Package 'rioja'

October 28, 2020

Type Package

Title Analysis of Quaternary Science Data

Version 0.9-26

Date 2020-10-26

Author Steve Juggins

Maintainer Steve Juggins <Stephen.Juggins@ncl.ac.uk>

Imports vegan, mgcv, grDevices

Suggests maptools, foreach

Description Constrained clustering, transfer functions, and other methods for analysing Quaternary science data.

License GPL-2

URL http://www.staff.ncl.ac.uk/stephen.juggins/, https://github.com/nsj3/rioja

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-10-28 16:20:03 UTC

R topics documented:

`rioja-package` *Analysis of Quaternary Science Data*

Description

`rioja`: An R package for the analysis of Quaternary science data. Contains functions for constrained clustering, transfer functions, and plotting stratigraphic data.

Details

The *rioja* package contains a number of tools for analysing and visualising (bio)stratigraphic data and for developing palaeoecological transfer functions from a dataset of modern species counts and environmental measurements. Resulting models can be cross-validated using the *crossval* function, which allows internal cross-validation using leave-one-out, leave-n-out, bootstrapping or h-block cross-validation.

Index of help topics:

`IK`	`Imbrie and Kipp foraminifera data`
`IKFA`	`Imbrie & Kipp Factor Analysis`
`LWR`	`Weighted averaging (LWR) regression and calibration`
`MAT`	`Palaeoenvironmental reconstruction using the Modern Analogue Technique (MAT)`
`MLRC`	`Palaeoenvironmental reconstruction using Maximum Likelihood Response Surfaces`
`MR`	`Multiple regression`
`Merge`	`Merges two or more data frames on the basis of common column names.`
`Ponds`	`Southeast England ponds and pools diatom and water chemistry dataset.`
`RLGH`	`Diatom stratigraphic data from the Round Loch of Glenhead, Galloway, Southwest Scotland`
`SWAP`	`SWAP surface sediment diatom data and lake-water pH.`

WA	Weighted averaging (WA) regression and calibration
WAPLS	Weighted averaging partial least squares (WAPLS) regression and calibration
aber	Abernethy Forest pollen data
chclust	Constrained hierarchical clustering
compare.datasets	Compare datasets for matching variables (species)
hulls	Graphic utilities.
inkspot	Two-way ordered bubble plot of a species by sites data table
interp.dataset	Interpolate a dataset
make.dummy	Utility functions.
performance	Palaeoecological transfer functions
randomPTF	Random transfer functions to calculate variable importance
rioja-package	Analysis of Quaternary Science Data
strat.plot	Plot a stratigraphic diagram

Author(s)

Steve Juggins

Maintainer: Steve Juggins <Stephen.Juggins@ncl.ac.uk>

---

aber *Abernethy Forest pollen data*

Description

Pollen stratigraphic data from Abernethy Forest, Scotland, spanning approximately 5500 - 12100 BP (from Birks & Mathews 1978). The data are a list with the following named components: spec Data are percentages of 36 dryland pollen taxa in 49 samples, (ages) core depths and ages for the 49 stratigraphic levels, and (names) codes and full names for the 36 taxa.

Usage

```
data(aber)
```

Source

Birks, HH & Mathews, RW (1978). Studies in the vegetational history of Scotland V. Late Devensian and early Flandrian macrofossil stratigraphy at Abernethy Forest, Invernessshire. *New Phytologist* **80**, 455-84.

Examples

```
data(aber)
strat.plot(aber$spec, scale.percent=TRUE, y.rev=TRUE)
```

chclust *Constrained hierarchical clustering*

Description

Constrained hierarchical clustering.

