# Package 'packMBPLSDA'

July 23, 2025

**Type** Package

**Title** Multi-Block Partial Least Squares Discriminant Analysis

**Version** 0.9.0

**Date** 2022-06-20

**Author** Marion Brandolini-Bunlon, Stephanie Bougeard, Melanie Petera, Estelle Pujos-Guillot

**Maintainer** Marion Brandolini-Bunlon <marion.brandolini-bunlon@inra.fr>

**Description** Several functions are provided to implement a MBPLSDA : components search, optimal model components number search, optimal model validity test by permutation tests, observed values evaluation of optimal model parameters and predicted categories, bootstrap values evaluation of optimal model parameters and predicted cross-validated categories. The use of this package is described in Brandolini-Bunlon et al (2019. Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134).

**License** GPL (>= 2.0)

**Depends** ade4, pROC

**Imports** MASS, parallel, doParallel, foreach, FactoMineR

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-06-20 16:00:02 UTC

## Contents

---

packMBPLSDA-package *Multi-Block Partial Least Squares Discriminant Analysis*

---

## Description

Several functions are provided to implement a MBPLSDA : components search, optimal model components number search, optimal model validity test by permutation tests, observed values evaluation of optimal model parameters and predicted categories, bootstrap values evaluation of optimal model parameters and predicted cross-validated categories. The use of this package is described in Brandolini-Bunlon et al (2019. Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134).

## Details

Index of help topics:

| | |
|---|---|
| boot_mbplsda | bootstraped simulations for multi-block partial least squares discriminant analysis |
| cvpred_mbplsda | Cross-validated predicted categories from a multi-block partial least squares discriminant model |
| disjunctive | Disjunctive table |
| ginv | generalized inverse of a matrix X |
| inertie | inertia of a matrix |
| mbplsda | Multi-block partial least squares discriminant analysis |
| medical | medical dataset |
| nutrition | nutritional dataset |
| omics | metabolomic dataset |
| packMBPLSDA-package | Multi-Block Partial Least Squares Discriminant Analysis |
| permut_mbplsda | Permutation testing of a multi-block partial least squares discriminant model |
| plot_boot_mbplsda | Plot the results of the fonction boot_mbplsda in a pdf file |
| plot_cvpred_mbplsda | Plot the results of the fonction cvpred_mbplsda |

| | |
|---|---|
| | in a pdf file |
| plot_permut_mbplsda | Plot the results of the fonction permut_mbplsda in a pdf file |
| plot_pred_mbplsda | Plot the results of the fonction pred_mbplsda in a pdf file |
| plot_testdim_mbplsda | Plot the results of the fonction testdim_mbplsda in a pdf file |
| pred_mbplsda | Observed parameters and predicted categories from a multi-block partial least squares discriminant model |
| status | physiopathological status data |
| testdim_mbplsda | Test of number of components by two-fold cross-validation for a multi-block partial least squares discriminant model |

## Author(s)

Marion Brandolini-Bunlon, Stephanie Bougeard, Melanie Petera, Estelle Pujos-Guillot

Maintainer: Marion Brandolini-Bunlon <marion.brandolini-bunlon@inra.fr>

## References

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

mbplsda testdim_mbplsda plot_testdim_mbplsda permut_mbplsda plot_permut_mbplsda pred_mbplsda plot_pred_mbplsda cvpred_mbplsda plot_cvpred_mbplsda boot_mbplsda plot_boot_mbplsda

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical, nutrition = nutrition, omics = omics))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 2)
```

`boot_mbplsda` *bootstraped simulations for multi-block partial least squares discriminant analysis*

## Description

Function to perform bootstraped simulations for multi-block partial least squares discriminant analysis, in order to get confidence intervals for regression coefficients, variable loadings, variable and block importances.

## Usage

```
boot_mbplsda(object, nrepet = 199, optdim, cpus = 1, ...)
```

## Arguments

| | |
|---|---|
| `object` | an object created by mbplsda |
| `nrepet` | integer indicating the number of repetitions |
| `optdim` | integer indicating the optimal number of global components to be introduced in the model |
| `cpus` | integer indicating the number of cpus to use when running the code in parallel |
| `...` | other arguments to be passed to methods |

## Details

no details are needed

## Value

| | |
|---|---|
| `XYcoef` | mean, standard deviation, quantiles (0.025;0.975), 95% confidence interval, median for regression coefficients |
| `faX` | mean, standard deviation, quantiles (0.025;0.975), 95% confidence interval, median for variable loadings |
| `vipc` | mean, standard deviation, quantiles (0.025;0.975), 95% confidence interval, median for cumulated variable importances |
| `bipc` | mean, standard deviation, quantiles (0.025;0.975), 95% confidence interval, median for cumulated block importances |

## Note

at least 30 bootstrap repetitions may be recommended, more than 100 beeing preferable

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Efron, B., Tibshirani, R.J. (1994). An Introduction to the Bootstrap. Chapman and Hall-CRC Monographs on Statistics and Applied Probability, Norwell, Massachusetts, United States.

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

mbplsda plot_boot_mbplsda packMBPLSDA-package

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical, nutrition = nutrition, omics = omics))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
ncpopt <- 1
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 2)
resboot <- boot_mbplsda(modelembplsQ, optdim = ncpopt, nrepet = 30, cpus=1)
```

---

# cvpred_mbplsda *Cross-validated predicted categories from a multi-block partial least squares discriminant model*

---

## Description

Function to perform 2-fold cross-validation for multi-block partial least squares discriminant analysis, in order to get for each observation the cross-validated predicted categories, and the statistical description of the predictions (mean, sd, 95

