Package ‘cocorresp’

May 7, 2021

Type Package
Title Co-Correspondence Analysis Methods
Version 0.4-3
Date 2021-05-07
Depends vegan (>= 2.5-0), R (>= 3.4.0)
Imports stats, graphics, utils, grDevices
Suggests testthat, knitr, rmarkdown
Maintainer Gavin L. Simpson <ucfagls@gmail.com>
Description Fits predictive and symmetric co-correspondence analysis (CoCA) models to relate one data matrix to another data matrix. More specifically, CoCA maximises the weighted covariance between the weighted averaged species scores of one community and the weighted averaged species scores of another community. CoCA attempts to find patterns that are common to both communities.
License GPL-2
URL https://gavinsimpson.github.io/cocorresp/
BugReports https://github.com/gavinsimpson/cocorresp/issues
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2021-05-07 08:50:08 UTC

R topics documented:
cocorresp-package ................................................. 2
beetles ................................................................. 3
biplot-methods ....................................................... 3
bryophyte ............................................................ 6
coca ................................................................. 7
Description

Fits predictive and symmetric co-correspondence analysis (CoCA) models to relate one data matrix to another data matrix. More specifically, CoCA maximises the weighted covariance between the weighted averaged species scores of one community and the weighted averaged species scores of another community. CoCA attempts to find patterns that are common to both communities.

Details

The main interface function is `coca` which accepts a formula or two community data matrices. An appropriate formula is \( Y \sim ., data = X \) and the associated data object from which \( Y \) will be looked up. The method argument is used to select from the two forms of CoCA: method = "predictive" for predictive CoCA (the default), and method = "symmetric" for symmetric CoCA.

Author(s)


Maintainer: Gavin L. Simpson <ucfagls@gmail.com>
### beetles

**Carabid beetles and vascular plants in Dutch roadside verges**

#### Description

Counts of 126 beetle taxa and abundances of 173 vascular plant taxa (expressed on the 1-9 van der Maarel scale) in 30 road-side verges in the Netherlands.

#### Usage

```r
data(beetles)
data(plants)
data(verges)
```

#### Details

This is the complete dataset of Raemakers et al (2001). ter Braak and Schaffers (2004) only analyse a subset of 91 beetle taxa.

#### Source


#### Examples

```r
data(beetles)
data(plants)
```

### biplot-methods

**Biplots of co-correspondence analysis models**

#### Description

Produces biplots of the response and predictor from the results of a co-correspondence analysis models.
Usage

```r
## S3 method for class 'symcoca'
biplot(x,
    which = "y1",
    choices = 1:2,
    benzecri = TRUE,
    type = NULL,
    xlim = NULL,
    ylim = NULL,
    col.species = "red",
    col.sites = "black",
    pch.species = 3,
    pch.sites = 1,
    cex = 0.7,
    main = "",
    sub = "",
    ylab, xlab,
    ann = par("ann"),
    axes = TRUE,
    ...)  
```

```r
## S3 method for class 'predcoca'
biplot(x,
    which = "response",
    choices = 1:2,
    type = NULL,
    xlim = NULL,
    ylim = NULL,
    col.species = "red",
    col.sites = "black",
    pch.species = 3,
    pch.sites = 1,
    cex = 0.7,
    main = "",
    sub = "",
    ylab, xlab,
    ann = par("ann"),
    axes = TRUE,
    ...)  
```

Arguments

- `x` an object of class "symcoca", the result of a call to `symcoca`.
- `which` character; should the response or predictor scores be plotted. Can be specified in several ways: `response` choices are one from c("y", "Y", "y1", "response"); `predictor` choices are one from c("x", "X", "y2", "predictor").
- `choices` a vector of length 2 indicating which predictive CoCA axes to plot.
benzecri logical, should a Benzecri plot be drawn? Such plots draw species scores, scaled by the quarter root of the respective eigenvalues, with unscaled site scores. A Benzecri plot is the recommended biplot for symmetric CoCA. See scores.symcoca.

type one of "points", or "text". Determines how the site and species scores are displayed. If type = "points", scores are plotted as points. If type = "text", then the row names of the scores matrices are plotted.

xlim, ylim limits for the x and y axes. If non supplied, suitable limits will be determined from the data.

col.species, col.sites, pch.species, pch.sites colours and plotting characters used when plotting the species and sites scores.

cex numeric; scaling factor when drawing points or text labels.

xlab, ylab labels for the x and y axes. If non supplied suitable labels are formed from the result object.

main, sub the main and sub titles for the plot.

ann logical, if TRUE plots are annotated and not if FALSE, currently ignored.

axes a logical value indicating whether axes and plot border should be drawn on the plot.

... other graphical parameters as in 'par' may also be passed as arguments.

Author(s)

Gavin L. Simpson.

References


See Also
coca, plot.default

Examples

```r
## symmetric CoCA
data(beetles)
data(plants)

## log transform the beetle data
beetles <- log(beetles + 1)

## fit the model
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")

## draw a biplot of the beetle results
biplot(bp.sym)

## biplot of both - Fig 1 in ter Braak & Schaffers (2004)
```
layou

```r
layout(matrix(1:2, ncol = 2))
biplot(bp.sym, which = "y1", main = "Beetles")
biplot(bp.sym, which = "y2", main = "Plants")
layout(1)
```

## predictive CoCA
bp.pred <- coca(beetles ~ ., data = plants)

## draw a biplot of the response
biplot(bp.pred)

## recreate Fig 3 in ter Braak & Schaffers (2004)
layout(matrix(1:2, ncol = 2))
biplot(bp.pred, which = "response", main = "Beetles")
biplot(bp.pred, which = "predictor", main = "Plants")
layout(1)

---

**bryophyte**

*Bryophytes and vascular plants in Carpathian spring meadows*

**Description**

The data consist of observations on 30 bryophyte and 123 vascular plant species in 70 spring meadows (sites, samples). The species data are a subset of only those species occurring in five or more meadows.

**Usage**

```r
data(bryophyte)
data(vascular)
```

**Source**


**Examples**

```r
data(bryophyte)
data(vascular)
```
**Description**

coca is used to fit Co-Correspondence Analysis (CoCA) models. It can fit predictive or symmetric models to two community data matrices containing species abundance data.

