Package 'stepwiseCM'

March 26, 2013

Type Package

Title Stepwise Classification of Cancer Samples using Clinical and Molecular Data

Version 1.4.0

Date 2011-05-16

Author Askar Obulkasim

Maintainer Askar Obulkasim <askar.wubulikasimu@vumc.nl>

Depends R (>= 2.14), randomForest, MAclinical, tspair, pamr, snowfall,glmpath, penalized, e1071

Description Stepwise classification of cancer samples using both clinical and molecular data

License GPL (http://www.gnu.org/copyleft/gpl.html)

R topics documented:

---

stepwiseCM-package *Stepwise classification of cancer samples using both clinical and molecular Data.*

Description

Given two types of data, it first evaluate the classification performances of two data types independently by the user define classification algorithm, then explore the two different data spaces using the proximity matrix from random forest algorithm. Based on the exact locations of the test samples in the clinical data space (presume clinical information of the test samples are given) and the "pseudo" locations in the molecular data space (presume molecular profiles of the test sample are not available), reclassification scores (RS) for each test sample is calculated. Large value of the RS means, sample benefits more if classify it with molecular data.

Details

Package:	stepwiseCM
Type:	Package
Version:	0.99.1
Date:	2011-05-16
License:	GPL (http://www.gnu.org/copyleft/gpl.html)
LazyLoad:	yes

Author(s)

Askar Obulkasim

Maintainer: Askar Obulkasim <askar.wubulikasimu@vumc.nl>

Examples

```
data(CNS)
train.cli <- t(CNS$cli[1:40, ])
test.cli <- t(CNS$cli[41:60, ])
train.gen <- CNS$mrna[, 1:40]
test.gen <- CNS$mrna[, 41:60]
train.label <- CNS$class[1:40]
test.label <- CNS$class[41:60]
result <- Curve.generator(train.cli, train.gen, train.label, test.cli, test.gen, test.label,
            type = c("GLM_L1", "GLM_L2"), RStype = "both", Parallel = FALSE,
            CVtype = "k-fold", outerkfold = 2, innerkfold = 2, N = 2, plot.it =FALSE)
names(result)
```

Classifier *A function to predict the class labels of the test set.*

Description

Given the training set and the type of classification algorithm, this function constructs the classification model based on the training set and predict the class labels of the test set.

Usage

```
Classifier(train, test = c(), train.label, type = c("TSP", "GLM", "GLM_L1",
        "GLM_L2", "PAM", "SVM", "plsrf_x", "plsrf_x_pv", "RF"),
        CVtype = c("loocv", "k-fold"), outerkfold = 5, innerkfold = 5,
        featurenames = NULL)
```

Arguments

train	A data frame or matrix of containing predictors for the training set, where columns correspond to samples and rows to features.
test	A data frame or matrix containing predictors for the test set (optional), where columns correspond to samples and rows to features.
train.label	A vector of the class labels (0 or 1) of the training set. NOTE: class labels should be numerical not factor.
type	Type of classification algorithms. Currently 9 different types of algorithm are available. They are: top scoring pair (TSP), logistic regression (GLM), GLM with L1 (lasso) penalty, GLM with L2 (ridge) penalty, prediction analysis for microarray (PAM), support vector machine (SVM), random forest method after partial least square dimension reduction (plsrf_x), random forest method after partial least square dimension reduction plus prevalidation (plsrf_x_pv), random forest (RF). NOTE: TSP, PAM, plsrf_x and plsrf_x_pv algorithms does not work with clinical data.
CVtype	Cross validation type.
outerkfold	Number of cross validation used in the training phase.
innerkfold	Number of cross validation used to estimate the model parameters.
featurenames	Feature names in molecular data (e.g. gene or probe names). If given, function also produces name of the selected feature during the training and test phases. Feature selection only works with "TSP", "GLM_L1" and "GLM_L2" algorithms. "RF" provides feature importance.

Value

A list object *Pred* which contains following components:

P.train	predicted class labels of the training set.
P.test	predicted class labels of the test set if the test set is given.
selfeatname_tr	A list object, size of *outerkfold*, containing the name of the selected features during the training phase if the *featurenames* is given.
selfeatname_te	A list object containing the name of the selected features during the test phase if the *test* and *featurenames* are given.

Author(s)

Askar Obulkasim

Maintainer: Askar Obulkasim <askar.wubulikasimu@vumc.nl>

References

Aik Choo Tan and Daniel Q. Naiman and Lei Xu and Raimond L. Winslow and Donald Geman(2005). Simple Decision Rules for Classifying Human Cancers from Gene Expression Profiles(TSP). *Bioinformatics, 21*, 3896-3904.