Usage

```
chclust(d, method = "coniss")

## S3 method for class 'chclust'
plot(x, labels = NULL, hang = 0.1, axes = TRUE,
      xvar=1:(length(x$height)+1), xlim=NULL, ylim=NULL,
      x.rev = FALSE, y.rev=FALSE, horiz=FALSE, ...)

bstick(n, ...)

## S3 method for class 'chclust'
bstick(n, ng=10, plot=TRUE, ...)
```

Arguments

d	a dissimilarity structure as produced, for example, by dist or vegdist.
method	the agglomeration method to be used. This should be (an unambiguous abbreviation of) either "coniss" or "conslink".
x, n	a constrained cluster object of class chclust produced by chclust.
xvar	numeric vector containing x-coordinates for the leaves of the dendrogram (see *details* below).
x.rev,y.rev	logical flags to reverse the x- or y-axis (and dendrogram labels). Defaults to FALSE.
horiz	logical indicating if the dendrogram should be drawn horizontally or not. Note that y-axis still refers to the dendrogram height even after rotating.
xlim, ylim	optional x- and y-limits of the plot, passed to the underlying plto function. The defaults for these show the full dendrogram.
labels, hang, axes	further arguments as in hclust.
ng	number of groups to display.
plot	logical to plot a broken stick model. Defaults to TRUE.
...	further graphical arguments. Use cex to change the text size of the x-axis labels, and cex.axis to change size of the y-axis values.

Details

chclust performs a constrained hierarchical clustering of a distance matrix, with clusters constrained by sample order. Returns an object of class chclust which can be plotted and interrogated. See Grimm (1987), Gordon & Birks (1972) and Birks & Gordon (1985) for discusssiom of the coniss and conslink algorithms. The resulting dendrogram can be plotted with plot. This is an extension of plot method for hclust that allows the dendrogram to be plotted horizontally or vertically (default). plot also accepts a numeric vector coordinates for x-axis positions of the leaves of the dendrogram. These could, for example, be the stratigraphic depths of core samples or geographic distances along a line transect.

bstick.chclust compares the dispersion of a hierarchical classification to that obtained from a broken stick model and displays the results graphically. See Bennett (1996) for details. bstick is a generic function and the default method is defined in package vegan. If package vegan is installed the function may be called using vegan::bstick, otherwise use bstick.chclust.

Value

Function chclust returns an object of class chclust, derived from hclust.

Author(s)

Steve Juggins

References

Bennett, K. (1996) Determination of the number of zones in a biostratigraphic sequence. *New Phytologist*, **132**, 155-170.

Birks, H.J.B. & Gordon, A.D. (1985) *Numerical Methods in Quaternary Pollen Analysis* Academic Press, London.

Gordon, A.D. & Birks, H.J.B. (1972) Numerical methods in Quaternary palaeoecology I. Zonation of pollen diagrams. *New Phytologist*, **71**, 961-979.

Grimm, E.C. (1987) CONISS: A FORTRAN 77 program for stratigraphically constrained cluster analysis by the method of incremental sum of squares. *Computers & Geosciences*, **13**, 13-35.

See Also

hclust, cutree, dendrogram.

Examples

```
data(RLGH)
diss <- dist(sqrt(RLGH$spec/100))
clust <- chclust(diss)
bstick(clust, 10)
# Basic diagram
plot(clust, hang=-1)
# Rotated through 90 degrees
plot(clust, hang=-1, horiz=TRUE)
# Rotated and observations plotted according to sample depth.
plot(clust, xvar=RLGH$depths$Depth, hang=-1, horiz=TRUE, x.rev=TRUE)
```

```
# Conslink for comparison
clust <- chclust(diss, method = "conslink")
plot(clust, hang=-1)
```

`compare.datasets` *Compare datasets for matching variables (species)*

Description

Compare two datasets and summarise species occurrance and abundance of species recorded in dataset one across dataset two. Useful for examining the conformity between sediment core and training set species data.

Usage

```
compare.datasets(y1, y2, n.cut=c(5, 10, 20, 50),
      max.cut=c(2, 5, 10, 20, 50))
```

Arguments

`y1, y2`	two data frames or matrices, usually of biological species abundance data, to compare.
`n.cut`	vector of abundances to be used for species occurrence calculations (see details).
`max.cut`	vector of occurences to be used for species maximum abundance calculations (see details).