**Usage**

coca(y, ...)

## Default S3 method:
coca(y, x, method = c("predictive", "symmetric"),
reg.method = c("simpls", "eigen"), weights = NULL,
n.axes = NULL, symmetric = FALSE, quiet = FALSE, ...)

## S3 method for class 'formula'
coca(formula, data, method = c("predictive", "symmetric"),
reg.method = c("simpls", "eigen"), weights = NULL,
n.axes = NULL, symmetric = FALSE, quiet = FALSE, ...)

**Arguments**

- **y**
a data frame containing the response community data matrix.
- **x**
a data frame containing the predictor community data matrix.
- **formula**
a symbolic description of the model to be fit. The details of model specification are given below.
- **data**
an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which coca is called.
- **method**
a character string indicating which co-correspondence analysis method to use. One of "predictive" (default), or "symmetric", can be abbreviated.
- **reg.method**
One of "simpls" (default) or "eigen". If method is "predictive" then reg.method controls whether the co-correspondence analysis should be fitted using the SIMPLS algorithm or via an eigen analysis.
- **weights**
a vector of length nrow(y) of user supplied weights for $R_0$. If weights = NULL (default) then the weights are determined from y (default) or x and y (symmetric = TRUE only).
- **n.axes**
the number of CoCA axes to extract. If missing (default) the n.axes is

$$\min(n\text{col}(y), n\text{col}(x), n\text{row}(y), n\text{row}(x)) - 1$$


symmetric

if method is "symmetric" then symmetric determines whether weights for $R_0$
are symmetric and taken as the average of the row sums of $x$ and $y$ (symmetric
= TRUE). If symmetric = FALSE (default) then the weights $R_0$ are taken as the
row sums of $y$ unless user defined weights are provided via argument weights.
Ignored if method is "predictive".

quiet

logical; suppress messages due to removal of species with no data.

... additional arguments to be passed to lower level methods.

Details
coca is the main user-callable function.

A typical model has the form response ~ terms where response is the (numeric) response data
frame and terms is a series of terms which specifies a linear predictor for response. A typical
form for terms is ., which is shorthand for "all variables" in data. If . is used, data must also be
provided. If specific species (variables) are required then terms should take the form spp1 + spp2 + spp3.
The default is to fit a predictive CoCA model using SIMPLS via a modified version of simpls.fit
from package pls. Alternatively, reg.method = "eigen" fits the model using an older, slower
eigen analysis version of the SIMPLS algorithm. reg.method = "eigen" is about 100% slower
than reg.method = "simpls".

Value
coca returns a list with method and reg.method determining the actual components returned.

nam.dat

list with components namY and namX containing the names of the response and
the predictor(s) respectively.

call

the matched call.

method

the CoCA method used, one of "predictive" or "symmetric".

scores

the species and site scores of the fitted model.

loadings

the site loadings of the fitted model for the response and the predictor. (Predictive
CoCA via SIMPLS only.)

fitted

the fitted values for the response. A list with 2 components Yhat (the fitted values
on the original scale) and Yhat1 (the fitted values on the chi-square transformed scale). (Predictive CoCA via SIMPLS only.)

varianceExp

list with components Yblock and Xblock containing the variances in the re-
response and the predictor respectively, explained by each fitted PLS axis. (Predictive
CoCA via SIMPLS only.)

totalVar

list with components Yblock and Xblock containing the total variance in the
response and the predictor respectively. (Predictive CoCA via SIMPLS only.)

lambda

the Eigenvalues of the analysis.

n.axes

the number of fitted axes

Ychi

a list containing the mean-centered chi-square matrices for the response (Ychi1)
and the predictor (Ychi2). (Predictive CoCA only.)

$R_0$

the (possibly user-supplied) row weights used in the analysis.
coca

X Matrix (symmetric CoCA only).

Residuals of a symmetric model (symmetric CoCA only).

Inertia list with components total and residual containing the total and residual inertia for the response and the predictor (symmetric CoCA only).

Rowsum a list with the row sums for the response (rsum1) and the predictor (rsum2) (symmetric CoCA only).

Colsum a list with the column sums for the response (csum1) and the predictor (csum2) (symmetric CoCA only).

Author(s)


References


See Also

crossval for cross-validation and permutest.coca for permutation test to determine the number of PLS axes to retain in for predictive CoCA.

summary.predcoca and summary.symcoca for summary methods.

Examples

## symmetric CoCA
data(beetles)
log transform the bettle data
beetles <- log(beetles + 1)
data(plants)
fit the model
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")
bp.sym
summary(bp.sym)
biplot(bp.sym) # produces a Benzecri biplot

## extract eigenvalues of the analysis
eigenvals(bp.sym)

## correlations between beetle and plant score scores on Co-CA axes
corAxis(bp.sym)

## predictive CoCA using SIMPLS and formula interface
bp.pred <- coca(beetles ~ ., data = plants)
## should retain only the useful PLS components for a parsimonious model
## Leave-one-out crossvalidation - this takes a while
crossval(beetles, plants)
## so 2 axes are sufficient
## permutation test to assess significant PLS components - takes a while
bp.perm <- permutest(bp.pred, permutations = 99)
bp.perm

## agrees with the Leave-one-out cross-validation
## refit the model with only 2 PLS components
bp.pred <- coca(beetles ~ ., data = plants, n.axes = 2)
bp.pred
summary(bp.pred)
biplot(bp.pred)  # plots correct scores or loadings

## predictive CoCA using Eigen-analysis
data(bryophyte)
data(vascular)
carp.pred <- coca(y = bryophyte, x = vascular, reg.method = "eigen")
carp.pred

## determine important PLS components - takes a while
crossval(bryophyte, vascular)
(carp.perm <- permutest(carp.pred, permutations = 99))

## 2 components again, refit
carp.pred <- coca(y = bryophyte, x = vascular, 
                 reg.method = "eigen", n.axes = 2)
carp.pred
## drawn biplot
biplot(carp.pred)

---

### coinertia

**Co-inertia analysis**

**Description**

Performs a co-inertia of the triplets \((Q_1, K_1, R_0)\) and \((Q_2, K_2, R_0)\).

