Package ‘MBAnalysis’

July 10, 2025

Title Multiblock Exploratory and Predictive Data Analysis

Version 2.1.0

Description Exploratory and predictive methods for the analysis of several blocks of variables measured on the same individuals.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.0.0)

LazyData false

Imports ggplot2, ggrepel, grDevices, stats, utils

NeedsCompilation no

Author Benjamin Mahieu [aut, cre],
Essomanda Tchandao Mangamana [aut],
Evelyne Vigneau [aut],
Veronique Cariou [aut]

Maintainer Benjamin Mahieu <`benjamin.mahieu@oniris-nantes.fr`>

Repository CRAN

Date/Publication 2025-07-10 10:00:02 UTC

Contents

`coef.MBPLS` *Regression coefficients of MBPLS models*

Description

Computes regression coefficients from `MBPLS`.

Usage

```
## S3 method for class 'MBPLS'
coef(object, ncomp = object$call$ncomp, ...)
```

Arguments

`object`	An object resulting from `MBPLS`.
`ncomp`	The number of components to be considered in the model. By default, all components computed in `MBPLS` or `MBWCov` are considered.
`...`	further arguments passed to or from other methods.

Value

A matrix of regression coefficients where each row corresponds to a variable in X and each column corresponds to a variable in Y.

See Also

`predict.MBPLS`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block = names(block))
coef(res.mbpls)
```

`coef.MBWCov` *Regression coefficients of MBWCov models*

Description

Computes regression coefficients from `MBWCov`.

Usage

```
## S3 method for class 'MBWCov'
coef(object, ncomp = object$call$ncomp, ...)
```

Arguments

`object`	An object resulting from `MBWCov`.
`ncomp`	The number of components to be considered in the model. By default, all components computed in `MBPLS` or `MBWCov` are considered.
`...`	further arguments passed to or from other methods.

Value

A matrix of regression coefficients where each row corresponds to a variable in X and each column corresponds to a variable in Y.

See Also

`predict.MBWCov`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbwcov <- MBWCov(X, Y, block, name.block = names(block))
coef(res.mbwcov)
```

`ComDim` *Common Dimensions analysis (ComDim)*

Description

Performs ComDim analysis on a set of quantitative blocks of variables. ComDim can be viewed as a Multiblock Weighted Principal Components Analysis (MBWPCA)

Usage

```
ComDim(
  X,
  block,
  name.block = NULL,
  ncomp = NULL,
  scale = TRUE,
  scale.block = TRUE,
  threshold = 1e-08
)
```

Arguments

`X`	Dataset obtained by horizontally merging all the blocks of variables.
`block`	Vector indicating the number of variables in each block.
`name.block`	names of the blocks of variables (NULL by default).
`ncomp`	Number of dimensions to compute. By default (NULL), all the global components are extracted.
`scale`	Logical, if TRUE (by default) then variables are scaled to unit variance (all variables are centered anyway).
`scale.block`	Logical, if TRUE (by default) each block of variables is divided by the square root of its inertia (Frobenius norm).
`threshold`	Convergence threshold

Value

Returns a list of the following elements:

`optimalcrit`	Numeric vector of the optimal value of the criterion (sum of squared saliences) obtained for each dimension.
`saliences`	Matrix of the specific weights of each block of variables on the global components, for each dimension.
`T.g`	Matrix of normed global components.
`Scor.g`	Matrix of global components (scores of individuals).

`W.g`	Matrix of global weights (normed) associated with deflated X.
`Load.g`	Matrix of global loadings (normed).
`Proj.g`	Matrix of global projection (to compute scores from pretreated X).
`explained.X`	Matrix of percentages of inertia explained in each block of variables.
`cumexplained`	Matrix giving the percentages, and cumulative percentages, of total inertia of X blocks explained by the global components.
`Block`	A list containing block components (T.b) and block weights (W.b)

References

E.M. Qannari, I. Wakeling, P. Courcoux, J.M. MacFie (2000). Defining the underlying sensory dimensions, Food Quality and Preference, 11: 151-154.

