

Package ‘ConsensusOPLS’

May 7, 2026

Title Consensus OPLS for Multi-Block Data Fusion

Description Merging data from multiple sources is a relevant approach for comprehensively evaluating complex systems. However, the inherent problems encountered when analyzing single tables are amplified with the generation of multi-block datasets, and finding the relationships between data layers of increasing complexity constitutes a challenging task. For that purpose, a generic methodology is proposed by combining the strength of established data analysis strategies, i.e. multi-block approaches and the Orthogonal Partial Least Squares (OPLS) framework to provide an efficient tool for the fusion of data obtained from multiple sources. The package enables quick and efficient implementation of the consensus OPLS model for any horizontal multi-block data structures (observation-based matching). Moreover, it offers an interesting range of metrics and graphics to help to determine the optimal number of components and check the validity of the model through permutation tests. Interpretation tools include score and loading plots, Variable Importance in Projection (VIP), functionality predict for SHAP computing, and performance coefficients such as R2, Q2, and DQ2 coefficients.

J. Boccard and D.N. Rutledge (2013) <[doi:10.1016/j.aca.2013.01.022](https://doi.org/10.1016/j.aca.2013.01.022)>.

Version 1.1.0

Depends R (>= 4.0.0), stats, utils, graphics, grDevices, methods

Imports parallel, reshape2

Suggests testthat (>= 3.0.0), knitr, ggplot2, ggrepel, plotly, psych,
DT, ComplexHeatmap

Config/testthat/edition 3

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Celine Bougel [aut] (ORCID: <<https://orcid.org/0000-0002-1598-7827>>),
 Julien Boccard [aut] (ORCID: <<https://orcid.org/0000-0001-5913-9566>>),
 Florence Mehl [aut] (ORCID: <<https://orcid.org/0000-0002-9619-1707>>),
 Marie Tremblay-Franco [fnd] (ORCID:
 <<https://orcid.org/0000-0003-4678-4456>>),
 Mark Ibberson [fnd] (ORCID: <<https://orcid.org/0000-0003-3152-5670>>),
 Van Du T. Tran [aut, cre] (ORCID:
 <<https://orcid.org/0000-0003-2074-5029>>)

Maintainer Van Du T. Tran <thuong.tran@sib.swiss>

Repository CRAN

Date/Publication 2025-02-27 08:40:02 UTC

Contents

ConsensusOPLS-package	2
ConsensusOPLS	3
ConsensusOPLS-class	5
demo_3_Omics	6
plotContribution	6
plotDQ2	7
plotLoadings	8
plotQ2	9
plotR2	10
plotScores	11
plotVIP	11
predict	12
Index	14

ConsensusOPLS-package *Consensus OPLS for Multi-Block Data Fusion*

Description

Merging data from multiple sources is a relevant approach for comprehensively evaluating complex systems. However, the inherent problems encountered when analyzing single tables are amplified with the generation of multi-block datasets, and finding the relationships between data layers of increasing complexity constitutes a challenging task. For that purpose, a generic methodology is proposed by combining the strengths of established data analysis strategies, i.e. multi-block approaches and the OPLS framework to provide an efficient tool for the fusion of data obtained from multiple sources. The package enables quick and efficient implementation of the consensus OPLS model for any horizontal multi-block data structure (observation-based matching). Moreover, it offers an interesting range of metrics and graphics to help to determine the optimal number of components and check the validity of the model through permutation tests. Interpretation tools include scores and loadings plots, as well as Variable Importance in Projection (VIP), and performance coefficients such as R2, Q2 and DQ2 coefficients.

This package uses functions from the K-OPLS package, developed by Max Bylesjo, University of Umea, Judy Fonville and Mattias Rantalainen, Imperial College.

Copyright (c) 2007-2010 Max Bylesjo, Judy Fonville and Mattias Rantalainen

This code has been extended and adapted under the terms of the GNU General Public License version 2 as published by the Free Software Foundation.

Author(s)

Maintainer: Van Du T. Tran <thuong.tran@sib.swiss> ([ORCID](#))

Authors:

- Celine Bougel <celine.bougel@laposte.net> ([ORCID](#))
- Julien Boccard <julien.boccard@unige.ch> ([ORCID](#))
- Florence Mehl <florence.mehl@sib.swiss> ([ORCID](#))

Other contributors:

- Marie Tremblay-Franco <marie.tremblay-franco@inrae.fr> ([ORCID](#)) [funder]
- Mark Ibberson <mark.ibberson@sib.swiss> ([ORCID](#)) [funder]

ConsensusOPLS

ConsensusOPLS

Description

Constructs the consensus OPLS model with the optimal number of orthogonal components for given data blocks and response, and evaluate the model quality w.r.t other models built with randomly permuted responses.