## Usage

```
cvpred_mbplsda(object, nrepet = 100, threshold = 0.5, bloY, optdim, cpus = 1,
algo = c("max", "gravity", "threshold"))
```

## Arguments

| | |
|---|---|
| object | an object created by mbplsda |
| nrepet | integer indicating the number of repetitions |
| threshold | numeric indicating the threshold, between 0 and 1, to consider the categories are predicted with the threshold prediction method. |
| bloY | integer vector indicating the number of categories per variable of the Y-block. |
| optdim | integer indicating the (optimal) number of components of the multi-block partial least squares discriminant model |
| cpus | integer indicating the number of cpus to use when running the code in parallel |
| algo | character vector indicating the method(s) of prediction to use (see details) |

## Details

Three different algorithms are available to predict the categories of observations. In the max, and respectively the threshold algorithms, numeric values are calculated from the matrix of explanatory variables and the regression coefficients. Then, the predicted categorie for each variable of the Y-block is the one which corresponds to the higher predicted value, respectively to the values higher than the indicated threshold. In the gravity algorithm, predicted scores of the observations on the components are calculated. Then, each observation is assigned to the observed category of which it is closest to the barycentre in the component space.

## Value

| | |
|---|---|
| TRUEnrepet | number of repetitions |
| matPredYc.max | with the max algorithm, boolean matrix indicating the cross-validated predicted categories on the calibration datasets, the prediction accuracy for each categorie, each Y-block variable, and overall |
| matPredYv.max | with the max algorithm, boolean matrix indicating the cross-validated predicted categories on the validation datasets, the prediction accuracy for each categorie, each Y-block variable, and overall |
| matPredYc.gravity | with the gravity algorithm, boolean matrix indicating the cross-validated predicted categories on the calibration datasets, the prediction accuracy for each categorie, each Y-block variable, and overall |
| matPredYv.gravity | with the gravity algorithm, boolean matrix indicating the cross-validated predicted categories on the validation datasets, the prediction accuracy for each categorie, each Y-block variable, and overall |
| matPredYc.threshold | with the threshold algorithm, boolean matrix indicating the cross-validated predicted categories on the calibration datasets, the prediction accuracy for each categorie, each Y-block variable, and overall |
| matPredYv.threshold | with the threshold algorithm, boolean matrix indicating the cross-validated predicted categories on the validation datasets, the prediction accuracy for each categorie, each Y-block variable, and overall |

`statPredYc.max` with the max algorithm, matrix indicating the statistical description of prediction categories for each observation on the calibration datasets: number of predictions as an observation of the calibration dataset, modal value, probability to be predicted with its standard deviation, 95% confidence interval, quantiles 0.025 and 0.975, median value

`statPredYv.max` with the max algorithm, matrix indicating the statistical description of prediction categories for each observation on the validation datasets: number of predictions as an observation of the validation dataset, modal value, probability to be predicted with its standard deviation, 95% confidence interval, quantiles 0.025 and 0.975, median value

`statPredYc.gravity`
with the gravity algorithm, matrix indicating the statistical description of prediction categories for each observation on the calibration datasets: number of predictions as an observation of the calibration dataset, modal value, probability to be predicted with its standard deviation, 95% confidence interval, quantiles 0.025 and 0.975, median value

`statPredYv.gravity`
with the gravity algorithm, matrix indicating the statistical description of prediction categories for each observation on the validation datasets: number of predictions as an observation of the validation dataset, modal value, probability to be predicted with its standard deviation, 95% confidence interval, quantiles 0.025 and 0.975, median value

`statPredYc.threshold`
with the threshold algorithm, matrix indicating the statistical description of prediction categories for each observation on the calibration datasets: number of predictions as an observation of the calibration dataset, modal value, probability to be predicted with its standard deviation, 95% confidence interval, quantiles 0.025 and 0.975, median value

`statPredYv.threshold`
with the threshold algorithm, matrix indicating the statistical description of prediction categories for each observation on the validation datasets: number of predictions as an observation of the validation dataset, modal value, probability to be predicted with its standard deviation, 95% confidence interval, quantiles 0.025 and 0.975, median value

## Note

at least 90 cross-validation repetitions may be recommended

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society B, 36(2), 111-147.

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

mbplsda plot_cvpred_mbplsda packMBPLSDA-package

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical[,1:10],
nutrition = nutrition[,1:10], omics = omics[,1:20]))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
ncpopt <- 1
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 2)
CVpred <- cvpred_mbplsda(modelembplsQ, nrepet = 30, threshold = 0.5, bloY = bloYobs,
optdim = ncpopt, cpus = 1, algo = c("max"))


data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical,
nutrition = nutrition, omics = omics))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
ncpopt <- 1
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 2)
CVpred <- cvpred_mbplsda(modelembplsQ, nrepet = 90, threshold = 0.5, bloY = bloYobs,
optdim = ncpopt, cpus = 1, algo = c("max"))
```

`disjunctive` *Disjunctive table*

## Description

Function to transform a boolean matrix in a disjunctive table

## Usage

```
disjunctive(y)
```

## Arguments

| | |
|---|---|
| `y` | boolean matrix indicating observations categories |

## Details

no details are needed

## Value

| | |
|---|---|
| `ydisj` | disjunctive table |

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

`packMBPLSDA-package`

## Examples

```
data(status)
disjonctif <- (disjunctive(status))
```

## ginv *generalized inverse of a matrix X*

### Description

function to calculate the generalized inverse of a matrix X

### Usage

```
ginv(X, tol = sqrt(.Machine$double.eps))
```

### Arguments

| | |
|---|---|
| X | Matrix for which the generalized inverse is required |
| tol | A relative tolerance to detect zero singular values |

## inertie *inertia of a matrix*

### Description

function to calculate the inertia of a matrix

### Usage