Details

Function `compare.datasets` compares two datasets. It summarise the species profile (number of occurences etc.) and sample profile (number of species in each sample etc.) of dataset 1. For those species recorded in dataset 1 it also provides summaries of their occurence and abundance in dataset 2. It is useful diagnostic for checking the conformity between core and training set data, specifically for identifying core taxa absent from the training set, and core samples with portions of their assemblage missing from the training set.

`plot.compare.datasets` provides a simple visualisation of the comparisons. It produces a matrix of plots, one for each sample in dataset 1, showing the abundance of each taxon in dataset 1 (x-axis) against the N2 value of that taxon in dataset 2 (y-axis, with symbols scaled according to abundance in dataset 2. The plots shouls aid identification of samples with high abundance of taxa that are rare (low N2) or have low abundance in the training set. Taxa thar are absent from the training set are indicated with a red "+".

Value

Function `compare.datasets` returns a list with two names elements:

`vars` data frame listing for each variable in the first dataset: N.occur = number of occurences in dataset 1, N2, Hill's N2 for species in dataset 1, Max = maximum value in dataset 1, N.2 = number of occurences in dataset 2, N2.2 = Hill's N2 for species in dataset 2, Max.2 = maximum value in dataset 2, N.005, number of occurences where the species is greater than 5 etc.

`objs` data frame listing for each observation in the first dataset: N.taxa = number of species greater than zero abundance, N2, Hill's N2 for samples, Max = maximum value, total = sample total, M.002 = number of taxa with a maximum abundance greater than 2 2 etc., N2.005 = number of taxa in dataset 1 with more than 5 occurences in 2 dataset 2 etc., Sum.N2.005 = sample total including only those taxa with at least 5 occurrences in dataset 2 etc., M2.005 = number of taxa in dataset 1 with maximum abundance greater than 2 in dataset 2 etc., and Sum. M2.005 = sample total including only those taxa with a maximum abundance greater than 2 in dataset 2 etc.

Author(s)

Steve Juggins

Examples

```
# compare diatom data from core from Round Loch of Glenhead
# with SWAP surface sample dataset
data(RLGH)
data(SWAP)
result <- compare.datasets(RLGH$spec, SWAP$spec)
result
```

---

gutils *Graphic utilities.*

Description

Functions to perform simple graphics or enhance existing plots.

Usage

```
hulls(x, y, gr, col.gr=NULL)

figCnvt(fig1, fig2)
```

Arguments

x, y	vectors of x, y coordinates.
gr	factor to grop observations.
col.gr	a single colour or a vector of colours of length nG, where nG is the number of groups.
fig1, fig2	original fig dimensions (fig1) and new fig2 dimensions (fig2). See details.

Details

Function hulls is a wrapper for chull to add convex hulls to a scatterplot, optionally specifying a different colour for each hull.

Function figCnvt projects a set of fig dimensions fig2 with respect to an original set fig1. Useful for laying out plots where the ploting region has already been partitioned using fig.

Value

Function figCnvt returns a vector of 4 values specifying the new new figure dimensions.

Author(s)

Steve Juggins

Examples

```
data(iris)
with(iris, plot(Sepal.Width, Sepal.Length, col=as.integer(Species)))
with(iris, hulls(Sepal.Width, Sepal.Length, gr=(Species)))
```

IK *Imbrie and Kipp foraminifera data*

Description

Core-top foraminifera data from the Atlantic and Indian Oceans and core V12.122 from the Caribbean published by Imbrie and Kipp (1971). The data are a list with the following named components: spec relative abundances (percentages) of 22 foraminifera taxa in 61 core-top samples, (env) sea surface temperature and salinity measurements for the core-top samples, and (core) relative abundances of 28 foraminifer taxa in 110 samples from core V12.122.