E. Tchandao Mangamana, V. Cariou, E. Vigneau, R. Glèlè Kakaï, E.M. Qannari (2019). Unsupervised multiblock data analysis: A unified approach and extensions, Chemometrics and Intelligent Laboratory Systems, 194, 103856.

See Also

`summary.ComDim` `plot.ComDim`

Examples

```
data(ham)
X=ham$X
block=ham$block
res.comdim <- ComDim(X,block,name.block=names(block))
summary(res.comdim)
plot(res.comdim)
```

`ham` *Ham data*

Description

Case study pertaining to the sensory evaluation of eight American dry-cured ham products, performed by a panel of trained assessors.

Usage

```
data(ham)
```

Format

An object of class `"list"` with 8 products, 3 blocks of X variables (Flavor, Aroma, Texture) and 1 block of Y variables corresponding to hedonic measures:

X dataframe of 8 products and 25 variables structured into 3 blocks: Flavor (11 variables), Aroma (8 variables) and Texture (6 variables)

Y dataframe of 8 products and 6 vectors of hedonic values corresponding to consumers' segmentation

block vector indicating the number of variables per block

References

M.D. Guardia, A.P. Aguiar, A. Claret, J. Arnau & L. Guerrero (2010). Sensory characterization of dry-cured ham using free-choice profiling. Food Quality and Preference, 21(1), 148-155. doi:10.1016/j.foodqual.2009.08.014

Examples

```
data(ham)
ham$X
ham$Y
ham$block
```

`MBPCA` *Multiblock Principal Components Analysis (MB-PCA)*

Description

Performs MB-PCA on a set of quantitative blocks of variables.

Usage

```
MBPCA(
  X,
  block,
  name.block = NULL,
  ncomp = NULL,
  scale = TRUE,
  scale.block = TRUE
)
```

Arguments

`X`	Dataset obtained by horizontally merging all the blocks of variables.
`block`	Vector indicating the number of variables in each block.
`name.block`	names of the blocks of variables (NULL by default).

`ncomp`	Number of dimensions to compute. By default (NULL), all the global components are extracted.
`scale`	Logical, if TRUE (by default) then variables are scaled to unit variance (all variables are centered anyway).
`scale.block`	Logical, if TRUE (by default) each block of variables is divided by the square root of its inertia (Frobenius norm).

Value

Returns a list of the following elements:

`optimalcrit`	Numeric vector of the optimal value of the criterion (sum of saliences) obtained for each dimension.
`saliences`	Matrix of the specific weights of each block of variables on the global components, for each dimension.
`T.g`	Matrix of normed global components.
`Scor.g`	Matrix of global components (scores of individuals).
`W.g`	Matrix of global weights (normed) associated with deflated X.
`Load.g`	Matrix of global loadings (normed) = W.g in the specific context of MB-PCA.
`Proj.g`	Matrix of global projection (to compute scores from pretreated X) = W.g in the specific context of MB-PCA.
`explained.X`	Matrix of percentages of inertia explained in each block of variables.
`cumexplained`	Matrix giving the percentages, and cumulative percentages, of total inertia of X blocks explained by the global components.
`Block`	A list containing block components (T.b) and block weights (W.b)

References

S. Wold, S. Hellberg, T. Lundstedt, M. Sjostrom, H. Wold (1987). Hierarchical multiblock PLS and PC models for easier model interpretation and as an alternative to variable selection, in: Proc. Symp. On PLS Model Building: Theory and Application, Frankfurt am Main.

E. Tchandao Mangamana, V. Cariou, E. Vigneau, R. Glèlè Kakaï, E.M. Qannari (2019). Unsupervised multiblock data analysis: A unified approach and extensions, Chemometrics and Intelligent Laboratory Systems, 194, 103856.

See Also

`summary.MBPCA` `plot.MBPCA`

Examples

```
data(ham)
X=ham$X
block=ham$block
res.mbpca <- MBPCA(X,block, name.block=names(block))
summary(res.mbpca)
plot(res.mbpca)
```

`MBplotBlocks` *Customizable plots of blocks related information*

Description

Plots blocks related information of `MBPCA`, `ComDim`, `MBPLS` or `MBWCov` with several options of customization.