Anne-Laure Boulesteix and Christine Porzelius and Martin Daumer(2008). Microarray-based Classification and Clinical Predictors: on Combined Classifiers and Additional Predictive Value. *Bioinformatics, 24*, 1698–1706.

See Also

Classifier.par

Examples

```
data(CNS)
train <- CNS$mrna[, 1:40]
test <- CNS$mrna[, 41:60]
train.label <- CNS$class[1:40]
Pred <- Classifier(train = train, test = test, train.label = train.label,
      type = "GLM_L1", CVtype = "k-fold", outerkfold = 2, innerkfold = 2)
Pred$P.train
Pred$P.test
```

Classifier.par — *A function to obtain the predicted class labels of the test set using the parallel processing procedure.*

Description

Classification has been embedded inside the parallel processing procedure to speed up the computation for large dataset.

Usage

```
Classifier.par(train, test = c(), train.label, type = c("TSP", "GLM", "GLM_L1",
            "GLM_L2", "PAM", "SVM", "plsrf_x", "plsrf_x_pv", "RF"),
            CVtype = c("loocv", "k-fold"), outerkfold = 5, innerkfold = 5,
            featurenames = NULL, ncpus = 2)
```

Arguments

train	A data frame or matrix of containing predictors for the training set, where columns correspond to samples and rows to features.
test	A data frame or matrix containing predictors for the test set (optional), where columns correspond to samples and rows to features.
train.label	A vector of the class labels (0 or 1) of the training set. NOTE: response values should be numerical not factor.
type	Type of classification algorithms. Currently 9 different types of algorithm are available. They are: top scoring pair (TSP), logistic regression (GLM), GLM with L1 (lasso) penalty, GLM with L2 (ridge) penalty, prediction analysis for microarray (PAM), support vector machine (SVM), random forest method after partial least square dimension reduction (plsrf_x), random forest method after partial least square dimension reduction plus prevalidation (plsrf_x_pv), random forest (RF). NOTE: TSP, PAM, plsrf_x and plsrf_x_pv algorithms does not work with clinical data.
CVtype	Cross validation type.
outerkfold	Number of cross validation used in the training phase.
innerkfold	Number of cross validation used to estimate the model parameters.

featurenames	Feature names in molecular data (e.g. gene or probe names). If given, function also produces name of the selected feature during the training and test phases. Feature selection only works with "TSP", "GLM_L1" and "GLM_L2" algorithms. "RF" provide feature importance.
ncpus	Number of CPUs assign to the parallel computation.

Value

A list object *Pred* which contains following components:

P.train	A vector of the predicted class labels of the training set.
P.test	A vector of the predicted class labels of the test set if the test set is given.
selfeatname_tr	A list object, size of *outerkfold*, containing the name of the selected features during the training phase if the *featurenames* is given.
selfeatname_te	A list object containing the name of the selected features during the test phase if the *test* and *featurenames* are given.

Author(s)

Askar Obulkasim

Maintainer: Askar Obulkasim <askar.wubulikasimu@vumc.nl>

References

Aik Choo Tan and Daniel Q. Naiman and Lei Xu and Raimond L. Winslow and Donald Geman(2005). Simple Decision Rules for Classifying Human Cancers from Gene Expression Profiles(TSP). *Bioinformatics, 21*, 3896-3904.

Anne-Laure Boulesteix and Christine Porzelius and Martin Daumer(2008). Microarray-based Classification and Clinical Predictors: on Combined Classifiers and Additional Predictive Value. *Bioinformatics, 24*, 1698–1706.

Examples

```
data(CNS)
train <- CNS$mrna[, 1:40]
test <- CNS$mrna[, 41:60]
train.label <- CNS$class[1:40]
## Not run: Pred <- Classifier.par(train = train, test = test, train.label = train.label, type = "GLM_L1",
                CVtype = "k-fold", outerkfold = 5, innerkfold = 5, ncpus = 5)
## End(Not run)
```

CNS *Central Nervous System (CNS) cancer data set.*

Description

A list object includes the measurement of the gene expression in 7128 genes and 60 samples and the clinical risk factors which are Chang stage (nominal), sex (binary), age (nominal), chemo Cx (binary), chemo VP (binary). 21 patients died (labeled 0) and 39 patients survived (labeled 1) within 24 months after the treatment.