Usage

```
ConsensusOPLS(  
  data,  
  Y,  
  maxPcomp = 1,  
  maxOcomp = 5,  
  modelType = "da",  
  nperm = 100,  
  cvType = "nfold",  
  nfold = 5,  
  nMC = 100,  
  cvFrac = 4/5,  
  kernelParams = list(type = "p", params = c(order = 1)),  
  mc.cores = 1,  
  verbose = FALSE  
)
```

Arguments

<code>data</code>	A list of data blocks. Each element of the list must be of matrix type. Rows and columns can be identified (names), in which case this will be retained during analysis. Any pre-processing of the data (e.g. scaling) must be carried out before building the model.
<code>Y</code>	A vector, factor, dummy matrix or numeric matrix for the response. The type of answer given will condition the model to be used: a numeric vector for linear regression, a factor or dummy matrix for logistic regression or a discriminant model.
<code>maxPcomp</code>	Maximum number of Y-predictive components used to build the optimal model. Default, 1.
<code>maxOcomp</code>	Maximum number of Y-orthogonal components used to build the optimal model. Default, 5.
<code>modelType</code>	String for type of OPLS regression model, either <code>reg</code> for regression or <code>da</code> for discriminant analysis. Default, <code>da</code> .
<code>nperm</code>	Number of random permutations desired in response Y. Default, 100.
<code>cvType</code>	String for type of cross-validation used. Either <code>nfold</code> for n-fold cross-validation, where <code>nfold</code> is look up, or <code>mccv</code> for Monte Carlo cross-validation, or <code>mccvb</code> for Monte Carlo class-balanced cross-validation, where <code>nMC</code> and <code>cvFrac</code> are used. Default, <code>nfold</code> , i.e. <code>nMC</code> and <code>cvFrac</code> are ignored.
<code>nfold</code>	Number of folds performed in n-fold cross-validation. This can be set to the number of samples to perform Leave-One-Out cross validation. Default, 5.
<code>nMC</code>	An integer indicating the number of rounds performed when <code>cvType</code> is <code>mccv</code> or <code>mccvb</code> . Default, 100.
<code>cvFrac</code>	A numeric value indicating the fraction of observations from data used in the training set for <code>mccv</code> or <code>mccvb</code> cross-validation. Default, $4/5 = 0.8$.
<code>kernelParams</code>	List of parameters for the kernel. Either <code>p</code> for polynomial kernel, which implies specifying the order of the polynomial by the <code>order</code> parameter, or <code>g</code> for Gaussian kernel. Default, <code>list(type='p', params = c(order=1.0))</code> .
<code>mc.cores</code>	Number of cores for parallel computing. Default, 1.
<code>verbose</code>	A logical indicating if detailed information (cross validation) will be shown. Default, <code>FALSE</code> .

Value

An object of class `ConsensusOPLS` representing the consensus OPLS model fit.

Examples

```
data(demo_3_Omics)
datablocks <- lapply(
  demo_3_Omics[c("MetaboData", "MicroData", "ProteoData")], scale)
res <- ConsensusOPLS(data=datablocks,
  Y=demo_3_Omics$Y,
  maxPcomp=1, maxOcomp=2,
```

```

                                modelType='da',
                                nperm=5)
res

```

ConsensusOPLS-class ConsensusOPLS *S4* class

Description

An object returned by the ConsensusOPLS function, of class ConsensusOPLS, and representing a fitted Consensus OPLS model.

Slots

`modelType` The type of requested OPLS regression model.

`response` The provided response variable (Y).

`nPcomp` Number of Y-predictive components (latent variables) of the optimal model.

`nOcomp` Number of Y-orthogonal components (latent variables) of the optimal model.

`blockContribution` Relative contribution of each block (normalized lambda values) to the latent variables.

`scores` Representation of the samples in the latent variables of the optimal model.

`loadings` Contribution of each block's variables to the latent variables of the optimal model.

`VIP` Variable importance in projection (VIP) for each block of data, assessing the relevance of the variables in explaining the variation in the response.

`R2X` Proportion of variation in data blocks explained by the optimal model.

`R2Y` Proportion of variation in the response explained by the optimal model.

`Q2` Predictive ability of the optimal model.

`DQ2` Predictive ability of the optimal discriminant model.

`permStats` Assessment of models with permuted response.

`model` The optimal model.