Usage

```
MBplotBlocks(
  res,
  which = "explained.blocks&Y",
  axes = c(1, 2),
  blocks.axes = 1:max(axes),
  title = NULL,
  size = 2.25
)
```

Arguments

`res`	An object resulting from `MBPCA`, `ComDim`, `MBPLS` or `MBWCov`.
`which`	Either "explained.blocks&Y", "scree", "structure" or "blocks.axes". See details.
`axes`	Which global dimensions should be plotted? Only useful if *which=structure* or *which=blocks.axes*
`blocks.axes`	Which individual blocks dimensions should be correlated with global ones? Only useful if *which=blocks.axes*
`title`	An optional title to be added to the plot.
`size`	The overall size of labels, points, etc.

Details

- **explained.blocks&Y**: Barplot of the percentages of inertia explained in each block of variables (and Y for `MBPLS` or `MBWCov`) by each global components.
- **scree**: Barplot of the saliences of each block of variables on each global components.
- **structure**: Blocks coordinates (saliences) on the global selected *axes*
- **blocks.axes**: Correlations of the selected individual *blocks.axes* with the global selected *axes*.

Value

The required plot.

See Also

`plot.MBPCA` `plot.ComDim` `plot.MBPLS` `plot.MBWCov`

Examples

```
# Unsupervised example

data(ham)
X=ham$X
block=ham$block
res.mbpca <- MBPCA(X,block, name.block=names(block))
MBplotBlocks(res.mbpca,which="explained.blocks&Y")
MBplotBlocks(res.mbpca,which="scree")
MBplotBlocks(res.mbpca,which="structure")
MBplotBlocks(res.mbpca,which="blocks.axes")

# Supervised example

data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block=names(block))
MBplotBlocks(res.mbpls,which="explained.blocks&Y")
MBplotBlocks(res.mbpls,which="scree")
MBplotBlocks(res.mbpls,which="structure")
MBplotBlocks(res.mbpls,which="blocks.axes")
```

`MBplotScores` *Customizable plots of scores related information*

Description

Plots scores related information of `MBPCA`, `ComDim`, `MBPLS` or `MBWCov` with several options of customization.

Usage

```
MBplotScores(
  res,
  axes = c(1, 2),
  block = 0,
  color = NULL,
  select = 1:nrow(res$Scor.g),
```

```
  title = NULL,
  size = 2.25
)
```

Arguments

`res`	An object resulting from `MBPCA`, `ComDim`, `MBPLS` or `MBWCov`.
`axes`	Which dimensions should be plotted?
`block`	Of which block? Block 0 corresponds to global components.
`color`	Either NULL (default) or a character vector of length *select*. Controls the color of each individual plotted. Useful if individuals pertain to different a priori known groups. By default individuals are colored in black for global components and in the block color (the same as in `MBplotVars`) for block components.
`select`	A numeric or integer vector to select which individuals should be plotted. By default, all individuals are plotted.
`title`	An optional title to be added to the plot.
`size`	The overall size of labels, points, etc.

Value

The required plot.

See Also

`plot.MBPCA` `plot.ComDim` `plot.MBPLS` `plot.MBWCov`

Examples

```
# Unsupervised example

data(ham)
X=ham$X
block=ham$block
res.mbpca <- MBPCA(X,block, name.block=names(block))
MBplotScores(res.mbpca)

# Supervised example

data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block=names(block))
MBplotScores(res.mbpls)
```

`MBplotVars` *Customizable plots of variables related information*

Description

Plots variables related information of `MBPCA`, `ComDim`, `MBPLS` or `MBWCov` with several options of customization.