References

Pomeroy, SL. and Tamayo, P. and Gaasenbeek, M. and Sturla, LM. and Angelo, M. and McLaughlin, ME. and Kim, JY. and Goumnerova, LC. and Black, PM. and Lau, C. and Allen, JC. and Zagzag, D. and Olson, JM. and Curran, T. and Wetmore, C. and Biegel, JA. and Poggio, T. and Mukherjee, S. and Rifkin, R. and Califano, A. and Stolovitzky, G. and Louis, DN. and Mesirov, JP. and Lander, ES. and Golub, TR. Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression. *Nature*., 415(6870):436-442.

Examples

```
data(CNS)
names(CNS)
```

Curve.generator *A function to generate accuracy with different cut points .*

Description

For given clinical and molecular data, this function first calculates the predicted class labels of the training sets and the proximity matrices using clinical and molecular data separately. In the second step based on the classification performances of the two data types and the sample distribution in the two spaces, it calculates the reclassification score for the test set. Then, produce a vector accuracies by passing different percentage of samples to molecular data. This curve can be used as a reference for choosing the RS threshold for incoming test samples.

Usage

```
Curve.generator(train.cli, train.gen, train.label, test.cli, test.gen,
    test.label, type = c("TSP", "GLM", "GLM_L1", "GLM_L2", "PAM", "SVM",
    "plsrf_x", "plsrf_x_pv", "RF"), RStype = c("rank", "proximity", "both"),
    Parallel = FALSE, CVtype = c("loocv", "k-fold"), outerkfold = 5,
    innerkfold = 5, ncpus = 2, N = 50, featurenames = NULL, plot.it = TRUE)
```

Arguments

train.cli	A data frame or matrix of containing predictors of the training set from clinical data, where columns correspond to samples and rows to features.
train.gen	A data frame or matrix of containing predictors of the training set from molecular data, where columns correspond to samples and rows to features.
train.label	A vector of the class labels (0 or 1) of the training set. NOTE: response values should be numeric not factor.
test.cli	A data frame or matrix of containing predictors of the test set from clinical data, where columns correspond to samples and rows to features.
test.gen	A data frame or matrix containing predictors of the test set from genomic data, where columns correspond to samples and rows to features.
test.label	A vector of the class labels (0 or 1) of the test set (optional). NOTE: response values should be numeric not factor.

type	Type of classification algorithms. Currently 9 different types of algorithm are available. They are: top scoring pair (TSP), logistic regression (GLM), GLM with L1 (lasso) penalty, GLM with L2 (ridge) penalty, prediction analysis for microarray (PAM), support vector machine (SVM), random forest method after partial least square dimension reduction (plsrf_x), random forest method after partial least square dimension reduction plus prevalidation (plsrf_x_pv), random forest (RF). NOTE: TSP, PAM, plsrf_x and plsrf_x_pv algorithms does not work with clinical data.
RStype	Which values are used to calculate the reclassification score (RS)? There are three options available: *proximity*, *rank* and *both*. If set to *proximity*, RS will be calculated directly from the proximity value. If set to *rank*, RS calculate based on the rank of the proximity values (more robust). If set to *both*, both of them will be calculated. Default is rank.
Parallel	Should class prediction and proximity calculation use the parallel processing procedure? Default is FALSE.
CVtype	Cross validation type.
outerkfold	Number of cross validation used in the training phase.
innerkfold	Number of cross validation used to estimate the model parameters.
ncpus	Number of CPUs assign to the parallel computation.
N	Number of repetition for calculating the proximity matrix, final proximity matrix is average of these repeats. We recommend setting this number high, so that more stable proximity matrix will be produced. Default is 50.
featurenames	Feature names in molecular data (e.g. gene or probe names). If given, function also produces name of the selected feature during the training and test phases. Feature selection only works with "TSP", "GLM_L1" and "GLM_L2" algorithms. "RF" provides feature importance.
plot.it	If set to TRUE, this function produces a plot in which Y axis denotes the accuracy and X denotes the percentage of samples passed to the second stage. In order to make this plot, class labels and molecular data for the test set must be given. Default is TRUE.

Details

This function requires the molecular information for a group of samples (called test here) which are not used in the training phase. Based on their RS a accuracy curve will be attained by setting different thresholds. If the option *type*, has length 2 (two classification algorithm types are given), then first one used for the prediction using clinical data and second one used for the prediction using molecular data. If only one algorithm type is given, the same algorithm used for both data types. Note that, TSP, PAM, plsrf_x and plsrf_x_pv algorithms does not work with clinical data.