**Usage**

coinertia(y, ...)

## Default S3 method:
coinertia(y, x, n.axes = NULL, weights = NULL,
          symmetric = FALSE, ...)
Arguments

- **y, x**: Matrices or data frames of the two data sets for which axes of covariance are sought.
- **n.axes**: Numeric; the number of coinertia axes to retain.
- **weights**: A vector of weights for the samples. If NULL, weights are chosen from the row sums of y (symmetric = FALSE) or the average of the row sums of y and x (symmetric = TRUE).
- **symmetric**: Logical; should a symmetric set of weights be used. See Details.
- **...**: Arguments passed to other functions. Currently ignored.

Value

An object of class "coinertia", which is a list with the following components:

- **scores**: A list of ordination scores, with components species and sites, each of which is a list with components Y and X that refer to the scores for the input matrices y and x respectively.
- **weights**: A numeric vector of row weights used in the analysis.
- **lambda**: A numeric vector of Eigenvalues.
- **n.axes**: Numeric; the number of coinertia axes extracted.
- **symmetric**: Logical; was a symmetric analysis performed?
- **call**: The matched call.

Author(s)


References


See Also

symcoca for the function that calls fitCoinertia and coinertial for co-inertia analysis using identity matrices for $K_1$, $K_2$, and $R_0$.

Examples

data(beetles, plants)
coin <- coinertia(beetles, plants)
coin
Coinertia analysis with identity matrices

Description
Performs a co-inertia of the triplets \((Q_1, K_1, R_0)\) and \((Q_2, K_2, R_0)\) with identity matrices \(K_1, K_2, R_0\).

Usage
\[
\text{coinertiaI}(X, Y, \text{fast} = \text{TRUE})
\]

Arguments
- \(X\) Species matrix \(X\).
- \(Y\) Species Matrix \(Y\).
- \(\text{fast}\) If "TRUE" only return the row scores of \(Y\).

Details
Argument \(\text{fast}\) is used to return only the row scores of \(Y\) in function \text{permutest.coca}, which speeds the permutation test considerably.

Value
If \(\text{fast} = \text{TRUE}\), a matrix of row scores for matrix \(Y\) (see scores below). If \(\text{fast} = \text{FALSE}\) a list with the following components:
- \(\text{weights}\) A list with components \(X\) and \(Y\) containing the left and right singular vectors respectively of the SVD on the triplets.
- \(\text{scores}\) A list with components \(X\) and \(Y\), containing the row scores of the \(X\) and \(Y\) species matrices. These are the result of a matrix multiplication of \(X\) by the left singular vectors and \(Y\) by the right singular vectors.
- \(\lambda\) the Eigenvalues of the analysis (the square of the singular values from the SVD).
- \(\text{call}\) the matched function call.

Note
This function is not meant to be called directly by the user. If you wish to use it study the code in \text{permutest.coca} to see how it should be called.

Author(s)

References
corAxis

See Also
   coinertia

Description
   Calculates the Pearson product-moment correlation coefficient for the site scores of ordination axes.

Usage
   corAxis(x, ...)
   ## Default S3 method:
   corAxis(x, ...)
   ## S3 method for class 'symcoca'
   corAxis(x, axes = NULL, ...)

Arguments
   x                      an ordination object. Only methods for objects of class symcoca are currently available.
   axes                   numeric; the number of axes to calculate the correlation coefficients for. If NULL, coefficients for all axes are returned.
   ...                    arguments to be passed on to other methods.

Value
   A named vector containing the correlation coefficients for the requested axes.

Note
   The arguments for cor are hard coded at their defaults, see cor for details. A more flexible version is planned that will allow arguments to be passed to cor.

Author(s)
   Gavin L. Simpson

See Also
   cor, for the main analysis function.
Examples

```r
## load some data
data(beetles)
data(plants)

## log transform the beetle data
beetles <- log(beetles + 1)

## symmetric Co-CA model
beetles.sym <- coca(beetles ~ ., data = plants, method = "symmetric")

## correlations between axes
corAxis(beetles.sym)
```

crossval

Cross-validation for predictive Co-Correspondence Analysis models

Description

Performs a leave-one-out cross-validation of a predictive Co-Correspondence Analysis model.

Usage

```r
crossval(y, x, n.axes = min(dim(x), dim(y)) - 1,
          centre = TRUE, verbose = TRUE)
```

```r
## S3 method for class 'crossval'
summary(object, axes = c(1:min(6, object$n.axes)), ...)
```

Arguments

- `y` the response species matrix.
- `x` the predictor species matrix.
- `n.axes` the number of axes to calculate the leave-one-out cross-validation for. Default is to perform the CV for all extractable axes.
- `centre` centre `y` and `x` during analysis? Currently ignored as it may not be necessary.
- `verbose` if `TRUE`, the default, print information on the progress of the cross-validation procedure.
- `object` an object of class `crossval` as returned by `crossval`.
- `axes` the number of axes to summarise results for.
- `...` further arguments to `print` - currently ignored.
Details

Performs a leave-one-out cross-validation of a predictive Co-Correspondence Analysis model. It can be slow depending on the number of columns in the matrices, and of course the number of sites.

Value

Returns a large list with the following components:

- `dimx, dimy` the dimensions of the input matrices x and y respectively.
- `press0` the `press0` statistic.
- `n.axes` the number of axes tested.
- `CVfit` the cross-validatory fit.
- `varianceExp` list with components `Yblock` and `Xblock` containing the variances in the response and the predictor respectively, explained by each fitted PLS axis.
- `totalVar` list with components `Yblock` and `Xblock` containing the total variance in the response and the predictor respectively.
- `nam.dat` list with components `namY` and `namX` containing the names of the response and the predictor(s) respectively.
- `call` the R call used.

Note

This function is not a bit out-of-date compared to some of the other functions. It should have a formular interface like `coca` or work on the results from `coca`, although that will have to be altered to store a copy of the data?