Usage

```
MBplotVars(
  res,
  axes = c(1, 2),
  which = ifelse(res$call$scale, "correlation", "loading"),
  block = 0,
  select = 0,
  title = NULL,
  size = 2.25
)
```

Arguments

`res`	An object resulting from `MBPCA`, `ComDim`, `MBPLS` or `MBWCov`.
`axes`	Which dimensions should be plotted?
`which`	Either "correlation" or "loading".
`block`	Selection of variables by blocks. A number or integer, possibly a vector, corresponding to the index of the blocks from which the variables should be plotted. For `MBPLS` and `MBWCov` the Y response block corresponds to the index *length(res$call$size.block)+1*. By default, all variables of all blocks are plotted. Only one of *block* and *select* can differ from 0 (the default) both at the same time.
`select`	Selection of variables by index. A number or integer, possibly a vector, corresponding to the index of the variables that should be plotted. For `MBPLS` and `MBWCov` the Y variables index start from *ncol(res$call$X)+1*. By default, all variables of all blocks are plotted. Only one of *block* and *select* can differ from 0 (the default) both at the same time.
`title`	An optional title to be added to the plot.
`size`	The overall size of labels, points, etc.

Value

The required plot.

See Also

`plot.MBPCA` `plot.ComDim` `plot.MBPLS` `plot.MBWCov`

Examples

```
# Unsupervised example

data(ham)
X=ham$X
block=ham$block
res.mbpca <- MBPCA(X,block, name.block=names(block))
MBplotVars(res.mbpca)

# Supervised example

data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block=names(block))
MBplotVars(res.mbpls)
```

`MBPLS` *Multiblock Partial Least Squares (MB-PLS) regression*

Description

MB-PLS regression applied to a set of quantitative blocks of variables.

Usage

```
MBPLS(
  X,
  Y,
  block,
  name.block = NULL,
  ncomp = NULL,
  scale = TRUE,
  scale.block = TRUE,
  scale.Y = TRUE
)
```

Arguments

`X`	Dataset obtained by horizontally merging all the predictor blocks of variables.
`Y`	Response block of variables.
`block`	Vector indicating the number of variables in each predictor block.
`name.block`	Names of the predictor blocks of variables (NULL by default).
`ncomp`	Number of dimensions to compute. By default (NULL), all the global components are extracted.

`scale`	Logical, if TRUE (by default) the variables in X are scaled to unit variance (all variables in X are centered anyway).
`scale.block`	Logical, if TRUE (by default) each predictor block of variables is divided by the square root of its inertia (Frobenius norm).
`scale.Y`	Logical, if TRUE (by default) then variables in Y are scaled to unit variance (all variables in Y are centered anyway).

Value

Returns a list of the following elements:

`optimalcrit`	Numeric vector of the optimal value of the criterion (sum of saliences) obtained for each dimension.
`saliences`	Matrix of the specific weights of each predictor block on the global components, for each dimension.
`T.g`	Matrix of normed global components.
`Scor.g`	Matrix of global components (scores of individuals).
`W.g`	Matrix of global weights (normed) associated with deflated X.
`Load.g`	Matrix of global loadings.
`Proj.g`	Matrix of global projection (to compute scores from pretreated X).
`explained.X`	Matrix of percentages of inertia explained in each predictor block.
`cumexplained`	Matrix giving the percentages, and cumulative percentages, of total inertia of X and Y blocks explained by the global components.
`Y`	A list containing un-normed Y components (U), normed Y weights (W.Y) and Y loadings (Load.Y)
`Block`	A list containing block components (T.b) and block weights (W.b)

References

S. Wold (1984). Three PLS algorithms according to SW. In: Symposium MULDAST (Multivariate Analysis in Science and Technology), Umea University, Sweden. pp. 26–30.

E. Tchandao Mangamana, R. Glèlè Kakaï, E.M. Qannari (2021). A general strategy for setting up supervised methods of multiblock data analysis. Chemometrics and Intelligent Laboratory Systems, 217, 104388.

See Also

`summary.MBPLS` `plot.MBPLS`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block = names(block))
summary(res.mbpls)
plot(res.mbpls)
```

`MBValidation` *Cross-Validation of MBPLS or MBWCov models*

Description

Computes MSEP and corresponding standard error based on Leave One Out (LOO) or Out Of Bag (OOB) Cross-Validation (CV) by number of components of a MBPLS or MBWCov model from `MBPLS` or `MBWCov`.

Usage

