn,
Xtest=xtest, Ztest=ztest, ordered=ordered, nbgene=20)
```

 plsrfr_xz_pv

Classification based on pre-validated PLS dimension reduction and random forests using both clinical and microarray predictors

Description

This function builds a prediction rule based on the learning data (both clinical and microarray predictors) and applies it to the test data. The classifier consists of two steps: PLS dimension reduction involving a pre-validation step for summarizing microarray data, and random forests applied to both PLS components and clinical predictors. See Boulesteix et al (2008) for more details.

The function `plsrfr_xz_pv` uses the functions `cforest` and `varimp` from the package `party` and the function `pls.regression` from the package `plsgenomics`.

Usage

```
plsrfr_xz_pv(Xlearn, Zlearn, Ylearn, Xtest, Ztest, ncomp=0:3,
            ordered=NULL, nbgene=NULL, fold=10, ...)
```

Arguments

<code>Xlearn</code>	A <code>nlearn</code> x <code>p</code> matrix giving the microarray predictors for the learning data set.
<code>Zlearn</code>	A <code>nlearn</code> x <code>q</code> matrix giving the clinical predictors for the learning data set.
<code>Ylearn</code>	A numeric vector of length <code>nlearn</code> giving the class membership of the learning observations, coded as 0,...,K-1 (where K is the number of classes).
<code>Xtest</code>	A <code>ntest</code> x <code>p</code> matrix giving the microarray predictors for the test data set.
<code>Ztest</code>	A <code>ntest</code> x <code>q</code> matrix giving the clinical predictors for the test data set.
<code>ncomp</code>	A numeric vector giving the candidate numbers of pre-validated PLS components. All numbers must be ≥ 0 . The number 0 corresponds to prediction based on clinical parameters only.
<code>ordered</code>	A vector of length <code>p</code> giving the order of the microarray predictors in terms of relevance for prediction. For instance, if the three first elements of <code>ordered</code> are 30,2,2400, it means that the most relevant genes are the genes in the 30th, 2nd and 2400th columns of the gene expression data matrix <code>Xlearn</code> . Note: if <code>ordered=NULL</code> , the columns of <code>Xlearn</code> and <code>Xtest</code> are assumed to be already ordered.
<code>nbgene</code>	The number of genes to be selected for use in dimension reduction. Default is <code>nbgene=NULL</code> , in which case all genes are used.
<code>fold</code>	The number of folds for the pre-validation step. See Boulesteix et al (2008) for more details. The default is <code>fold=10</code> .
<code>...</code>	Other arguments to be passed to the function <code>cforest_control</code> from the <code>party</code> package.

Details

See Boulesteix et al (2008).

Value

A list with the elements:

prediction	A numeric vector of length <code>nrow(Xtest)</code> giving the predicted class for each observation from the test data set.
importance	The variable importance information output by the function <code>varimp</code> from the package <code>party</code> for the corresponding forest.
bestncomp	The best number of pre-validated PLS components, as obtained using the model selection method based on the out-of-bag error.
OOB	A numeric vector of length <code>ncomp</code> giving the out-of-bag error of the forest constructed with the corresponding number of pre-validated PLS components.

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

Tibshirani R, Efron B, 2002. Pre-validation and inference in microarrays. *Stat. Appl. Genet. Mol. Biol.* 1:1.

See Also

[testclass](#), [testclass_simul](#), [simulate](#), [plsr_x](#), [plsr_x_pv](#), [plsr_xz](#), [rf_z](#), [logistic_z](#), [svm_x](#).

Examples

```
# load MAclinical library
# library(MAclinical)