Author(s)


See Also

The model fitting function `coca`

Examples

```r
## load the data sets
data(beetles)
data(plants)

## log transform the bettle data
beetles <- log(beetles + 1)

## predictive CoCA using SIMPLS and formula interface
bp.pred <- coca(beetles ~ ., data = plants)
## should retain only the useful PLS components for a
## parsimonious model
```
envfit.coca

## Leave-one-out crossvalidation - this takes a while
## Not run:
crossval(beetles, plants)
## End(Not run)
## so 2 axes are sufficient

envfit.coca  

*Fit an environmental vector or factor to a Co-CA ordination*

**Description**

The function fits environmental vectors or factors to a Co-CA ordination. The projections of points onto vectors have maximum correlation with corresponding environmental variables, and the factors show the averages of factor levels.

**Usage**

```r
## S3 method for class 'coca'
envfit(ord, env, which = c("response", "predictor"),
      choices = c(1, 2), scaling = FALSE, w, na.rm = FALSE,
      strata = NULL, permutations = 999, ...)
```

**Arguments**

- `ord`  
a Co-CA ordination object, the result of a call to `coca`.
- `env`  
a data frame, matrix or vector of environmental/external variable(s) to be fitted to the ordination. The variables may be of a mixed type (factors and continuous variables) in a data frame.
- `which`  
character; which of the response or predictor ordinations should be used during fitting of vectors and factors.
- `choices`  
numeric; the axes to which vectors and factors are fitted.
- `scaling`  
logical; should sacling be applied. See `scores.symcoca`.
- `w`  
weights used in fitting vectors and factors.
- `na.rm`  
Remove points with missing values in ordination scores or environmental variables. The operation is casewise; the whole row of data is removed if there is a missing value and `na.rm = TRUE`.
- `strata`  
An integer vector or factor specifying the strata for permutation. If supplied, observations are permuted only within the specified strata.
- `permutations`  
Number of permutations for assessing significance of vectors or factors. Set to 0 to skip permutations.
- `...`  
Arguments passed to `vectorfit` and `factorfit`. 

### fitCoinertia

**Description**

Performs a co-inertia of the triplets \((Q_1, K_1, R_0)\) and \((Q_2, K_2, R_0)\).

**Details**

See `envfit` for details of the method.

**Value**

Returns an object of class `envfit`.

**Author(s)**

Gavin L. Simpson. The code interfaces with and uses code from `envfit` for the main computations, which was written by Jari Oksanen.

**See Also**

`coca` for fitting models. `envfit` for details of the generic function and the computations performed.

**Examples**

```r
## symmetric CoCA
data(beetles)
data(plants)

## log transform the bettle data
beetles <- log(beetles + 1)

## fit the model
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")

## load the environmental data
data(verges)

## fit vectors for the environmental data
sol <- envfit(bp.sym, verges, which = "response")
sol

## plot the response matrix and the fitted vectors
biplot(bp.sym, which = "y1")
plot(sol)
```
Usage

fitCoinertia(X, Dp, Y, Dq, Dn, n.axes)

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

Arguments

- **X** \( Q_1 \), matrix of expected abundances under row-column independence in the original \( Y \) species matrix when treated as a contingency table.
- **Dp** \( K_1 \), species (column) weights for \( X \).
- **Y** \( Q_2 \), matrix of expected abundances under row-column independence in the original \( X \) species matrix when treated as a contingency table.
- **Dq** \( K_2 \), species (column) weights for \( Y \).
- **Dn** site weights \( R_0 \).
- **n.axes** number of axes to calculate the co-inertia analysis for.
- **object** an object of class coinertia.
- **axes** the number of axes to display when printing.
- **...** arguments passed to other functions. Currently ignored.

Value

A list with the following components:

- **U1** column weights of \( X \).
- **U2** column weights of \( Y \).
- **X1** rowscores of \( X \).
- **X2** rowscores of \( Y \).
- **lambda** the Eigenvalues (squares of the singular values).
- **n.axes** number of axes requested.
- **call** the matched function call.

Note

This function is not yet meant to be called directly by the user. If you wish to use it directly, see the function definition for symcoca which demonstrates how to prepare the relevant input matrices.

Note that in this function, \( X \) corresponds to the input matrix \( y \) and \( Y \) corresponds to the input matrix \( x \) in symcoca. Confusing! This will be changed in a future release but for now the arguments follow those of the original Matlab code - perhaps a little too closely!
Author(s)


References


See Also

symcoca for the function that calls fitCoinertia and coinertiaI for co-inertia analysis using identity matrices for $K_1$, $K_2$, and $R_0$

fitted.symcoca Fitted values of a Symmetric Co-Correspondence analysis model.

Description

Calculates and extracts the fitted values of a Symmetric Co-Correspondence analysis model.

Usage

## S3 method for class 'symcoca'
fitted(object, which = c("y1","y2"), ...)

Arguments

object an object of class "symcoca"

which character; should the response or predictor scores be plotted. Can be specified in several ways: response choices are one from c("y", "Y", "y1", "response"); predictor choices are one from c("x", "X", "y2", "predictor").

... arguments to be passed to other methods.

Value

A list with the following components:

Y the fitted values for the “response” matrix.

X the fitted values for the “predictor” matrix.

nam.dat a vector containing the names of the “response” and “predictor” matrices respectively. Used for printing the results.
Note

This function needs an update and to allow option to restrict fitted values to specified axes, and the names of the returned objects need making more obvious!

Author(s)


References


See Also

The model fitting function `coca`

Examples

```r
## symmetric CoCA
data(beetles)
data(plants)

## log transform the bettle data
beetles <- log(beetles + 1)

## fit the model
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")

## fitted values
bp.fit <- fitted(bp.sym)
bp.fit

## fitted values for beetles only
beetle.fit <- fitted(bp.sym, which = "y1")
```

Description

Extract CoCA species loadings from fitted objects.
Usage

loadings(x, ...)

## S3 method for class 'predcoca'
loadings(x, choices = c(1, 2),
          which = c("response", "predictor"), ...)

## S3 method for class 'symcoca'
loadings(x, choices = c(1, 2),
          which = c("y1", "y2"), ...)

Arguments

x          an object resulting from a call to coca.
choices    numeric; vector of Co-CA axes to extract loadings for.
which      character; should the response or predictor scores be plotted. Can be specified in
            several ways: response choices are one from c("y", "Y", "y1", "response");
            predictor choices are one from c("x", "X", "y2", "predictor").
...        additional arguments to be passed to lower level methods.