`cv` Cross-validation result towards the optimal model. Contains `AllYhat` (all predicted Y values as a concatenated matrix), `cvTestIndex` (indexes for the test set observations during the cross-validation rounds), `DQ2Yhat` (total discriminant Q-square result for all Y-orthogonal components), `nOcompOpt` (optimal number of Y-orthogonal components (latent variables) for the optimal model), and `Q2Yhat` (total Q-square result for all Y-orthogonal components).

 demo_3_Omics

Three-block omics data

Description

A demonstration case study available from a public repository of the National Cancer Institute, namely the NCI-60 data set, was used to illustrate the method's potential for omics data fusion. A subset of NCI-60 data (transcriptomics, proteomics and metabolomics) involving experimental data from 14 cancer cell lines from two tissue origins, i.e. colon and ovary, was used. The object proposed in this package contains, in a list, all the information needed to make a model: the three data blocks, a list of observation names (samples) and the binary response matrix Y .

Author(s)

Boccard & Rutledge

References

J. Boccard and D.N. Rutledge. A consensus OPLS-DA strategy for multiblock Omics data fusion. *Analytica Chimica Acta*, 769, 30-39, 2013.

 plotContribution

Block contribution plot

Description

Plot of relative contribution of each data block in the optimal model.

Usage

```
plotContribution(object, col = NULL, ...)
```

```
## S4 method for signature 'ConsensusOPLS'
plotContribution(object, col = NULL, ...)
```

Arguments

object	An object of class ConsensusOPLS.
col	A vector of color codes or names, one for each block. Default, NULL, 2 to number of blocks + 1.
...	barplot arguments.

Value

No return value, called for side effects.

`plotDQ2`*DQ2 plot*

Description

Plot of DQ2 of models with permuted response.

Usage

```
plotDQ2(  
  object,  
  breaks = 10,  
  xlab = "DQ2",  
  main = "DQ2 in models with permuted response",  
  col = "blue",  
  lty = "dashed",  
  ...  
)
```

```
## S4 method for signature 'ConsensusOPLS'  
plotDQ2(  
  object,  
  breaks = 10,  
  xlab = "DQ2",  
  main = "DQ2 in models with permuted response",  
  col = "blue",  
  lty = "dashed",  
  ...  
)
```

Arguments

<code>object</code>	An object of class <code>ConsensusOPLS</code> .
<code>breaks</code>	See <code>hist</code> .
<code>xlab</code>	See <code>hist</code> .
<code>main</code>	See <code>hist</code> .
<code>col</code>	A color code or name for DQ2 in the optimal model. Default, 2. See <code>abline</code> .
<code>lty</code>	A line type code or name for DQ2 in the optimal model. Default, 2. See <code>abline</code> .
<code>...</code>	<code>hist</code> arguments.

Value

No return value, called for side effects.

plotLoadings

Loading plot

Description

Plot of variable loadings in the optimal model.

Usage

```
plotLoadings(
  object,
  comp1 = "p_1",
  comp2 = "o_1",
  blockId = NULL,
  col = NULL,
  pch = NULL,
  ...
)

## S4 method for signature 'ConsensusOPLS'
plotLoadings(
  object,
  comp1 = "p_1",
  comp2 = "o_1",
  blockId = NULL,
  col = NULL,
  pch = NULL,
  ...
)
```

Arguments

object	An object of class ConsensusOPLS.
comp1	Latent variable for X-axis. Default, the first predictive component, p_1.
comp2	Latent variable for Y-axis. Default, the first orthogonal component, o_1.
blockId	The positions or names of the blocks for the plot. Default, NULL, all.
col	A vector of color codes or names, one for each block. Default, NULL, 2 to length(blockId)+1.
pch	A vector of graphic symbols, one for each block. Default, NULL, 1 to length(blockId).
...	plot arguments.

Value

No return value, called for side effects.

`plotQ2`*Q2 plot*

Description

Plot of Q2 of models with permuted response.

Usage

```
plotQ2(  
  object,  
  breaks = 10,  
  xlab = "Q2",  
  main = "Q2 in models with permuted response",  
  col = "blue",  
  lty = "dashed",  
  ...  
)
```

```
## S4 method for signature 'ConsensusOPLS'  
plotQ2(  
  object,  
  breaks = 10,  
  xlab = "Q2",  
  main = "Q2 in models with permuted response",  
  col = "blue",  
  lty = "dashed",  
  ...  
)
```

Arguments

<code>object</code>	An object of class <code>ConsensusOPLS</code> .
<code>breaks</code>	See <code>hist</code> .
<code>xlab</code>	See <code>hist</code> .
<code>main</code>	See <code>hist</code> .
<code>col</code>	A color code or name for Q2 in the optimal model. Default, 2. See <code>abline</code> .
<code>lty</code>	A line type code or name for Q2 in the optimal model. Default, 2. See <code>abline</code> .
<code>...</code>	<code>hist</code> arguments.

Value

No return value, called for side effects.

`plotR2`*R2 plot*

Description

Plot of R2 of models with permuted response.