# Generating xlearn, zlearn, ylearn, xtest, ztest
xlearn<-matrix(rnorm(3000), 30, 100)
zlearn<-matrix(rnorm(120), 30, 4)
ylearn<-sample(0:1, 30, replace=TRUE)
xtest<-matrix(rnorm(2000), 20, 100)
ztest<-matrix(rnorm(80), 20, 4)

my.prediction1<-plsr_xz_pv(Xlearn=xlearn, Zlearn=zlearn, Ylearn=ylearn,
Xtest=xtest, Ztest=ztest)

ordered<-sample(100)
my.prediction2<-plsr_xz_pv(Xlearn=xlearn, Zlearn=zlearn, Ylearn=ylearn,
Xtest=xtest, Ztest=ztest, ordered=ordered, nbgene=20)
my.prediction3<-plsr_xz_pv(Xlearn=xlearn, Zlearn=zlearn, Ylearn=ylearn,
Xtest=xtest, Ztest=ztest, fold=30)
```

plsrf_x_pv

Classification based on pre-validated PLS dimension reduction and random forests using microarray data only

Description

This function builds a prediction rule based on the learning data (microarray predictors only) and applies it to the test data. The classifier consists of two steps: PLS dimension reduction with pre-validation step for summarizing microarray data, and random forests applied to the obtained PLS components. See Boulesteix et al (2008) for more details.

The function `plsrf_x_pv` uses the functions `cforest` and `varimp` from the package `party` and the function `pls.regression` from the package `plsgenomics`.

Usage

```
plsrf_x_pv(Xlearn, Zlearn=NULL, Ylearn, Xtest, Ztest=NULL, ncomp=0:3,
ordered=NULL, nbgene=NULL, fold=10, ...)
```

Arguments

<code>Xlearn</code>	A <code>nlearn x p</code> matrix giving the microarray predictors for the learning data set.
<code>Zlearn</code>	A <code>nlearn x q</code> matrix giving the clinical predictors for the learning data set. This argument is ignored.
<code>Ylearn</code>	A numeric vector of length <code>nlearn</code> giving the class membership of the learning observations, coded as <code>0,...,K-1</code> (where <code>K</code> is the number of classes).
<code>Xtest</code>	A <code>ntest x p</code> matrix giving the microarray predictors for the test data set.
<code>Ztest</code>	A <code>ntest x q</code> matrix giving the clinical predictors for the test data set. This argument is ignored.
<code>ncomp</code>	A numeric vector giving the candidate numbers of pre-validated PLS components. All numbers must be >0 .
<code>ordered</code>	A vector of length <code>p</code> giving the order of the microarray predictors in terms of relevance for prediction. For instance, if the three first elements of <code>ordered</code> are <code>30,2,2400</code> , it means that the most relevant genes are the genes in the 30th, 2nd and 2400th columns of the gene expression data matrix <code>Xlearn</code> . Note: if <code>ordered=NULL</code> , the columns of <code>Xlearn</code> and <code>Xtest</code> are assumed to be already ordered.
<code>nbgene</code>	The number of genes to be selected for use in dimension reduction. Default is <code>nbgene=NULL</code> , in which case all genes are used.
<code>fold</code>	The number of folds for the pre-validation step. See Boulesteix et al (2008) for more details. The default is <code>fold=10</code> .
<code>...</code>	Other arguments to be passed to the function <code>cforest_control</code> from the <code>party</code> package.

Details

See Boulesteix et al (2008).

Value

A list with the elements:

prediction	A numeric vector of length <code>nrow(Xtest)</code> giving the predicted class for each observation from the test data set.
importance	The variable importance information output by the function <code>varimp</code> from the package <code>party</code> for the corresponding forest.
bestncomp	The best number of pre-validated PLS components, as obtained using the model selection method based on the out-of-bag error.
OOB	A numeric vector of length <code>ncomp</code> giving the out-of-bag error of the forest constructed with the corresponding number of pre-validated PLS components.

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

Tibshirani R, Efron B, 2002. Pre-validation and inference in microarrays. *Stat. Appl. Genet. Mol. Biol.* 1:1.

See Also

[testclass](#), [testclass_simul](#), [simulate](#), [plsr_x](#), [plsr_xz](#), [plsr_xz_pv](#), [rf_z](#), [logistic_z](#), [svm_x](#).

Examples

```
# load MAclinical library
# library(MAclinical)