Details

loadings() is an extractor function to access the loadings of a fitted CoCA model.

This is a generic function, replacing the loadings function, which is preserved as the exported
default S3 method. Methods are provided for both predictive and symmetric CoCA.

Value

A list of data frames or a single data frame depending on other arguments.

Author(s)

Gavin L. Simpson

See Also

coca for how to fit CoCA models.

Examples

## symmetric CoCA
data(beetles)
## log transform the bettle data
beetles <- log(beetles + 1)
data(plants)
## fit the model
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")
```r
## extract the loadings
lds <- loadings(bp.sym)
```

---

**mcChi**

*Standardised chi-square residuals*

### Description

Scales a matrix, \( Y \), to its standardised chi-square residuals \((o - e)/\sqrt{e}\) (if \( R_0 = R \), where \( R \) contains the row sums of matrix \( Y \)) so that further analysis can be unweighted.

### Usage

```r
cmcChi(Y, R0, eps = 1e-06)
```

### Arguments

- **Y**: A matrix for which standardised chi-square residuals are to be calculated.
- **R0**: Row weights.
- **eps**: Tolerance - leave as default.

### Details

This function implements equation 8 of ter Braak and Schaffers (2004) by firstly applying equation 7 to form matrix \( Q \) using row and column sums of \( Y \) as weights, and, secondly, by applying equation 8 to form a matrix of standardised chi-square residuals from \( Q \) by pre-multiplication of \( Q \) by \( \sqrt{R_0} \) and post-multiplication of \( Q \) by \( \sqrt{K} \), where \( K \) is the column sums of \( Y \).

### Value

A list with the following components:

- **Ychi**: The matrix of standardised chi-squared residuals of \( Y \).
- **K**: The column sums (\( K \)) of \( Y \) divided by \( \text{sum}(K) \).

### Note

This function is not intended for casual use by users.

### Author(s)


### References

Permutation test for predictive co-correspondence analysis models

Description

A permutation test for predictive co-correspondence analysis models to assess the significance of each CoCA ordination axes.

Usage

```r
## S3 method for class 'coca'
permutest(x, R0 = NULL, permutations = 99,
         n.axes = x$n.axes, verbose = TRUE, ...)

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

Arguments

- **x**: an object of class "predcoca".
- **R0**: row weights to use in the analysis. If missing, the default, these are determined from x.
- **permutations**: the number of permutations to perform.
- **n.axes**: The number of axes to test. Defaults to the number of axes stated in x$n.axes.
- **verbose**: if TRUE, the default, print information on the progress of the permutation test procedure.
- **object**: an object of class "permutest.coca".
- **...**: arguments to be passed to other methods.

Details

An alternative approach to cross-validation (see `crossval`) to select the number of axes to retain in a predictive co-correspondence analysis is to test the statistical significance of each ordination axis using permutation tests.

The test statistic used is the $F$-ratio based on the fit of the first axis to the response data (ter Braak and Smilauer 2002). The second and subsequent axes are tested by treating previous axes as co-variables.

To be precise, this approach does not test the significance of SIMPLS axes, but those of NIPALS-PLS axes (ter Braak and de Jong 1998).
Value

A list with the following components:

- **pval**: a vector of *P*-values for each ordination axis.
- **permstat**: a vector of values for the test statistic for each axis.
- **total.inertia**: the total inertia in the response matrix.
- **inertia**: a vector containing the residualised inertia. This is the total inertia in the response after removing the inertia explained by all previous axes. For the first CoCA axis this is, by definition, the total inertia in the response.
- **fitax**: a vector containing the amount of inertia in the response matrix explained by each ordination axis.
- **pcent.fit**: a vector containing the fit of each axis to the response as a percentage of the total inertia (variance).
- **n.axes**: the number of axes in the ordination.
- **call**: the matched call.

Warning

This function is **slow**. Beware setting argument permutations higher than the default. Determine how long it takes for the default 99 permutations to complete before going crazy and asking for thousands of permutations - you’ve been warned, have a good book to hand.

Note

Argument \( R_0 \) is provided for compatibility with the original MATLAB code. The R usage paradigm makes this argument redundant in the current code and it may be invalid to supply different row weights \( (R_0) \) as \( R_0 \). This argument will likely be removed in future versions.

Author(s)


References


See Also

coca, for the model fitting function, **crossval**, for a leave-one-out cross-validation procedure, which is the preferred way to select axes in a predictive co-correspondence analysis.
Examples

```r
## load some data
data(beetles)
data(plants)

## log transform the bettle data
beetles <- log(beetles + 1)
## predictive CoCA using SIMPLS and formula interface
bp.pred <- coca(beetles ~ ., data = plants)

## should retain only the useful PLS components for a parsimonious model

## Leave-one-out crossvalidation - this takes a while
crossval(beetles, plants)
## so 2 axes are sufficient

## permutation test
## (Testing the first 2 axes & only 50 perms for speed.)
bp.perm <- permutest(bp.pred, permutations = 50, n.axes = 2)
bp.perm
```

---

**plot.predcoca**  
*Biplots for predictive co-correspondence analysis*

**Description**

Produces biplots of the response and predictor from the results of a predictive co-correspondence analysis.

**Usage**

```r
## S3 method for class 'predcoca'
plot(x,
     which = "response",
     choices = 1:2,
     display = c("species", "sites"),
     type,
     xlim = NULL,
     ylim = NULL,
     main = "",
     sub = "",
     ylab, xlab,
     ann = par("ann"),
     axes = TRUE,
     ...)```
Arguments

- **x**: an object of class "predcoca", the result of a call to `coca`.
- **which**: character; should the response or predictor scores be plotted. Can be specified in several ways: *response* choices are one from c("y","Y","y1","response"); *predictor* choices are one from c("x","X","y2","predictor").
- **choices**: a vector of length 2 indicating which predictive CoCA axes to plot.
- **display**: which sets of scores are drawn. See `scores.predcoca`.
- **type**: one of "points", "text", or "none". Determines how the site and species scores are displayed. If `type = "points"`, scores are plotted as points. If `type = "text"`, then the row names of the scores matrices are plotted. If `type = "none"`, then the scores are not plotted.
- **xlim, ylim**: limits for the x and y axes. If non supplied, suitable limits will be determined from the data.
- **xlab, ylab**: labels for the x and y axes. If non supplied suitable labels are formed from the result object.
- **main, sub**: the main and sub titles for the plot.
- **ann**: logical, if TRUE plots are annotated and not if FALSE, currently ignored.
- **axes**: a logical value indicating whether both axes should be drawn on the plot.
- **...**: other graphical parameters as in 'par' may also be passed as arguments.