Value

A list object with the following components:

Predicted.cli	A list object includes the predicted class labels of the training set and the test set if test is given. This is classification result with clinical data.
Predicted.gen	A list object includes the predicted class labels of the training set and the test set if test is given. This is classification result with molecular data.

Proximity	A list object of proximity matrices which includes the proximity matrix of test samples in the clinical data space (ncol(test.cli) by ncol(train.cli) matrix) and the proximity matrix of the training samples in the genomic data space (ncol(train.gen) by ncol(train.gen) matrix).
RS	If the "type" set to "rank" ("proximity"), it gives a vector of re-classification scores calculated from the ranking (proximity) approach , otherwise it gives a matrix with two columns and size of rows equal number of test samples, calculated using the both approaches.
Accuracy	If *test.gen* is given, accuracies corresponding to different percentage of samples are classified with molecular data are produced. If *RStype* is set to rank or proximity, accuracy will be a vector. If RStype is set to both, accuracy will be a matrix with two columns and eleven rows. First column corresponding to accuracies when RS is calculated using the rank of proximity, second column corresponding to accuracies when RS is calculated using the proximity.
Param	A list object contains the values of parameters specified by user.
Matrices	A list object contains the input data matrices.

Author(s)

Askar Obulkasim

Maintainer: Askar Obulkasim <askar.wubulikasimu@vumc.nl>

See Also

Classifier, Classifier.par, Proximity, RS.generator

Examples

```
data(CNS)
tr.cli <- t(CNS$cli[1:40, ])
te.cli <- t(CNS$cli[41:60, ])
tr.gen <- CNS$mrna[, 1:40]
te.gen <- CNS$mrna[, 41:60]
tr.label <- CNS$class[1:40]
te.label <- CNS$class[41:60]
result <- Curve.generator(train.cli=tr.cli, train.gen=tr.gen, train.label=tr.label, test.cli=te.cli,
                  test.gen=te.gen, test.label=te.label, type = c("GLM_L1", "GLM_L2"),
                  RStype = "rank", Parallel = FALSE, CVtype = "k-fold", outerkfold = 2,
                  innerkfold = 2, N = 2)
names(result)
```

Proximity *A function to calculate proximity matrix.*

Description

A function to compute the proximity matrices using clinical and genomic data separately.

Usage

```
Proximity(train.cli, train.label, test.cli, train.gen, N = 50,
      Parallel = FALSE, ncpus = 2)
```

Arguments

train.cli	A data frame or matrix of containing predictors for the training set, where columns correspond to samples and rows to features.
train.label	A vector of the class labels (0 or 1) of the training set. NOTE: response values should be numeric not factor.
test.cli	A data frame or matrix of containing predictors for the test set, where columns correspond to samples and rows to features.
train.gen	A data frame or matrix of containing predictors of the training set from genomic data, where columns correspond to samples and rows to features.
N	Number of repetition for calculating the proximity matrix, final proximity matrix is average of these repeats. We recommend setting this number high, so that more stable proximity matrix will be produced. Default is 50.
Parallel	Should proximity calculation use the parallel processing procedure? Default is FALSE.
ncpus	Number of CPUs assign to the parallel computation. Default is 2.

Details

Proximity matrix is calculated using the random forest algorithm. Proximity values ranges from 0 (least similar) to 1 (perfect match).

Value

A list object with the following components:

Prox.cli	A matrix, size of ncol(test.cli) by ncol(train.cli), contains the proximity values between the test set and the training set in the clinical data space .
Prox.gen	A square matrix, size of ncol(train.gen), contains the proximity values between training set in the genomic data space.

Author(s)

Askar Obulkasim

Maintainer: Askar Obulkasim <askar.wubulikasimu@vumc.nl>

References

Breiman, L. (2001), *Random Forest, 45*, 5-32.

Examples

```
data(CNS)
train.cli <- t(CNS$cli[1:40,])
test.cli <- t(CNS$cli[41:60,])
train.gen <- CNS$mrna[,1:40]
train.label <- CNS$class[1:40]
##without parallel processing procedure
Prox <- Proximity(train.cli, train.label, test.cli, train.gen, N = 2)
##with parallel processing procedure
## Not run: Prox <- Proximity(train.cli, train.label, test.cli, train.gen,
            N = 50, Parallel = TRUE, ncpus = 10)
## End(Not run)
```

RS.generator *A function to generate reclassification score.*

Description

A function to calculates the reclassification score (RS) using both clinical and molecular data.