# Generating xlearn, zlearn, ylearn, xtest, ztest
xlearn<-matrix(rnorm(3000), 30, 100)
ylearn<-sample(0:1, 30, replace=TRUE)
xtest<-matrix(rnorm(2000), 20, 100)

my.prediction1<-plsr_x_pv(Xlearn=xlearn, Ylearn=ylearn, Xtest=xtest)

ordered<-sample(100)
my.prediction2<-plsr_x(Xlearn=xlearn, Ylearn=ylearn, Xtest=xtest, ordered=ordered, nbgene=20)
my.prediction3<-plsr_x_pv(Xlearn=xlearn, Ylearn=ylearn, Xtest=xtest, fold=30)
```

rf_z	<i>Class prediction based on random forests using clinical parameters only</i>
------	--

Description

This function builds a prediction rule based on the learning data (clinical predictors only) and applies it to the test data. It uses the function `cforest` from the package `party`. See Boulesteix et al (2008) for more details.

Usage

```
rf_z(Xlearn=NULL, Zlearn, Ylearn, Xtest=NULL, Ztest, ...)
```

Arguments

Xlearn	A nlearn x p matrix giving the microarray predictors for the learning data set. This argument is ignored.
Zlearn	A nlearn x q matrix giving the clinical predictors for the learning data set.
Ylearn	A numeric vector of length nlearn giving the class membership of the learning observations, coded as 0,...,K-1 (where K is the number of classes).
Xtest	A ntest x p matrix giving the microarray predictors for the test data set. This argument is ignored.
Ztest	A ntest x q matrix giving the clinical predictors for the test data set.
...	Other arguments to be passed to the function <code>cforest_control</code> from the <code>party</code> package.

Details

See Boulesteix et al (2008).

Value

A list with the elements:

prediction	A numeric vector of length <code>nrow(Xtest)</code> giving the predicted class for each observation from the test data set.
importance	The variable importance information output by the function <code>varimp</code> from the package <code>party</code> .
OOB	The out-of-bag error of the constructed forest.

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

See Also

`testclass`, `testclass_simul`, `simulate`, `plsr_x_pv`, `plsr_xz_pv`, `plsr_x`, `plsr_xz`, `logistic_z`, `svm_x`.

Examples

```
# load MAclinical library
# library(MAclinical)

# Generating zlearn, ylearn, ztest
zlearn<-matrix(rnorm(120), 30, 4)
ylearn<-sample(0:1, 30, replace=TRUE)
ztest<-matrix(rnorm(80), 20, 4)

my.prediction<-rf_z(Zlearn=zlearn, Ylearn=ylearn, Ztest=ztest)
my.prediction
```

simulate

Simulating data

Description

This function simulates a list of data sets as described in Boulesteix et al (2008), section 3.1.

Usage

```
simuldata_list(niter=50, n=500, p=1000, psig=50, q=5, muX=0, muZ=0)
simuldatacluster_list(niter=50, n=500, p=1000, psig=50, q=5, muX=0, muZ=0)
```

Arguments

<code>niter</code>	The number of data sets to be simulated.
<code>n</code>	The number of observations.
<code>p</code>	The number of microarray variables (genes).
<code>psig</code>	The number of significant microarray variables (must be $< p$).
<code>q</code>	The number of clinical variables.
<code>muX</code>	The class mean difference for the <code>psig</code> relevant genes.
<code>muZ</code>	The class mean difference for the <code>q</code> clinical variables.

Details

With the function `simuldata_cluster`, observations with $y=1$ are assumed to come from two different subgroups, 1a and 1b, each with probability 0.5. Relevant genes are generated such that they separate class 1a from the rest, whereas clinical variables separate class 1b from the rest.

Value

A `niter`-list of simulated data sets. Each data set is given as a list with three elements:

<code>y</code>	the n -vector of class memberships, coded as 0,1.
<code>x</code>	the $n \times p$ matrix of gene expressions levels. Each row corresponds to an observation, each column to a variable (gene).
<code>z</code>	the $n \times q$ matrix of clinical variables. Each row corresponds to an observation, each column to a clinical variable.

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

See Also

`testclass`, `testclass_simul`, `plsr_x_pv`, `plsr_xz_pv`, `plsr_x`, `plsr_xz`, `logistic_z`, `rf_z`, `svm_x`.

Examples

```
# load MAclinical library
# library(MAclinical)