```
inertie(tab)
```

### Arguments

| | |
|---|---|
| tab | a matrix |

## mbplsda *Multi-block partial least squares discriminant analysis*

### Description

Function to perform a multi-block partial least squares discriminant analysis (MBPLSDA) of several explanatory blocks defined as an object of class ktab, to explain a dependent dataset (Y-block) defined as an object of class dudi, in order to get model parameters for the indicated number of components.

### Usage

```
mbplsda(dudiY, ktabX, scale = TRUE, option = c("uniform", "none"),
scannf = TRUE, nf = 2)
```

## Arguments

| | |
|---|---|
| dudiY | an object of class dudi containing the dependent variables |
| ktabX | an object of class ktab containing the blocks of explanatory variables |
| scale | logical value indicating whether the explanatory variables should be standardized |
| option | option for the block weighting. If uniform, the weight of each explanatory block is equal to 1/number of explanatory blocks, and the weight of the Y-block is eqyual to 1. If none, the block weight is equal to the block inertia. |
| scannf | logical value indicating whether the eigenvalues bar plot should be displayed |
| nf | integer indicating the number of components to be calculated |

## Details

no details are needed

## Value

| | |
|---|---|
| call | the matching call |
| tabX | data frame of explanatory variables centered, eventually scaled (if scale=TRUE)and weighted (if option="uniform") |
| tabY | data frame of dependent variables centered, eventually scaled (if scale=TRUE)and weighted (if option="uniform") |
| nf | integer indicating the number of kept dimensions |
| lw | numeric vector of row weights |
| X.cw | numeric vector of column weights for the explanalatory dataset |
| blo | vector of the numbers of variables in each explanatory dataset |
| rank | rank of the analysis |
| eig | numeric vector containing the eigenvalues |
| TL | dataframe useful to manage graphical outputs |
| TC | dataframe useful to manage graphical outputs |
| faX | matrix containing the global variable loadings associated with the global explanatory dataset |
| Tc1 | matrix containing the partial variable loadings associated with each explanatory dataset(unit norm) |
| Yc1 | matrix of the variable loadings associated with the dependent dataset |
| lX | matrix of the global components associated with the whole explanatory dataset(scores of the individuals) |
| TlX | matrix containing the partial components associated with each explanatory dataset |
| lY | matrix of the components associated with the dependent dataset |
| cov2 | squared covariance between lY and TlX |
| XYcoef | list of matrices of the regression coefficients of the whole explanatory dataset onto the dependent dataset |

| | |
|---|---|
| intercept | intercept of the regression of the whole explanatory dataset onto the dependent dataset |
| XYcoef.raw | list of matrices of the regression coefficients of the whole raw explanatory dataset onto the raw dependent dataset |
| intercept.raw | intercept of the regression of the whole raw explanatory dataset onto the raw dependent dataset |
| bip | block importances for a given dimension |
| bipc | cumulated block importances for a given number of dimensions |
| vip | variable importances for a given dimension |
| vipc | cumulated variable importances for a given number of dimensions |

## Note

This function is coming from the mbpls function of the R package ade4 (application in order to explain a disjunctive table, limitation of the number of calculated components)

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Bougeard, S. and Dray, S. (2018) Supervised Multiblock Analysis in R with the ade4 Package.Journal of Statistical Software,86(1), 1-17.

## See Also

packMBPLSDA-package

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical, nutrition = nutrition, omics = omics))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 2)
```

`medical` *medical dataset*

## Description

extract of modified medical data obtained from physical examination and questionnaires in a human cohort study

## Usage

```
data("medical")
```

## Format

A data frame with 40 observations on the following 18 variables.

`medic1` a numeric vector
`medic2` a numeric vector
`medic3` a numeric vector
`medic4` a numeric vector
`medic5` a numeric vector
`medic6` a numeric vector
`medic7` a numeric vector
`medic8` a numeric vector
`medic9` a numeric vector
`medic10` a numeric vector
`medic11` a numeric vector
`medic12` a numeric vector
`medic13` a numeric vector
`medic14` a numeric vector
`medic15` a numeric vector
`medic16` a numeric vector
`medic17` a numeric vector
`medic18` a numeric vector

## Details

no details are needed

## Source

non-real data

## Examples

```
data(medical)
```

nutrition *nutritional dataset*

## Description

extract of modified nutritional data obtained by analysis of food questionnaires in a human cohort study

## Usage

```
data("nutrition")
```

## Format

A data frame with 40 observations on the following 33 variables.

`nutri1` a numeric vector

`nutri2` a numeric vector

`nutri3` a numeric vector

`nutri4` a numeric vector

`nutri5` a numeric vector

`nutri6` a numeric vector

`nutri7` a numeric vector

`nutri8` a numeric vector

`nutri9` a numeric vector

`nutri10` a numeric vector

`nutri11` a numeric vector

`nutri12` a numeric vector

`nutri13` a numeric vector

`nutri14` a numeric vector

`nutri15` a numeric vector

`nutri16` a numeric vector

`nutri17` a numeric vector

`nutri18` a numeric vector

`nutri19` a numeric vector

`nutri20` a numeric vector

`nutri21` a numeric vector

`nutri22` a numeric vector

`nutri23` a numeric vector

`nutri24` a numeric vector

`nutri25` a numeric vector

`nutri26` a numeric vector

`nutri27` a numeric vector

`nutri28` a numeric vector

`nutri29` a numeric vector

`nutri30` a numeric vector

`nutri31` a numeric vector

`nutri32` a numeric vector

`nutri33` a numeric vector

## Details

no details are needed

## Source

non-real data

## Examples

```
data(nutrition)
```

---

# omics *metabolomic dataset*

---

## Description

extract of modified metabolomic data obtained by LC-MS analysis of human plasma samples in a cohort study

## Usage

```
data("omics")
```

## Format

A data frame with 40 observations on the following 46 variables.