Author(s)

Gavin L. Simpson.

References


See Also

coca, plot.default

Examples

```r
## predictive CoCA
data(beetles)
data(plants)

## log transform the beetle data
beetles <- log(beetles + 1)

## predictive CoCA using SIMPLS and formula interface
bp.pred <- coca(beetles ~ ., data = plants)

## draw the plot for the response scores
plot(bp.pred)
```
## plot of both
layout(matrix(1:2, ncol = 2))
plot(bp.pred, which = "response", main = "Beetles")
plot(bp.pred, which = "predictor", main = "Plants")
layout(1)

### plot.symcoca

**Plots for symmetric co-correspondence analysis**

**Description**

Produces plots of the response and predictor from the results of a symmetric co-correspondence analysis.

**Usage**

```r
## S3 method for class 'symcoca'
plot(x,
    which = "response",
    choices = 1:2,
    display = c("species", "sites"),
    scaling = FALSE,
    type,
    xlim = NULL,
    ylim = NULL,
    main = "",
    sub = "",
    ylab, xlab,
    ann = par("ann"),
    axes = TRUE,
    ...)
```

**Arguments**

- **x**
  an object of class "symcoca", the result of a call to `symcoca`.
- **which**
  character; should the response or predictor scores be plotted.
- **choices**
  a vector of length 2 indicating which predictive CoCA axes to plot.
- **display**
  which sets of scores are drawn. See `scores.symcoca`.
- **scaling**
  logical, whether scaling should be applied. See `scores.symcoca`.
- **type**
  one of "points", "text", or "none". Determines how the site and species scores are displayed. If type = "points", scores are plotted as points. If type = "text", then the row names of the scores matrices are plotted. If type = "none", then the scores are not plotted.
- **xlim, ylim**
  limits for the x and y axes. If non supplied, suitable limits will be determined from the data.
Add points to a Co-CA plot

Description

Draws points on the current graphic device based on supplied `coca` model object.
Usage

```r
## S3 method for class 'coca'
points(x, display = c("sites", "species"),
       which = c("response", "predictor"), choices = c(1, 2),
       scaling = FALSE, select, ...)
```

Arguments

- **x**: an object inheriting from class `coca`.
- **display**: character; one of the stated choices. Indicates which scores to use to draw points.
- **which**: character; one of the stated choices. Indicates which of the response or predictor data sets is used to select scores from.
- **choices**: The Co-CA axes to draw points for.
- **scaling**: logical; should species scores in a symmetric Co-CA be rescaled?
- **select**: Items to be displayed. This can either be a logical vector which is TRUE for displayed items or a vector of indices of displayed items.
- **...**: Arguments passed to other methods

Details

The visual appearance of the plotted points can be controlled by supplying appropriate graphical parameters via the `...` argument. See `par` for details.

Value

Returns the plotted x and y coordinates as a matrix.

Author(s)

Gavin L. Simpson

See Also

- `plot` methods; `plot.predcoca` and `plot.symcoca`.

Examples

```r
## symmetric CoCA
data(beetles)
data(plants)

## log transform the bettle data
beetles <- log(beetles + 1)

## fit the model
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")

## draw a plot of the response scores
```
```r
plot(bp.sym, type = "none")
points(bp.sym, display = "sites", col = "blue", pch = 16)
points(bp.sym, display = "species", col = "red", pch = 3, cex = 0.8)
```

---

**rescale**

*Rescales CoCA species scores*

**Description**

Rescales CoCA species scores to the quarter root of the eigenvalues.

**Usage**

```r
rescale(object, ...)
```

## Default S3 method:

```r
rescale(object, ...)
```

## S3 method for class 'symcoca'

```r
rescale(object, choices = NULL,
        display = c("species", "sites"), ...)
```

**Arguments**

- `object` an R object. Currently only objects of class "symcoca" are supported.
- `choices` numeric; which axes to rescale and return.
- `display` character; which type of scores to rescale.
- `...` other arguments to be passed to rescale methods. Currently not used.

**Details**

Currently only implemented for objects of class "symcoca".

**Value**

Returns a list with one or both of the following components:

- `species` rescaled species scores for the response
- `sites` rescaled species scores for the predator

**Author(s)**


**See Also**

`symcoca`
Examples

```r
data(bryophyte)
data(vascular)

bryo.sym <- coca(bryophyte ~ ., data = vascular, method = "symmetric")
rescale(bryo.sym, axes = 1:2)
```

### resid.symcoca

**Extract Model Residuals**

#### Description

Extracts the residuals of the fitted model of a symmetric CoCA to the response and the predictor.

#### Usage

```r
## S3 method for class 'symcoca'
resid(object, ...)
```

#### Arguments

- `object` an object of class "symcoca".
- `...` arguments to be passed to other methods.

#### Value

A list containing the residuals for the response and the predictor with the following components:

- `Y` residuals of the fit to the response.
- `X` residuals of the fit to the predictor.

#### Author(s)

Gavin L. Simpson

#### See Also

- `symcoca`

#### Examples

```r
data(bryophyte)
data(vascular)
bryo.sym <- coca(bryophyte ~ ., data = vascular, method = "symmetric")
eps <- resid(bryo.sym)
```
scaleChi

Standardised chi-square residuals

Description

Scales a matrix, Y, to is standardised chi-square residuals \((o - e)/\sqrt{e}\) (given \(K_n\) and \(R_0\) metrics derived from an external matrix \(Y_0\)) so that further analysis can be unweighted.