# Generating 3 simulated data sets
my.data<-simuldata_list(niter=3,n=100,p=150,psig=10,q=5,muX=2,muZ=1)
length(my.data)
dim(my.data[[1]]$x)
dim(my.data[[1]]$z)
length(my.data[[1]]$y)
```

svm_x	<i>Classification based on support vector machines using microarray data only</i>
-------	---

Description

This function builds a prediction rule based on the learning data (microarray predictors only) and applies it to the test data. Prediction is performed based on support vector machines. The function `svm_x` uses the function `svm` from the package `e1071`.

Usage

```
svm_x(Xlearn, Zlearn=NULL, Ylearn, Xtest, Ztest=NULL, ordered=NULL, nbgene=NULL, ...)
```

Arguments

<code>Xlearn</code>	A <code>nlearn x p</code> matrix giving the microarray predictors for the learning data set.
<code>Zlearn</code>	A <code>nlearn x q</code> matrix giving the clinical predictors for the learning data set. This argument is ignored.
<code>Ylearn</code>	A numeric vector of length <code>nlearn</code> giving the class membership of the learning observations, coded as 0,1.
<code>Xtest</code>	A <code>ntest x p</code> matrix giving the microarray predictors for the test data set.
<code>Ztest</code>	A <code>ntest x q</code> matrix giving the clinical predictors for the test data set. This argument is ignored.
<code>ordered</code>	A vector of length <code>p</code> giving the order of the microarray predictors in terms of relevance for prediction. For instance, if the three first elements of <code>ordered</code> are 30,2,2400, it means that the most relevant genes are the genes in the 30th, 2nd and 2400th columns of the gene expression data matrix <code>Xlearn</code> . Note: if <code>ordered=NULL</code> , the columns of <code>Xlearn</code> and <code>Xtest</code> are assumed to be already ordered.
<code>nbgene</code>	The number of genes to be selected for use in dimension reduction. Default is <code>nbgene=NULL</code> , in which case all genes are used.
<code>...</code>	Other arguments to be passed to the function <code>svm</code> from the <code>e1071</code> package.

Details

This function is included in the package for comparison.

Value

A list with the element

<code>prediction</code>	A numeric vector of length <code>nrow(Xtest)</code> giving the predicted class for each observation from the test data set.
-------------------------	---

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

See Also

[testclass](#), [testclass_simul](#), [simulate](#), [plsr_x_pv](#), [plsr_xz](#), [plsr_xz_pv](#), [rf_z](#), [logistic_z](#).

Examples

```
# load MAclinical library
# library(MAclinical)

# Generating xlearn, zlearn, ylearn, xtest, ztest
xlearn<-matrix(rnorm(3000), 30, 100)
ylearn<-sample(0:1, 30, replace=TRUE)
xtest<-matrix(rnorm(2000), 20, 100)

my.prediction1<-svm_x(Xlearn=xlearn, Ylearn=ylearn, Xtest=xtest)

ordered<-sample(100)
my.prediction2<-svm_x(Xlearn=xlearn, Ylearn=ylearn, Xtest=xtest, ordered=ordered, nbgene=20)
```

testclass

Evaluating a classification method based on several learning subsets

Description

This function evaluates classifiers built using microarray data and/or clinical predictors, based on several pairs of learning and test data sets.

Usage