`omic1` a numeric vector of relative intensities

`omic2` a numeric vector of relative intensities

`omic3` a numeric vector of relative intensities

`omic4` a numeric vector of relative intensities

`omic5` a numeric vector of relative intensities

`omic6` a numeric vector of relative intensities

`omic7` a numeric vector of relative intensities
`omic8` a numeric vector of relative intensities
`omic9` a numeric vector of relative intensities
`omic10` a numeric vector of relative intensities
`omic11` a numeric vector of relative intensities
`omic12` a numeric vector of relative intensities
`omic13` a numeric vector of relative intensities
`omic14` a numeric vector of relative intensities
`omic15` a numeric vector of relative intensities
`omic16` a numeric vector of relative intensities
`omic17` a numeric vector of relative intensities
`omic18` a numeric vector of relative intensities
`omic19` a numeric vector of relative intensities
`omic20` a numeric vector of relative intensities
`omic21` a numeric vector of relative intensities
`omic22` a numeric vector of relative intensities
`omic23` a numeric vector of relative intensities
`omic24` a numeric vector of relative intensities
`omic25` a numeric vector of relative intensities
`omic26` a numeric vector of relative intensities
`omic27` a numeric vector of relative intensities
`omic28` a numeric vector of relative intensities
`omic29` a numeric vector of relative intensities
`omic30` a numeric vector of relative intensities
`omic31` a numeric vector of relative intensities
`omic32` a numeric vector of relative intensities
`omic33` a numeric vector of relative intensities
`omic34` a numeric vector of relative intensities
`omic35` a numeric vector of relative intensities
`omic36` a numeric vector of relative intensities
`omic37` a numeric vector of relative intensities
`omic38` a numeric vector of relative intensities
`omic39` a numeric vector of relative intensities
`omic40` a numeric vector of relative intensities
`omic41` a numeric vector of relative intensities
`omic42` a numeric vector of relative intensities
`omic43` a numeric vector of relative intensities
`omic44` a numeric vector of relative intensities
`omic45` a numeric vector of relative intensities
`omic46` a numeric vector of relative intensities

## Details

no details are needed

## Source

non-real data

## Examples

```
data(omics)
```

---

`permut_mbplsda` *Permutation testing of a multi-block partial least squares discriminant model*

---

## Description

Function to perform permutation testing with 2-fold cross-validation for multi-block partial least squares discriminant analysis, in order to evaluate model validity and predictivity

## Usage

```
permut_mbplsda(object, optdim, bloY, algo = c("max", "gravity", "threshold"),
threshold = 0.5, nrepet = 100, npermut = 100, nbObsPermut = NULL,
outputs = c("ER", "ConfMat", "AUC"), cpus = 1)
```

## Arguments

| | |
|---|---|
| `object` | an object created by mbplsda_nfX |
| `optdim` | integer indicating the (optimal) number of components of the multi-block partial least squares discriminant model |
| `bloY` | integer vector indicating the number of categories per variable of the Y-block. |
| `algo` | character vector indicating the method(s) of prediction to use (see details) |
| `threshold` | numeric indicating the threshold, between 0 and 1, to consider the categories are predicted with the threshold prediction method. |
| `nrepet` | integer indicating the number of repetitions |
| `npermut` | integer indicating the number of Y-block with switching observations |
| `nbObsPermut` | integer indicating the number of switching observations in all the modified Y-blocks |
| `outputs` | character vector indicating the wanted outputs (see details) |
| `cpus` | integer indicating the number of cpus to use when running the code in parallel |

## Details

Three different algorithms are available to predict the categories of observations. In the max, and respectively the threshold algorithms, numeric values are calculated from the matrix of explanatory variables and the regression coefficients. Then, the predicted categorie for each variable of the Y-block is the one which corresponds to the higher predicted value, respectively to the values higher than the indicated threshold. In the gravity algorithm, predicted scores of the observations on the components are calculated. Then, each observation is assigned to the observed category of which it is closest to the barycentre in the component space.

If nbObsPermut is not NULL, t-test are performed to compare mean cross-validated overall prediction error rates (or aera under ROC curve) evaluated on permuted Y-blocks, with the cross-validated overall prediction error rate (or aera under ROC curve) evaluated on the original Y-block.

Available outputs are Error Rates (ER), Confusion Matrix (ConfMat), Aera Under Curve (AUC).

## Value

`RV.YYpermut.values`
: RV coefficient between Y-block and each Y-block with permuted values

`cor.YYpermut.values`
: correlation coefficient between categories in the Y-block and each Y-block with permuted values

`prctGlob.Ychange.values`
: overall percentage of modified values in each Y-block with permuted values

`prct.Ychange.values`
: percentage per category of modified values in each Y-block with permuted values

`descrYperm`
: statistical description of RV.YYpermut, cor.YYpermut, prctGlob.Ychange, prct.Ychange

`TruePosC.max`, `TruePosC.gravity`, `TruePosC.threshold`
: statistical description of cross-validated percentages of true positive observations per category, evaluated on calibration datasets, with the different algorithms (TruePosC.max for "max", TruePosC.gravity for "gravity", TruePosC.threshold for "threshold"), for each Y-block with permuted values

`TruePosV.max`, `TruePosV.gravity`, `TruePosV.threshold`
: statistical description of cross-validated percentages of true positive observations per category, evaluated on validation datasets, with the different algorithms (TruePosV.max for "max", TruePosV.gravity for "gravity", TruePosV.threshold for "threshold"), for each Y-block with permuted values