Usage

```
RS.generator(pred.cli, pred.gen, train.label, prox.gen, prox.cli,
        type = c("rank", "proximity", "both"))
```

Arguments

pred.cli	A vector containing the predicted labels of training set from clinical data.
pred.gen	A vector containing the predicted labels of training set from genomic data.
train.label	A vector of the class labels (0 or 1) of the training set. NOTE: response values should be numerical not factor.
prox.gen	A square matrix, size of ncol(train.gen), contains the proximity values between training set in the genomic data space.
prox.cli	A matrix, size of ncol(test.cli) by ncol(train.cli), contains the proximity values between the test set and the training set in the clinical data space.
type	Which values are used to construct the reclassification score (RS)? There are three options available: *proximity*, *rank* and *both*. If set to proximity, RS will be calculated directly from the proximity value. If set to rank, RS calculate from the rank of proximity values (more robust). If set to *both*, both of them will be calculated. Default is *rank*.

Details

For each test sample, RS is calculated using the given classification results from clinical and genomic data. Algorithm project each test sample onto the clinical space check its neighborhood, also tries to gain some information about this test samples "pseudo" neighbors in the genomic space by the indirect mapping. If algorithm finds that the location of this test samples in the clinical space are more "safe" (more neighbors are correctly classified) and the location in the genomic space is surrounded by wrongly classified samples, then it will give this test sample high RS score and vice versa. After obtaining the RS, user can order them in descending order and pass the top ranked certain portion (decided by user) of samples to genomic data to classify.

Value

If the "type" set to "rank" ("proximity"), then RS will be a vector of RS calculated from the ranking (proximity) approach , otherwise RS will be a matrix of RS, with two columns and size of rows equal number of test samples, calculated using the both approaches.

Author(s)

Askar Obulkasim

Maintainer: Askar Obulkasim <askar.wubulikasimu@vumc.nl>

Examples

```
data(CNS)
train.cli <- t(CNS$cli[1:40,])
test.cli <- t(CNS$cli[41:60,])
train.gen <- CNS$mrna[,1:40]
train.label <- CNS$class[1:40]
pred.cli <- Classifier(train = train.cli, train.label = train.label, type = "GLM_L1",
        CVtype = "k-fold", outerkfold = 2, innerkfold = 2)
pred.gen <- Classifier(train = train.gen, train.label = train.label, type = "GLM_L1",
        CVtype = "k-fold", outerkfold = 2, innerkfold = 2)
prox <- Proximity(train.cli, train.label, test.cli, train.gen, N = 2)
RS.generator(pred.cli$P.train, pred.gen$P.train, train.label, prox$Prox.gen,
         prox$Prox.cli, type = "both")
```

Step.pred *A function to calculate the RS for the test set and decide which samples should be classified with molecular data.*

Description

This function needs the output from the "Curve.generator" and the percentage of samples allows to classify with molecular data. Based on the specified percentage, it finds the RS threshold. Compare a test sample's RS with this threshold, decide whether a test samples should be classified with molecular data or not.

Usage

```
Step.pred(curve, test.cli, percent)
```

Arguments

curve	A list obejct generated by the "Curve.generator" function.
test.cli	A data frame or matrix containing the clinical variables of the test set, where columns correspond to samples and rows to features.
percent	percentage of samples allow to classify with molecular data

Value

Pred	predicted class labels of the test set
RS	If the "type" set to "rank" ("proximity") was used in generating the "curve" object, then RS will be a vector of RS calculated from the ranking (proximity) approach , otherwise RS will be a matrix of RS, with two columns and size of rows equal number of test samples, calculated using the both approaches.
Threshold	a RS which corresponding to the specified re-classification percentage.
Pass	a vector of binary values. 1 means samples's RS higher than the threshold, so classify it with molecular data is more beneficial and vice versa.

Author(s)

Askar Obulkasim

Maintainer: Askar Obulkasim <askar.wubulikasimu@vumc.nl>

Examples

```
data(CNS)
tr.cli <- t(CNS$cli[1:40, ])
te.cli <- t(CNS$cli[41:60, ])
tr.gen <- CNS$mrna[, 1:40]
te.gen <- CNS$mrna[, 41:60]
tr.label <- CNS$class[1:40]
te.label <- CNS$class[41:60]
curve <- Curve.generator(train.cli=tr.cli, train.gen=tr.gen, train.label=tr.label, test.cli=te.cli,
                         test.gen=te.gen, test.label=te.label, type = c("RF", "GLM_L1"),
                         RStype = "rank", Parallel = FALSE, CVtype = "k-fold", outerkfold = 2,
                         innerkfold = 2, N = 2, plot.it = FALSE)
A <- Step.pred(curve, te.cli, 30)
```

Index