```
testclass(x=NULL, y, z=NULL, learningsets, classifier, ncomp=0:3,
varsel=NULL, nbgene=NULL, fold=10, ...)
```

Arguments

<code>x</code>	A $n \times p$ matrix giving the gene expression levels of p genes (columns) for n patients (rows).
<code>y</code>	A numeric vector of length n giving the class membership of the n patients, coded as $0, \dots, K-1$ (where K is the number of classes).
<code>z</code>	A $n \times q$ data frame giving the q clinical predictors for the n patients. Nominal variables should be given as factors, variables with an at least ordinal scale should be given as numeric.
<code>learningsets</code>	A matrix with <code>niter</code> rows giving the indices of the arrays to be included in the learning sets for the <code>niter</code> iterations, as generated by the function <code>generate.learningsets</code> . The i -th row gives the indices of the arrays included in the learning set for the i -th iteration. For instance, in LOOCV, the i -th row of the matrix <code>learningsets</code> contains all the integers from 1 to n except i . Note that an observation may be included twice or more in the same learning set (for instance in bootstrap sampling).
<code>classifier</code>	The function used to construct a classifier. The function must have the same structure as <code>plsrf_xz_pv</code> .
<code>ncomp</code>	The candidate numbers of PLS components (if PLS dimension reduction is used).
<code>varsel</code>	A <code>niter</code> \times p matrix giving the indices of the genes ordered by the chosen gene selection criterion. For example, the element in the first row and the first column is the index of the gene that is ranked best using the first learning set.
<code>nbgene</code>	The number of genes to use for classifier construction. Default is <code>nbgene=NULL</code> , corresponding to all genes.
<code>fold</code>	The number of folds for the pre-validation step. See Boulesteix et al (2008) for more details. Default is <code>fold=10</code> .
<code>...</code>	Other arguments to be passed to the function <code>cforest_control</code> from the <code>party</code> package or to the function <code>svm</code> from the package <code>e1071</code> , depending on the specified classifier.

Details

For an overview of different methods used to generate the learning sets defined by `generate.learningsets`, see Boulesteix et al (2007). These methods include (repeated) cross-validation, subsampling, bootstrap sampling.

Value

<code>error</code>	A numeric vector of length <code>niter</code> giving the misclassification rate for each iteration.
<code>bestncomp</code>	A numeric vector of length <code>niter</code> giving the best number of (pre-validated) PLS components, as obtained using the model selection method based on the out-of-bag error by Boulesteix et al (returned only for the classifiers <code>plsrf_xz_pv</code> , <code>plsrf_xz</code> , <code>plsrf_x_pv</code> , <code>plsrf_x</code>).

OOB A list of length `niter`, whose elements are numeric vectors of the same length as `ncomp` giving the out-of-bag error of the forest constructed with the corresponding number of (pre-validated) PLS components (returned only for the classifiers `plsrfr_xz_pv`, `plsrfr_xz`, `plsrfr_x_pv`, `plsrfr_x`, `rf_z`. For `rf_z`, no model selection is performed: OOB is just the out-of-bag error of the constructed forest.)

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

Boulesteix AL, Strobl C, Augustin T, Daumer D, 2008. Evaluating microarray-based classifiers: an overview. *Cancer Informatics* 6:77-97.

See Also

[testclass_simul](#), [simulate](#), [generate.learningsets](#), [plsrfr_xz_pv](#), [plsrfr_x_pv](#), [plsrfr_xz](#), [plsrfr_x](#), [rf_z](#), [svm_x](#), [logistic_z](#).

Examples

```
# load MAclinical library
# library(MAclinical)

# Generate data
x<-matrix(rnorm(20000),100,200)
z<-matrix(rnorm(500),100,5)
y<-sample(0:1,100,replace=TRUE)

# Generate learningsets (5-fold CV)
my.learningsets<-generate.learningsets(n=100,method="CV",fold=5)

# Evaluate accuracy of the PLS-PV-RF method
my.eval<-testclass(x=x,y=y,z=z,learningsets=my.learningsets,classifier=plsrfr_xz_pv,ncomp=5,
varsel=NULL,nbgene=NULL,fold=10)

# With variable selection
my.varsel<-matrix(0,5,200)
for (i in 1:5)
{
  my.varsel[i,]<-order(abs(studentt.stat(X=x[my.learningsets[i,],],
L=y[my.learningsets[i,]]+1)),decreasing=TRUE)
}

my.eval<-testclass(x=x,y=y,z=z,learningsets=my.learningsets,classifier=plsrfr_xz_pv,ncomp=5,
varsel=my.varsel,nbgene=15,fold=10)
```

testclass_simul *Evaluating a classification method based on simulated data*

Description

This function evaluates classifiers built using microarray data and/or clinical predictors, based on simulated data generated using the functions `simuldata_list` and `simuldatacluster_list` (see [simulate](#)).

Usage