Usage

scaleChi(Y, Kn, R0, eps = 1e-06)

Arguments

- **Y**: a matrix for which standardised chi-square residuals are to be calculated.
- **Kn**: the column sums (K) of Y divided by \(\text{sum}(K)\).
- **R0**: row weights.
- **eps**: a tolerance.

Value

- **Yr**: the matrix of standardised chi-squared residuals of Y.

Note

This function is not intended for casual use by users.

Author(s)


References

**Description**

Function to access either species or site scores for specified axes in co-correspondence analysis ordination methods.

**Usage**

```r
## S3 method for class 'predcoca'
scores(x, choices = c(1, 2),
       display = c("sites","species"), ...)

## S3 method for class 'symcoca'
scores(x, choices = c(1, 2),
       display = c("sites","species"), scaling = FALSE, ...)
```

**Arguments**

- `x` an ordination result
- `display` partial match to access scores for "sites" "species", "loadings" or "xmatrix". The latter two are only available for `symcoca`.
- `choices` numeric; the ordination axes to return.
- `scaling` logical; whether scores should be rescaled by the quarter root of the eigenvalues using `rescale.symcoca`.
- `...` arguments to be passed to other methods.

**Details**

Implements a `scores` method for symmetric co-correspondence analysis ordination results.

**Value**

A list with one or more components containing matrices of the requested scores:

- `species` A list with two components, `Y` and `X`, containing the species scores for the response matrix `Y` and the predictor matrix `X` respectively.
- `sites` A list with two components, `Y` and `X`, containing the site scores for the response matrix `Y` and the predictor matrix `X` respectively.
- `loadings` A list with two components, `Y` and `X` containing the loadings for the response and predictor matrix. For `symcoca` only.
- `xmatrix` The `X` matrix. For `symcoca` only.
**Author(s)**


**References**


**See Also**

`scores`, for further details on the method.

**Examples**

```r
## load some data
data(beetles)
data(plants)

## log transform the bettle data
beetles <- log(beetles + 1)

## fit the model, a symmetric CoCA
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")

## extract the scores
scr <- scores(bp.sym)

## predictive CoCA using SIMPLS and formula interface
bp.pred <- coca(beetles ~ ., data = plants)
scr2 <- scores(bp.pred)
```

**Description**

Produces a screeplot of the eigenvalues of a symmetric co-correspondence analysis.

**Usage**

```r
## S3 method for class 'symcoca'
screenplot(x, type = "b",
            xlab = NULL, ylab = NULL,
            ...)```
Arguments

x  an object of class "symcoca", the result of a call to symcoca.
type  the type of points to draw; see points.
xlab, ylab  labels for the x and y axes. If none supplied suitable labels are formed from the result object.
...  other graphical parameters as in 'par' may also be passed as arguments.

Author(s)

Gavin L. Simpson.

See Also

coca, screeplot, plot.default

Examples

## symmetric CoCA
data(beetles)
data(plants)

## log transform the bettle data
beetles <- log(beetles + 1)

## fit the model
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")

## draw a screeplot of the eignevalues
screeplot(bp.sym)

simples

Modified version of Sijmen de Jong's SIMPLS

Description

Fits a PLSR model with the SIMPLS algorithm, modified to allow a weighted analysis.

Usage

simples(X, Y, ncomp, stripped = FALSE, ...)
Arguments

- **X**: a matrix of observations. NAs and Infs are not allowed.
- **Y**: a vector or matrix of responses. NAs and Infs are not allowed.
- **ncomp**: the number of components to be used in the modelling.
- **stripped**: logical. If TRUE the calculations are stripped as much as possible for speed; this is meant for use with cross-validation or simulations when only the coefficients are needed. Defaults to FALSE.
- ... other arguments. Currently ignored.

Details

This function is a modified version of `simpls.fit` from package `pls`. Four modifications have been made:

1. The input matrices X and Y are not centered,
2. The scores (tt in the code) are not centered,
3. Added code to calculate the total variance in the Y matrix, `Ytotvar`, and the variance in Y accounted for by each PLS axis, `Yvar` (See Value below), and
4. Additional components are returned if argument `stripped` is TRUE.

This function should not be called directly, but through the generic function `coca`.

SIMPLS is much faster than the NIPALS algorithm, especially when the number of X variables increases, but gives slightly different results in the case of multivariate Y. SIMPLS truly maximises the covariance criterion. According to de Jong, the standard PLS2 algorithms lie closer to ordinary least-squares regression where a precise fit is sought; SIMPLS lies closer to PCR with stable predictions.

Value

A list containing the following components is returned:

- **coefficients**: an array of regression coefficients for 1, ..., ncomp components. The dimensions of coefficients are c(nvar, npred, ncomp) with nvar the number of X variables and npred the number of variables to be predicted in Y.
- **scores**: a matrix of scores.
- **loadings**: a matrix of loadings.
- **Yscores**: a matrix of Y-scores.
- **Yloadings**: a matrix of Y-loadings.
- **projection**: the projection matrix used to convert X to scores.
- **Xmeans**: a vector of means of the X variables.
- **Ymeans**: a vector of means of the Y variables.
- **fitted.values**: an array of fitted values. The dimensions of fitted.values are c(nobj, npred, ncomp) with nobj the number samples and npred the number of Y variables.
- **residuals**: an array of regression residuals. It has the same dimensions as fitted.values.
Xvar a vector with the amount of X-variance explained by each number of components.

Yvar a vector with the amount of Y-variance explained by each number of components.

Xtotvar Total variance in X.

Ytotvar Total variance in Y.

If stripped is TRUE, only the components coefficients, Xmeans and Ymeans, Xvar and Yvar, and Xtotvar and Ytotvar are returned.

Author(s)

Based on simpls.fit by Ron Wehrens and Bjorn-Helge Mevik, with simple modifications by Gavin L. Simpson.

References


See Also

coca

Description

summary methods for classes "predcoca" and "symcoca". These provide a summary of the main results of a Co-Correspondence Analysis model.

Usage

## S3 method for class 'predcoca'
summary(object, axes = NULL, ...)

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

Arguments

object an object of class "predcoca" or "symcoca". Generally the result of a call to coca.

axes numeric; how many axes to summarise? The default is to display 6 axes or all available axes, whichever is the smaller.

... arguments to be passed to other methods.
Value

A list with the some of the following components:

- **cocaScores**: The site and/or species scores for the axes requested.
- **call**: The call used to fit the model.
- **lambda**: The eigenvalues for the axes requested. Not for `predcoca.simpls`.
- **namY, namX**: The names of the response and predictor either supplied by the user or derived from the original call.
- **loadings**: a list with two components `loadings1` and `loadings2`, which refer to the response and the predictor matrices respectively. (Only for predictive CoCA models.)
- **varianceExp**: a list with components `Yblock` and `Xblock` containing the amount of variance explained on each CoCA axis in the response and the predictor respectively. (Only for predictive CoCA models.)
- **totalVar**: a list with components `Yblock` and `Xblock` containing the total variance in the response and the predictor data sets respectively
- **inertia**: a list with components `total` and `residual` containing the total and residual inertia (variance) in the response and the predictor matrices of a symmetric CoCA model. (Only for symmetric CoCA models.)
- **scaling**: the scaling used/requested. (Only for symmetric CoCA models.)

Author(s)

Gavin L. Simpson

See Also

The model fitting function `coca`

Examples