Usage

```
plotR2(  
  object,  
  breaks = 10,  
  xlab = "R2",  
  main = "R2 in models with permuted response",  
  col = "blue",  
  lty = "dashed",  
  ...  
)
```

```
## S4 method for signature 'ConsensusOPLS'  
plotR2(  
  object,  
  breaks = 10,  
  xlab = "R2",  
  main = "R2 in models with permuted response",  
  col = "blue",  
  lty = "dashed",  
  ...  
)
```

Arguments

<code>object</code>	An object of class <code>ConsensusOPLS</code> .
<code>breaks</code>	See <code>hist</code> .
<code>xlab</code>	See <code>hist</code> .
<code>main</code>	See <code>hist</code> .
<code>col</code>	A color code or name for R2 in the optimal model. Default, 2. See <code>abline</code> .
<code>lty</code>	A line type code or name for R2 in the optimal model. Default, 2. See <code>abline</code> .
<code>...</code>	<code>hist</code> arguments.

Value

No return value, called for side effects.

plotScores	<i>Score plot</i>
------------	-------------------

Description

Plot of samples in the space of latent variables of the optimal model.

Usage

```
plotScores(object, comp1 = "p_1", comp2 = "o_1", col = NULL, pch = 19, ...)
```

```
## S4 method for signature 'ConsensusOPLS'
```

```
plotScores(object, comp1 = "p_1", comp2 = "o_1", col = NULL, pch = 19, ...)
```

Arguments

object	An object of class ConsensusOPLS.
comp1	Latent variable for abscissa. Default, the first predictive component, p_1.
comp2	Latent variable for ordinate. Default, the first orthogonal component, o_1.
col	A vector of color codes or names. Default, NULL, generated following the response.
pch	Graphic symbol. Default, 19.
...	plot arguments.

Value

No return value, called for side effects.

plotVIP	<i>VIP plot</i>
---------	-----------------

Description

Plot of VIP versus variable loadings in the optimal model.

Usage

```
plotVIP(
  object,
  comp1 = "p_1",
  comp2 = "p",
  blockId = NULL,
  col = NULL,
  pch = NULL,
```

```

    xlab = NULL,
    ylab = NULL,
    ...
)

## S4 method for signature 'ConsensusOPLS'
plotVIP(
  object,
  comp1 = "p_1",
  comp2 = "p",
  blockId = NULL,
  col = NULL,
  pch = NULL,
  xlab = NULL,
  ylab = NULL,
  ...
)

```

Arguments

object	An object of class ConsensusOPLS.
comp1	Latent variable for loadings on Y-axis. Default, the first predictive component, p_1.
comp2	Latent variable for VIPs on X-axis. Default, the predictive component, p.
blockId	The positions or names of the blocks for the plot. Default, NULL, all.
col	A vector of color codes or names, one for each block. Default, NULL, 2 to length(blockId)+1.
pch	A vector of graphic symbols, one for each block. Default, NULL, 1 to length(blockId).
xlab	X-axis label. Default, NULL, Loading on comp.
ylab	Y-axis label. Default, NULL, VIP on comp.
...	plot arguments.

Value

No return value, called for side effects.

predict

Model prediction

Description

Predicts the response on new data with a fitted model.

Usage

```
predict(object, newdata = NULL, n0comp = NULL)
```

```
## S4 method for signature 'ConsensusOPLS'  
predict(object, newdata = NULL, n0comp = NULL)
```

Arguments

<code>object</code>	An object of class <code>ConsensusOPLS</code> .
<code>newdata</code>	A list of data frames of new data to predict. If omitted, the fitted values in <code>object</code> are returned.
<code>n0comp</code>	Number of Y-orthogonal components to consider. Default, <code>NULL</code> , the number of Y-orthogonal components in the optimal model.

Index

ConsensusOPLS, [3](#)
ConsensusOPLS-class, [5](#)
ConsensusOPLS-package, [2](#)

demo_3_Omics, [6](#)

plotContribution, [6](#)
plotContribution, ConsensusOPLS-method
(plotContribution), [6](#)

plotDQ2, [7](#)
plotDQ2, ConsensusOPLS-method (plotDQ2),
[7](#)

plotLoadings, [8](#)
plotLoadings, ConsensusOPLS-method
(plotLoadings), [8](#)

plotQ2, [9](#)
plotQ2, ConsensusOPLS-method (plotQ2), [9](#)

plotR2, [10](#)
plotR2, ConsensusOPLS-method (plotR2), [10](#)

plotScores, [11](#)
plotScores, ConsensusOPLS-method
(plotScores), [11](#)

plotVIP, [11](#)
plotVIP, ConsensusOPLS-method (plotVIP),
[11](#)

predict, [12](#)
predict, ConsensusOPLS-method (predict),
[12](#)