```
testclass_simul(datalist, nlearn=100, classifier, ncomp=0:3, nbgene=NULL,
varsel=NULL, fold=10, ...)
```

Arguments

<code>datalist</code>	A list of <code>niter</code> simulated data sets as generated by the functions <code>simuldata_list</code> and <code>simuldatacluster_list</code> (see simulate).
<code>nlearn</code>	The number of observations to be included in the learning data set. It must be smaller than the total number of observations of the data sets.
<code>classifier</code>	The function used to construct a classifier. The function must have the same structure as <code>plsrf_xz_pv</code> .
<code>ncomp</code>	The candidate numbers of PLS components (if PLS dimension reduction is used).
<code>nbgene</code>	The number of genes to use for classifier construction. Default is <code>nbgene=NULL</code> , corresponding to all genes.
<code>varsel</code>	A <code>niter</code> x <code>p</code> matrix giving the indices of the genes ordered by the chosen gene selection criterion. For example, the element in the first row and the first column is the index of the gene that is ranked best using in the first simulation iteration.
<code>fold</code>	The number of folds for the pre-validation step, if any. See Boulesteix et al (2008) for more details. Default is <code>fold=10</code> .
<code>...</code>	Other arguments to be passed to the function <code>cforest_control</code> from the <code>party</code> package or to the function <code>svm</code> from the package <code>e1071</code> , depending on the specified classifier.

Details

See Boulesteix et al (2008).

Value

error	A numeric vector of length <code>niter</code> giving the misclassification rate for each iteration.
bestncomp	A numeric vector of length <code>niter</code> giving the best number of (pre-validated) PLS components, as obtained using the model selection method based on the out-of-bag error by Boulesteix et al (returned only for the classifiers <code>plsr_xz_pv</code> , <code>plsr_xz</code> , <code>plsr_x_pv</code> , <code>plsr_x</code>).
OOB	A list of length <code>niter</code> , whose elements are numeric vectors of the same length as <code>ncomp</code> giving the out-of-bag error of the forest constructed with the corresponding number of (pre-validated) PLS components (returned only for the classifiers <code>plsr_xz_pv</code> , <code>plsr_xz</code> , <code>plsr_x_pv</code> , <code>plsr_x</code> , <code>rf_z</code> . For <code>rf_z</code> , no model selection is performed: OOB is just the out-of-bag error of the constructed forest.)

Author(s)

Anne-Laure Boulesteix (http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/020_professuren/boulesteix/)

References

Boulesteix AL, Porzelius C, Daumer M, 2008. Microarray-based classification and clinical predictors: On combined classifiers and additional predictive value. *Bioinformatics* 24:1698-1706.

See Also

`testclass`, `plsr_xz_pv`, `simulate`, `plsr_xz_pv`, `plsr_x_pv`, `plsr_xz`, `plsr_x`, `rf_z`, `svm_x`, `logistic_z`.

Examples

```
# load MAclinical library
# library(MAclinical)

# Generating 3 simulated data sets
my.data<-simuldata_list(niter=3,n=100,p=150,psig=10,q=5,muX=2,muZ=1)

# Perform prediction of the 60 last observations using the first 40 observations,
# based on PLS (without pre-validation) and random forests

testclass_simul(my.data,nlearn=40,classifier=plsr_xz)
```

Index

*Topic **htest**

- logistic_z, 3
- plsr_x, 5
- plsr_x_pv, 11
- plsr_xz, 7
- plsr_xz_pv, 9
- rf_z, 13
- simulate, 14
- svm_x, 16
- testclass_simul, 20

*Topic **multivariate**

- generate.learningsets, 2
- testclass, 17

generate.learningsets, 2, 18, 19

logistic_z, 3, 6, 8, 10, 12, 14, 15, 17, 19, 21

plsr_x, 4, 5, 8, 10, 12, 14, 15, 18, 19, 21
plsr_x_pv, 4, 6, 8, 10, 11, 14, 15, 17–19, 21

plsr_xz, 4, 6, 7, 10, 12, 14, 15, 17–19, 21
plsr_xz_pv, 4, 6, 8, 9, 12, 14, 15, 17–21

rf_z, 4, 6, 8, 10, 12, 13, 15, 17, 19, 21

simulate, 4, 6, 8, 10, 12, 14, 14, 17, 19–21
simuldata_list (simulate), 14
simuldatacluster_list (simulate), 14

svm_x, 4, 6, 8, 10, 12, 14, 15, 16, 19, 21

testclass, 4, 6, 8, 10, 12, 14, 15, 17, 17, 21
testclass_simul, 4, 6, 8, 10, 12, 14, 15, 17, 19, 20