`TrueNegC.max`, `TrueNegC.gravity`, `TrueNegC.threshold`
: statistical description of cross-validated percentages of true negative observations per category, evaluated on calibration datasets, with the different algorithms (TrueNegC.max for "max", TrueNegC.gravity for "gravity", TrueNegC.threshold for "threshold"), for each Y-block with permuted values

`TrueNegV.max`, `TrueNegV.gravity`, `TrueNegV.threshold`
: statistical description of cross-validated percentages of true negative observations per category, evaluated on validation datasets, with the different algorithms (TrueNegV.max for "max", TrueNegV.gravity for "gravity", TrueNegV.threshold for "threshold"), for each Y-block with permuted values

`FalsePosC.max`, `FalsePosC.gravity`, `FalsePosC.threshold`
: statistical description of cross-validated percentages of false positive observations per category, evaluated on calibration datasets, with the different algorithms (FalsePosC.max for "max", FalsePosC.gravity for "gravity", FalsePosC.threshold for "threshold"), for each Y-block with permuted values

`FalsePosV.max`, `FalsePosV.gravity`, `FalsePosV.threshold`
: statistical description of cross-validated percentages of false positive observations per category, evaluated on validation datasets, with the different algorithms (FalsePosV.max for "max", FalsePosV.gravity for "gravity", FalsePosV.threshold for "threshold"), for each Y-block with permuted values

`FalseNegC.max`, `FalseNegC.gravity`, `FalseNegC.threshold`
: statistical description of cross-validated percentages of false negative observations per category, evaluated on calibration datasets, with the different algorithms (FalseNegC.max for "max", FalseNegC.gravity for "gravity", FalseNegC.threshold for "threshold"), for each Y-block with permuted values

`FalseNegV.max`, `FalseNegV.gravity`, `FalseNegV.threshold`
: statistical description of cross-validated percentages of false negative observations per category, evaluated on validation datasets, with the different algorithms (FalseNegV.max for "max", FalseNegV.gravity for "gravity", FalseNegV.threshold for "threshold"), for each Y-block with permuted values

`ErrorRateC.max`, `ErrorRateC.gravity`, `ErrorRateC.threshold`
: statistical description of cross-validated prediction error rates per category, evaluated on calibration datasets, with the different algorithms (ErrorRateC.max for "max", ErrorRateC.gravity for "gravity", ErrorRateC.threshold for "threshold"), for each Y-block with permuted values

`ErrorRateV.max`, `ErrorRateV.gravity`, `ErrorRateV.threshold`
: statistical description of cross-validated prediction error rates per category, evaluated on validation datasets, with the different algorithms (ErrorRateV.max for "max", ErrorRateV.gravity for "gravity", ErrorRateV.threshold for "threshold"), for each Y-block with permuted values

`ErrorRateCglobal.max`, `ErrorRateCglobal.gravity`, `ErrorRateCglobal.threshold`
: statistical description of cross-validated overall prediction error rates, evaluated on calibration datasets, with the different algorithms (ErrorRateCglobal.max for "max", ErrorRateCglobal.gravity for "gravity", ErrorRateCglobal.threshold for "threshold"), for each Y-block with permuted values

`ErrorRateVglobal.max`, `ErrorRateVglobal.gravity`, `ErrorRateVglobal.threshold`
: statistical description of cross-validated overall prediction error rates, evaluated on validation datasets, with the different algorithms (ErrorRateVglobal.max for "max", ErrorRateVglobal.gravity for "gravity", ErrorRateVglobal.threshold for "threshold"), for each Y-block with permuted values

`AUCc`
: if all Y-block variables are binary, statistical description of cross-validated aera under ROC curve values per category, evaluated on the validation datasets, for each Y-block with permuted values

`AUCv`
: if all Y-block variables are binary, statistical description of cross-validated aera under ROC curve values per category, evaluated on the validation datasets, for each Y-block with permuted values

`AUCc.global` if all Y-block variables are binary, statistical description of cross-validated overall aera under ROC curve values, evaluated on the validation datasets, for each Y-block with permuted values

`AUCv.global` if all Y-block variables are binary, statistical description of cross-validated overall aera under ROC curve values, evaluated on the validation datasets, for each Y-block with permuted values

`reg.GlobalRes_prctYchange` results of linear regression of overall prediction error rates, and overall aera under ROC curve, onto percentages of modified values in Y-block

`ttestMeanERv` if nbObsPermut is not NULL, results of the t-test comparing mean cross-validated overall prediction error rates (and eventually aera under ROC curve) evaluated on permuted Y-blocks, with the cross-validated overall prediction error rate (and eventually aera under ROC curve) evaluated on the original Y-block

## Note

at least 30 cross-validation repetitions and 100 Y-block with switching observations may be recommended

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Westerhuis, J.A., Hoefsloot, H.C.J., Smit, S., Vis, D.J., Smilde, A.K., van Velzen, E.J.J., van Duijnhoven, J.P.M., van Dorsten, F.A. (2008). Assessment of PLSDA cross validation. Metabolomics, 4, 81-89.