```
MBValidation(
  res,
  ncomp.max = min(res$call$ncomp, nrow(res$call$X) - 2, ncol(X)),
  method = "LOO",
  nboot = 1000,
  graph = TRUE,
  size.graph = 2.25
)
```

Arguments

`res`	An object resulting from `MBPLS` or `MBWCov`.
`ncomp.max`	The maximum number of components to be investigated in the CV procedure.
`method`	Either "LOO" or "OOB". Default is LOO.
`nboot`	Number of bootstrap samples to be generated in case of OOB CV.
`graph`	Logical. Should the results be plotted? Default is TRUE.
`size.graph`	If *graph=TRUE*, the overall size of labels, points, etc.

Value

A matrix with two rows (MSEP and std.error) and *ncomp.max+1* columns. The +1 column corresponds to the null model (Dim.0) where Y is predicted by its empirical average on the training sample.

See Also

predict.MBPLS predict.MBWCov

Examples

```
# With MBPLS

data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block = names(block))
MBValidation(res.mbpls)

# With MBWCov

data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbwcov <- MBWCov(X, Y, block, name.block = names(block))
MBValidation(res.mbwcov)
```

`MBWCov` *Multiblock Weighted Covariate analysis (MB-WCov)*

Description

MB-WCov analysis applied to a set of quantitative blocks of variables.

Usage

```
MBWCov(
  X,
  Y,
  block,
  name.block = NULL,
  ncomp = NULL,
  scale = TRUE,
  scale.block = TRUE,
  scale.Y = TRUE,
  threshold = 1e-08
)
```

Arguments

`X`	Dataset obtained by horizontally merging all the predictor blocks of variables.
`Y`	Response block of variables.
`block`	Vector indicating the number of variables in each predictor block.
`name.block`	Names of the predictor blocks of variables (NULL by default).
`ncomp`	Number of dimensions to compute. By default (NULL), all the global components are extracted.
`scale`	Logical, if TRUE (by default) the variables in X are scaled to unit variance (all variables in X are centered anyway).
`scale.block`	Logical, if TRUE (by default) each predictor block of variables is divided by the square root of its inertia (Frobenius norm).
`scale.Y`	Logical, if TRUE (by default) then variables in Y are scaled to unit variance (all variables in Y are centered anyway).
`threshold`	Convergence threshold

Value

`optimalcrit`	Numeric vector of the optimal value of the criterion (sum of squared saliences) obtained for each dimension.
`saliences`	Matrix of the specific weights of each predictor block on the global components, for each dimension.
`T.g`	Matrix of normed global components.
`Scor.g`	Matrix of global components (scores of individuals).
`W.g`	Matrix of global weights (normed) associated with deflated X.
`Load.g`	Matrix of global loadings.
`Proj.g`	Matrix of global projection (to compute scores from pretreated X).
`explained.X`	Matrix of percentages of inertia explained in each predictor block.
`cumexplained`	Matrix giving the percentages, and cumulative percentages, of total inertia of X and Y blocks explained by the global components.
`Y`	A list containing un-normed Y components (U), normed Y weights (W.Y) and Y loadings (Load.Y)
`Block`	A list containing block components (T.b) and block weights (W.b)

References

E. Tchandao Mangamana, R. Glèlè Kakaï, E.M. Qannari (2021). A general strategy for setting up supervised methods of multiblock data analysis. Chemometrics and Intelligent Laboratory Systems, 217, 104388.

See Also

`summary.MBWCov` `plot.MBWCov`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbwcov <- MBWCov(X, Y, block, name.block = names(block))
summary(res.mbwcov)
plot(res.mbwcov)
```

`plot.ComDim` *Default plots for `ComDim` objects*

Description

Successively performs `MBplotScores`, `MBplotVars` and `MBplotBlocks` with the default values of parameters but axes and size.

Usage

```
## S3 method for class 'ComDim'
plot(x, axes = c(1, 2), size = 2.25, ...)
```

Arguments

`x`	An object resulting from `ComDim`.
`axes`	Which dimensions should be plotted?
`size`	The overall size of labels, points, etc.
`...`	further arguments passed to or from other methods.

Value

The default plots.