```r
## symmetric CoCA
data(beetles)
data(plants)

## log transform the bettle data
beetles <- log(beetles + 1)

## fit the model
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")
summary(bp.sym)

## Predictive CoCA
bp.pred <- coca(beetles ~ ., data = plants)
summary(bp.pred, axes = 1:2)
```
weights.symcoca  Extract CoCA model weights

Description
Extractor function to identify and select appropriate analysis weights from Co-correspondence Analysis models.

Usage

## S3 method for class 'symcoca'
weights(object, ...)

## S3 method for class 'predcoca'
weights(object, ...)

Arguments

object  an object of class "symcoca" or "predcoca".
...
arguments passed to other methods. Not used.

Value
A numeric vector of common site weights is currently returned. These correspond to $R_0$ in ter Braak and Schaffers (2004).

Author(s)
Gavin L. Simpson

References

See Also
data(coca) uses these weights to estimate weighted correlations in ordination space. See *coca* for details on creating CoCA models.

Examples

## symmetric CoCA
data(beetles)
data(plants)

## log transform the bettle data
beetles <- log(beetles + 1)

## fit the model
bp.sym <- coca(beetles ~ ., data = plants, method = "symmetric")

## weights == R[0]
weights(bp.sym)
Index

* datasets
  beetles, 3
  bryophyte, 6

* hplot
  biplot-methods, 3
  plot.predcoca, 25
  plot.symcoca, 27
  screeplot.symcoca, 34

* htest
  envfit.coca, 16

* methods
  biplot-methods, 3
  loadings, 20
  plot.symcoca, 27
  points.coca, 28
  scores-methods, 33
  screeplot.symcoca, 34
  weights.symcoca, 39

* models
  coca, 7
  coinertia, 10
  coinertiaI, 12
  fitCoinertia, 17

* multivariate
  coca, 7
  coinertia, 10
  coinertiaI, 12
  crossval, 14
  envfit.coca, 16
  fitCoinertia, 17
  fitted.symcoca, 19
  mcChi, 22
  permutest.coca, 23
  rescale, 30
  resid.symcoca, 31
  scaleChi, 32
  scores-methods, 33
  simpls, 35
  summary-methods, 37

* ordination
  coca, 7

* package
  cocorresp-package, 2

* regression
  coca, 7
  simpls, 35

* univar
  corAxis, 13

* utilities
  loadings, 20
  weights.symcoca, 39

  beetles, 3
  biplot-methods, 3
  biplot.predcoca (biplot-methods), 3
  biplot.symcoca (biplot-methods), 3
  bryophyte, 6

  coca, 2, 5, 7, 15–17, 20, 21, 24, 26, 28, 29, 35–39
  cocorresp (cocorresp-package), 2
  cocorresp-package, 2
  coinertia, 10, 13
  coinertiaI, 11, 12, 19
  cor, 13
  corAxis, 13
  crossval, 9, 14, 23, 24
  envfit, 17
  envfit.coca, 16, 39

  factorfit, 16
  fitCoinertia, 17
  fitted.symcoca, 19

  loadings, 20, 21
  loadings.default (loadings), 20

  mcChi, 22
ordiParseFormula, 9
par, 29
permutest.coca, 9, 12, 23
plants (beetles), 3
plot, 29
plot.default, 5, 26, 28, 35
plot.symcoca, 27, 29
points, 35
points.coca, 28
predcoca.simpls, 38
print.crossval (crossval), 14
print.fitCoinertia (fitCoinertia), 17
print.fitted.symcoca (fitted.symcoca), 19
print.permutest.coca (permutest.coca), 23
print.predcoca (coca), 7
print.summary.crossval (crossval), 14
print.summary.fitCoinertia (fitCoinertia), 17
print.summary.permutest.coca (permutest.coca), 23
print.summary.predcoca (summary-methods), 37
print.summary.symcoca (summary-methods), 37
print.symcoca (coca), 7
rescale, 30
rescale.symcoca, 33
resid.symcoca, 31
scaleChi, 32
scores, 33, 34
scores-methods, 33
scores.predcoca, 26
scores.predcoca (scores-methods), 33
scores.symcoca, 5, 16, 27
scores.symcoca (scores-methods), 33
screeplot, 35
screeplot.symcoca, 34
simpls, 35
simpls.fit, 8, 36, 37
summary-methods, 37
summary.crossval (crossval), 14
summary.fitCoinertia (fitCoinertia), 17
summary.permutest.coca (permutest.coca), 23
summary.predcoca, 9
summary.predcoca (summary-methods), 37
summary.symcoca, 9
summary.symcoca (summary-methods), 37
symcoca, 4, 11, 13, 18, 19, 27, 30, 31, 33, 35
vascular (bryophyte), 6
vectorfit, 16
verges (beetles), 3
weights.predcoca (weights.symcoca), 39
weights.symcoca, 39