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

`mbplsda` `plot_permut_mbplsda` `packMBPLSDA-package`

## Examples

```
data(status)
data(medical)
data(omics)
```

```
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical[1:20,], omics = omics[1:20,]))
disjonctif <- (disjunctive(data.frame(status=status[1:20,],
row.names = rownames(status)[1:20])))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
ncpopt <- 1
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform",
scannf = FALSE, nf = 1)
rtsPermut <- permut_mbplsda(modelembplsQ, nrepet = 30, npermut = 100, optdim = ncpopt,
outputs = c("ER"), bloY = bloYobs, nbObsPermut = 10, cpus=1, algo = c("max"))
```

---

`plot_boot_mbplsda` *Plot the results of the fonction boot_mbplsda in a pdf file*

---

## Description

Fonction to draw the results of the fonction boot_mbplsda (2-fold cross-validated parameter values) in a pdf file

## Usage

```
plot_boot_mbplsda(obj, filename = "PlotBootstrapMbplsda", propbestvar = 0.5)
```

## Arguments

| | |
|---|---|
| `obj` | object type list containing the results of the fonction boot_mbplsda |
| `filename` | a string of characters indicating the given pdf filename |
| `propbestvar` | numeric value between 0 and 1, indicating the pourcentage of variables with the best VIPc values to plot |

## Details

no details are needed

## Value

no numeric result

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Efron, B., Tibshirani, R.J. (1994). An Introduction to the Bootstrap. Chapman and Hall-CRC Monographs on Statistics and Applied Probability, Norwell, Massachusetts, United States.

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

mbplsda boot_mbplsda packMBPLSDA-package

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical, nutrition = nutrition, omics = omics))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
ncpopt <- 1
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 2)
resboot <- boot_mbplsda(modelembplsQ, optdim = ncpopt, nrepet = 30, cpus=1)
plot_boot_mbplsda(resboot,"plotBoot_nf1_30rep", propbestvar=0.20)
```

---

# plot_cvpred_mbplsda *Plot the results of the fonction cvpred_mbplsda in a pdf file*

---

## Description

Fonction to draw the results of the fonction cvpred_mbplsda (2-fold cross-validated predictions) in a pdf file

## Usage

```
plot_cvpred_mbplsda(obj, filename = "PlotCVpredMbplsda")
```

## Arguments

| | |
|---|---|
| obj | object type list containing the results of the fonction cvpred_mbplsda |
| filename | a string of characters indicating the given pdf filename |

## Details

no details are needed

## Value

no numeric result

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society B, 36(2), 111-147.

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

mbplsda cvpred_mbplsda packMBPLSDA-package

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical[,1:10],
nutrition = nutrition[,1:10], omics = omics[,1:20]))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
ncpopt <- 1
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform",
scannf = FALSE, nf = 2)
```

```
CVpred <- cvpred_mbplsda(modelembplsQ, nrepet = 30, threshold = 0.5, bloY=bloYobs,
optdim=ncpopt, cpus = 1, algo = c("max"))
plot_cvpred_mbplsda(CVpred,"plotCVPred_nf1_30rep")



data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical,
nutrition = nutrition, omics = omics))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
ncpopt <- 1
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform",
scannf = FALSE, nf = 2)
CVpred <- cvpred_mbplsda(modelembplsQ, nrepet = 90, threshold = 0.5, bloY=bloYobs,
optdim=ncpopt, cpus = 1, algo = c("max"))
plot_cvpred_mbplsda(CVpred,"plotCVPred_nf1_90rep")
```

---

`plot_permut_mbplsda` *Plot the results of the fonction permut_mbplsda in a pdf file*

---

## Description

Fonction to draw the results of the fonction permut_mbplsda (plot and regression line of cross validated prediction error rates, evaluated on the validation datasets, in function of the percent of modified Y-block values) in a pdf file

## Usage

```
plot_permut_mbplsda(obj, filename = "PlotPermutationTest",
MainPlot = "Permutation test results \n (subset of validation)")
```

## Arguments

| | |
|---|---|
| `obj` | object type list containing the results of the fonction permut_mbplsda |
| `filename` | a string of characters indicating the given pdf filename |
| `MainPlot` | a string of characters indicating the given main title |

## Details

no details are needed

## Value

no numeric result

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Westerhuis, J.A., Hoefsloot, H.C.J., Smit, S., Vis, D.J., Smilde, A.K., van Velzen, E.J.J., van Duijnhoven, J.P.M., van Dorsten, F.A. (2008). Assessment of PLSDA cross validation. Metabolomics, 4, 81-89.

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

mbplsda permut_mbplsda packMBPLSDA-package

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical[1:20,], omics = omics[1:20,]))
disjonctif <- (disjunctive(data.frame(status=status[1:20,],
row.names = rownames(status)[1:20])))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 1)
ncpopt <- 1
rtsPermut <- permut_mbplsda(modelembplsQ, nrepet = 30, npermut = 100, optdim = ncpopt,
outputs = c("ER"), bloY=bloYobs, nbObsPermut = 10, cpus = 1, algo = c("max"))
plot_permut_mbplsda(rtsPermut,"plotPermut_nf1_30rep_100perm")
```

---

# plot_pred_mbplsda *Plot the results of the fonction pred_mbplsda in a pdf file*

---

## Description

Fonction to draw the results of the fonction pred_mbplsda (observed parameter values and predictions) in a pdf file

## Usage

```
plot_pred_mbplsda(obj, filename = "PlotPredMbplsda", propbestvar = 0.5)
```

## Arguments

| | |
|---|---|
| obj | object type list containing the results of the fonction pred_mbplsda |
| filename | a string of characters indicating the given pdf filename |
| propbestvar | numeric value between 0 and 1, indicating the pourcentage of variables with the best VIPc values to plot |

## Details

no details are needed

## Value

no numeric result

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

mbplsda pred_mbplsda packMBPLSDA-package

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical, nutrition = nutrition, omics = omics))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
```

```
ncpopt <- 1
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 2)
predictions <- pred_mbplsda(modelembplsQ, optdim = ncpopt, threshold = 0.5,
bloY=bloYobs, algo = c("max", "gravity", "threshold"))
plot_pred_mbplsda(predictions,"plotPred_nf1", propbestvar=0.20)
```

---

`plot_testdim_mbplsda` *Plot the results of the fonction testdim_mbplsda in a pdf file*

---

## Description

Fonction to draw the results of the fonction testdim_mbplsda (cross validated prediction error rates, or aera under ROC curve, in function of the number of components in the model) in a pdf file

## Usage

```
plot_testdim_mbplsda(obj, filename = "PlotTestdimMbplsda")
```

## Arguments

| | |
|---|---|
| `obj` | object type list containing the results of the fonction testdim_mbplsda |
| `filename` | a string of characters indicating the given pdf filename |

## Details

no details are needed

## Value

no numeric result

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society B, 36(2), 111-147.