See Also

`MBplotScores` `MBplotVars` `MBplotBlocks`

Examples

```
data(ham)
X=ham$X
block=ham$block
res.comdim <- ComDim(X,block,name.block=names(block))
plot(res.comdim)
```

`plot.MBPCA` *Default plots for* `MBPCA` *objects*

Description

Successively performs `MBplotScores`, `MBplotVars` and `MBplotBlocks` with the default values of parameters but axes and size.

Usage

```
## S3 method for class 'MBPCA'
plot(x, axes = c(1, 2), size = 2.25, ...)
```

Arguments

`x`	An object resulting from `MBPCA`.
`axes`	Which dimensions should be plotted?
`size`	The overall size of labels, points, etc.
`...`	further arguments passed to or from other methods.

Value

The default plots.

See Also

`MBplotScores` `MBplotVars` `MBplotBlocks`

Examples

```
data(ham)
X=ham$X
block=ham$block
res.mbpca <- MBPCA(X,block, name.block=names(block))
plot(res.mbpca)
```

`plot.MBPLS` *Default plots for* `MBPLS` *objects*

Description

Successively performs `MBplotScores`, `MBplotVars` and `MBplotBlocks` with the default values of parameters but axes and size.

Usage

```
## S3 method for class 'MBPLS'
plot(x, axes = c(1, 2), size = 2.25, ...)
```

Arguments

`x`	An object resulting from `MBPLS`.
`axes`	Which dimensions should be plotted?
`size`	The overall size of labels, points, etc.
`...`	further arguments passed to or from other methods.

Value

The default plots.

See Also

`MBplotScores` `MBplotVars` `MBplotBlocks`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block = names(block))
plot(res.mbpls)
```

`plot.MBWCov` *Default plots for* `MBWCov` *objects*

Description

Successively performs `MBplotScores`, `MBplotVars` and `MBplotBlocks` with the default values of parameters but axes and size.

Usage

```
## S3 method for class 'MBWCov'
plot(x, axes = c(1, 2), size = 2.25, ...)
```

Arguments

`x`	An object resulting from `MBWCov`.
`axes`	Which dimensions should be plotted?
`size`	The overall size of labels, points, etc.
`...`	further arguments passed to or from other methods.

Value

The default plots.

See Also

`MBplotScores` `MBplotVars` `MBplotBlocks`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbwcov <- MBWCov(X, Y, block, name.block = names(block))
plot(res.mbwcov)
```

`predict.MBPLS` *Prediction from MBPLS models*

Description

Computes predictions of Y from `MBPLS` using calibration X (default) or new X observations.

Usage

```
## S3 method for class 'MBPLS'
predict(object, newdata = object$call$X, ncomp = object$call$ncomp, ...)
```

Arguments

`object`	An object resulting from `MBPLS`.
`newdata`	A matrix or data.frame of (new) observations having the same ncol and same colnames as the X of fitting observations.
`ncomp`	The number of components to be considered in the model to perform the predictions. By default, all components computed in `MBPLS` are considered.
`...`	further arguments passed to or from other methods.

Value

A matrix of predicted Y values where each row corresponds to an observation and each column corresponds to a Y variable.

See Also

`coef.MBPLS` `MBValidation`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block = names(block))
predict(res.mbpls)
```

`predict.MBWCov` *Prediction from MBWCov models*

Description

Computes predictions of Y from `MBWCov` using calibration X (default) or new X observations.

Usage

```
## S3 method for class 'MBWCov'
predict(object, newdata = object$call$X, ncomp = object$call$ncomp, ...)
```

Arguments

`object`	An object resulting from `MBWCov`.
`newdata`	A matrix or data.frame of (new) observations having the same ncol and same colnames as the X of fitting observations.
`ncomp`	The number of components to be considered in the model to perform the predictions. By default, all components computed in `MBWCov` are considered.
`...`	further arguments passed to or from other methods.

Value

A matrix of predicted Y values where each row corresponds to an observation and each column corresponds to a Y variable.

See Also

`coef.MBWCov` `MBValidation`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbwcov <- MBWCov(X, Y, block, name.block = names(block))
predict(res.mbwcov)
```

`print.ComDim` *Print of* `ComDim` *objects*

Description

Print of `ComDim` objects

Usage

```
## S3 method for class 'ComDim'
print(x, ...)
```

Arguments

`x`	An object resulting from `ComDim`.
`...`	further arguments passed to or from other methods.

See Also

`summary.ComDim`

Examples

```
data(ham)
X=ham$X
block=ham$block
res.comdim <- ComDim(X,block,name.block=names(block))
print(res.comdim)
```

`print.MBPCA` *Print of* `MBPCA` *objects*

Description

Print of `MBPCA` objects

Usage

```
## S3 method for class 'MBPCA'
print(x, ...)
```

Arguments

`x`	An object resulting from `MBPCA`.
`...`	further arguments passed to or from other methods.

See Also

`summary.MBPCA`

Examples

```
data(ham)
X=ham$X
block=ham$block
res.mbpca <- MBPCA(X,block, name.block=names(block))
print(res.mbpca)
```

`print.MBPLS` *Print of* `MBPLS` *objects*

Description

Print of `MBPLS` objects

Usage

```
## S3 method for class 'MBPLS'
print(x, ...)
```

Arguments

`x`	An object resulting from `MBPLS`.
`...`	further arguments passed to or from other methods.

See Also

`summary.MBPLS`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block = names(block))
print(res.mbpls)
```

`print.MBWCov` *Print of `MBWCov` objects*

Description

Print of `MBWCov` objects

Usage

```
## S3 method for class 'MBWCov'
print(x, ...)
```

Arguments

`x`	An object resulting from `MBWCov`.
`...`	further arguments passed to or from other methods.

See Also

`summary.MBWCov`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbwcov <- MBWCov(X, Y, block, name.block = names(block))
print(res.mbwcov)
```

`summary.ComDim` *Summary of `ComDim` objects*

Description

Edits the Cumulative Explained Variance, Block Explained Variance per Dimension and Block Saliences per Dimension of a `ComDim` object.

Usage

```
## S3 method for class 'ComDim'
summary(object, ...)
```

Arguments

`object`	An object resulting from `ComDim`.
`...`	further arguments passed to or from other methods.

Value

The summary.

See Also

`plot.ComDim`

Examples

```
data(ham)
X=ham$X
block=ham$block
res.comdim <- ComDim(X,block,name.block=names(block))
summary(res.comdim)
```

---

summary.MBPCA *Summary of `MBPCA` objects*

Description

Edits the Cumulative Explained Variance, Block Explained Variance per Dimension and Block Saliences per Dimension of a `MBPCA` object.

Usage

```
## S3 method for class 'MBPCA'
summary(object, ...)
```

Arguments

`object`	An object resulting from `MBPCA`.
`...`	further arguments passed to or from other methods.

Value

The summary.

See Also

`plot.MBPCA`

Examples

```
data(ham)
X=ham$X
block=ham$block
res.mbpca <- MBPCA(X,block, name.block=names(block))
summary(res.mbpca)
```

`summary.MBPLS` *Summary of* `MBPLS` *objects*

Description

Edits the Cumulative Explained Variance, Block Explained Variance per Dimension and Block Saliences per Dimension of a `MBPLS` object.

Usage

```
## S3 method for class 'MBPLS'
summary(object, ...)
```

Arguments

`object`	An object resulting from `MBPLS`.
`...`	further arguments passed to or from other methods.

Value

The summary.

See Also

`plot.MBPLS`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbpls <- MBPLS(X, Y, block, name.block = names(block))
summary(res.mbpls)
```

`summary.MBWCov` *Summary of* `MBWCov` *objects*

Description

Edits the Cumulative Explained Variance, Block Explained Variance per Dimension and Block Saliences per Dimension of a `MBWCov` object.

Usage

```
## S3 method for class 'MBWCov'
summary(object, ...)
```

Arguments

`object`	An object resulting from `MBWCov`.
`...`	further arguments passed to or from other methods.

Value

The summary.

See Also

`plot.MBWCov`

Examples

```
data(ham)
X=ham$X
block=ham$block
Y=ham$Y
res.mbwcov <- MBWCov(X, Y, block, name.block = names(block))
summary(res.mbwcov)
```

Index