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

mbplsda testdim_mbplsda packMBPLSDA-package

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical[,1:10],
nutrition = nutrition[,1:10], omics = omics[,1:20]))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 3)
resdim <- testdim_mbplsda(object=modelembplsQ, nrepet = 30, threshold = 0.5,
bloY=bloYobs, cpus=1, algo = c("max"), outputs = c("ER"))
plot_testdim_mbplsda(resdim, "plotTDim")
```

---

# pred_mbplsda *Observed parameters and predicted categories from a multi-block partial least squares discriminant model*

---

## Description

Fonction to perform categories predictions from a multi-block partial least squares discriminant model.

## Usage

```
pred_mbplsda(object, optdim , threshold = 0.5, bloY,
algo = c("max", "gravity", "threshold"))
```

## Arguments

| | |
|---|---|
| object | an object created by mbplsda |
| optdim | integer indicating the (optimal) number of components of the multi-block partial least squares discriminant model |
| threshold | numeric indicating the threshold, between 0 and 1, to consider the categories are predicted with the threshold prediction method. |
| bloY | integer vector indicating the number of categories per variable of the Y-block. |
| algo | character vector indicating the method(s) of prediction to use (see details) |

## Details

Three different algorithms are available to predict the categories of observations. In the max, and respectively the threshold algorithms, numeric values are calculated from the matrix of explanatory variables and the regression coefficients. Then, the predicted categorie for each variable of the Y-block is the one which corresponds to the higher predicted value, respectively to the values higher than the indicated threshold. In the gravity algorithm, predicted scores of the observations on the components are calculated. Then, each observation is assigned to the observed category of which it is closest to the barycentre in the component space.

## Value

| | |
|---|---|
| `XYcoef` | list of matrices of the regression coefficients of the whole explanatory dataset onto the dependent dataset |
| `VIPc` | cumulated variable importances for a given number of dimensions |
| `BIPc` | cumulated block importances for a given number of dimensions |
| `faX` | matrix containing the global variable loadings associated with the global explanatory dataset |
| `lX` | matrix of the global components associated with the whole explanatory dataset(scores of the individuals) |
| `ConfMat.ErrorRate` | confidence matrix and prediction error rate per category |
| `ErrorRate.global` | confidence matrix and prediction error rate, per Y-block variable and overall |
| `PredY.max` | predictions and accuracy of predictions with the "max" algorithm |
| `PredY.gravity` | predictions and accuracy of predictions with the "gravity" algorithm |
| `PredY.threshold` | predictions and accuracy of predictions with the "threshold" algorithm |
| `AUC` | aera under ROC cuve value and 95% confidence interval, per category, per Y-block variable and overall |

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

mbplsda plot_pred_mbplsda packMBPLSDA-package

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical, nutrition = nutrition, omics = omics))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
ncpopt <- 1
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 2)
predictions <- pred_mbplsda(modelembplsQ, optdim = ncpopt, threshold = 0.5, bloY=bloYobs,
algo = c("max", "gravity", "threshold"))
```

---

# status *physiopathological status data*

---

## Description

physiopathological status of men in a human cohort study

## Usage

```
data("status")
```

## Format

A data frame with 40 observations on the following variable.

`status` a factor with levels `cas` `temoin`

## Details

no details are needed

## Source

extract of data not yet published

## Examples

```
data(status)
```

`testdim_mbplsda` *Test of number of components by two-fold cross-validation for a multi-block partial least squares discriminant model*

## Description

Function to perform a two-fold cross-validation in order to select the optimal number of dimensions of a multi-block partial least squares discriminant model, according to the classification error rate or to the area under ROC curve

## Usage

```
testdim_mbplsda(object, nrepet = 100, algo = c("max", "gravity", "threshold"),
threshold = 0.5, bloY, outputs = c("ER", "ConfMat", "AUC"), cpus = 1)
```

## Arguments

| | |
|---|---|
| `object` | an object created by mbplsda_nfX |
| `nrepet` | integer indicating the number of repetitions |
| `algo` | character vector indicating the method(s) of prediction to use (see details) |
| `threshold` | numeric indicating the threshold, between 0 and 1, to consider the categories are predicted with the threshold prediction method. |
| `bloY` | integer vector indicating the number of categories per variable of the Y-block. |
| `outputs` | character vector indicating the wanted outputs (see details) |
| `cpus` | integer indicating the number of cpus to use when running the code in parallel |

## Details

Three different algorithms are available to predict the categories of observations. In the max, and respectively the threshold algorithms, numeric values are calculated from the matrix of explanatory variables and the regression coefficients. Then, the predicted categorie for each variable of the Y-block is the one which corresponds to the higher predicted value, respectively to the values higher than the indicated threshold. In the gravity algorithm, predicted scores of the observations on the components are calculated. Then, each observation is assigned to the observed category of which it is closest to the barycentre in the component space.

Available outputs are Error Rates (ER), Confusion Matrix (ConfMat), Aera Under Curve (AUC).

## Value

| | |
|---|---|
| `TRUEnrepet` | number of repetitions |
| `TruePosC.max`, `.gravity`, `.threshold` | statistical description of percentages of true positive observations per category, evaluated on the calibration dataset, with the different algorithms (TPcM for "max", TPcG for "gravity", TPcT for "threshold"), for a number of components ranging from 1 to its maximum value |

`TruePosV.max`, `.gravity`, `.threshold`
: statistical description of percentages of true positive observations per category, evaluated on the validation dataset, with the different algorithms (TPvM for "max", TPvG for "gravity", TPvT for "threshold"), for a number of components ranging from 1 to its maximum value

`TrueNegC.max`, `.gravity`, `.threshold`
: statistical description of percentages of true negative observations per category, evaluated on the calibration dataset, with the different algorithms (TNcM for "max", TNcG for "gravity", TNcT for "threshold"), for a number of components ranging from 1 to its maximum value

`TrueNegV.max`, `.gravity`, `.threshold`
: statistical description of percentages of true negative observations per category, evaluated on the validation dataset, with the different algorithms (TNvM for "max", TNvG for "gravity", TNvT for "threshold"), for a number of components ranging from 1 to its maximum value

`FalsePosC.max`, `.gravity`, `.threshold`
: statistical description of percentages of false positive observations per category, evaluated on the calibration dataset, with the different algorithms (FPcM for "max", FPcG for "gravity", FPcT for "threshold"), for a number of components ranging from 1 to its maximum value

`FalsePosV.max`, `.gravity`, `.threshold`
: statistical description of percentages of false positive observations per category, evaluated on the validation dataset, with the different algorithms (FPvM for "max", FPvG for "gravity", FPvT for "threshold"), for a number of components ranging from 1 to its maximum value

`FalseNegC.max`, `.gravity`, `.threshold`
: statistical description of percentages of false negative observations per category, evaluated on the calibration dataset, with the different algorithms (FNcM for "max", FNcG for "gravity", FNcT for "threshold"), for a number of components ranging from 1 to its maximum value

`FalseNegV.max`, `.gravity`, `.threshold`
: statistical description of percentages of false negative observations per category, evaluated on the validation dataset, with the different algorithms (FNvM for "max", FNvG for "gravity", FNvT for "threshold"), for a number of components ranging from 1 to its maximum value

`ErrorRateC.max`, `.gravity`, `.threshold`
: statistical description of prediction error rates per category, evaluated on the calibration dataset, with the different algorithms (ERcM for "max", ERcG for "gravity", ERcT for "threshold"), for a number of components ranging from 1 to its maximum value

`ErrorRateV.max`, `.gravity`, `.threshold`
: statistical description of prediction error rates per category, evaluated on the validation dataset, with the different algorithms (ERvM for "max", ERvG for "gravity", ERvT for "threshold"), for a number of components ranging from 1 to its maximum value

`ErrorRateCglobal.max`, `.gravity`, `.threshold`
: statistical description of global prediction error rates, evaluated on the calibration dataset, with the different algorithms (ERcM.global for "max", ERcG.global

for "gravity", ERcT.global for "threshold"), for a number of components ranging from 1 to its maximum value

`ErrorRateVglobal.max`, `.gravity`, `.threshold`
: statistical description of global prediction error rates, evaluated on the validation dataset, with the different algorithms (ERvM.global for "max", ERvG.global for "gravity", ERvT.global for "threshold"), for a number of components ranging from 1 to its maximum value

`AUCc`
: statistical description of aera under ROC curve values per category, evaluated on the calibration dataset, if all Y-block variables are binary, for a number of components ranging from 1 to its maximum value

`AUCv`
: statistical description of aera under ROC curve values per category, evaluated on the validation dataset, if all Y-block variables are binary, for a number of components ranging from 1 to its maximum value

`AUCc.global`
: statistical description of global aera under ROC curve values, evaluated on the calibration dataset, if all Y-block variables are binary, for a number of components ranging from 1 to its maximum value

`AUCv.global`
: statistical description of global aera under ROC curve values, evaluated on the validation dataset, if all Y-block variables are binary, for a number of components ranging from 1 to its maximum value

## Note

at least 30 cross-validation repetitions may be recommended

## Author(s)

Marion Brandolini-Bunlon (<marion.brandolini-bunlon@inra.fr>) and Stephanie Bougeard (<stephanie.bougeard@anses.fr>

## References

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society B, 36(2), 111-147.

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at 12emes Journees Scientifiques RFMF, Clermont-Ferrand, FRA(05-21-2019 - 05-23-2019).

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2019). Multi-block PLS discriminant analysis for the joint analysis of metabolomic and epidemiological data. Metabolomics, 15(10):134

Brandolini-Bunlon, M., Petera, M., Gaudreau, P., Comte, B., Bougeard, S., Pujos-Guillot, E.(2020). A new tool for multi-block PLS discriminant analysis of metabolomic data: application to systems epidemiology. Presented at Chimiometrie 2020, Liege, BEL(01-27-2020 - 01-29-2020).

## See Also

`mbplsda` `plot_testdim_mbplsda` `packMBPLSDA-package`

## Examples

```
data(status)
data(medical)
data(omics)
data(nutrition)
ktabX <- ktab.list.df(list(medical = medical[,1:10],
nutrition = nutrition[,1:10], omics = omics[,1:20]))
disjonctif <- (disjunctive(status))
dudiY   <- dudi.pca(disjonctif , center = FALSE, scale = FALSE, scannf = FALSE)
bloYobs <- 2
modelembplsQ <- mbplsda(dudiY, ktabX, scale = TRUE, option = "uniform", scannf = FALSE, nf = 3)
resdim <- testdim_mbplsda(object = modelembplsQ, nrepet = 30, threshold = 0.5,
bloY = bloYobs, cpus = 1, algo = c("max"), outputs = c("ER"))
```

# Index