Usage

```
data(IK)
```

References

Imbrie, J. & Kipp, N.G. (1971). A new micropaleontological method for quantitative paleoclimatology: application to a Late Pleistocene Caribbean core. In *The Late Cenozoic Glacial Ages* (ed K.K. Turekian), pp. 77-181. Yale University Press, New Haven.

Examples

```
data(IK)
names(IK$spec)
pairs(IK$env)
```

IKFA *Imbrie & Kipp Factor Analysis*

Description

Functions for reconstructing (predicting) environmental values from biological assemblages using Imbrie & Kipp Factor Analysis (IKFA), as used in palaeoceanography.

Usage

```
IKFA(y, x, nFact = 5, IsPoly = FALSE, IsRot = TRUE,
      ccoef = 1:nFact, check.data=TRUE, lean=FALSE, ...)

IKFA.fit(y, x, nFact = 5, IsPoly = FALSE, IsRot = TRUE,
      ccoef = 1:nFact, lean=FALSE)

## S3 method for class 'IKFA'
 predict(object, newdata=NULL, sse=FALSE, nboot=100,
      match.data=TRUE, verbose=TRUE, ...)

communality(object, y)

## S3 method for class 'IKFA'
crossval(object, cv.method="loo", verbose=TRUE, ngroups=10,
      nboot=100, h.cutoff=0, h.dist=NULL, ...)

## S3 method for class 'IKFA'
performance(object, ...)

## S3 method for class 'IKFA'
rand.t.test(object, n.perm=999, ...)

## S3 method for class 'IKFA'
screeplot(x, rand.test=TRUE, ...)

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

## S3 method for class 'IKFA'
summary(object, full=FALSE, ...)

## S3 method for class 'IKFA'
```

```
plot(x, resid=FALSE, xval=FALSE, nFact=max(x$ccoef),
      xlab="", ylab="", ylim=NULL, xlim=NULL, add.ref=TRUE,
      add.smooth=FALSE, ...)

## S3 method for class 'IKFA'
residuals(object, cv=FALSE, ...)

## S3 method for class 'IKFA'
coef(object, ...)

## S3 method for class 'IKFA'
fitted(object, ...)
```

Arguments

y	a data frame or matrix of biological abundance data.
x, object	a vector of environmental values to be modelled or an object of class wa.
newdata	new biological data to be predicted.
nFact	number of factor to extract.
IsRot	logical to rotate factors.
ccoef	vector of factor numbers to include in the predictions.
IsPoly	logical to include quadratic of the factors as predictors in the regression.
check.data	logical to perform simple checks on the input data.
match.data	logical indicate the function will match two species datasets by their column names. You should only set this to FALSE if you are sure the column names match exactly.
lean	logical to exclude some output from the resulting models (used when cross-validating to speed calculations).
full	logical to show head and tail of output in summaries.
resid	logical to plot residuals instead of fitted values.
xval	logical to plot cross-validation estimates.
xlab, ylab, xlim, ylim	additional graphical arguments to plot.wa.
add.ref	add 1:1 line on plot.
add.smooth	add loess smooth to plot.
cv.method	cross-validation method, either "loo", "lgo", "bootstrap" or "h-block".
verbose	logical to show feedback during cross-validation.
nboot	number of bootstrap samples.
ngroups	number of groups in leave-group-out cross-validation, or a vector contain leave-out group menbership.
h.cutoff	cutoff for h-block cross-validation. Only training samples greater than h.cutoff from each test sample will be used.

h.dist	distance matrix for use in h-block cross-validation. Usually a matrix of geographical distances between samples.
sse	logical indicating that sample specific errors should be calculated.
rand.test	logical to perform a randomisation t-test to test significance of cross validated factors.
n.perm	number of permutations for randomisation t-test.
cv	logical to indicate model or cross-validation residuals.
...	additional arguments.

Details

Function IKFA performs Imbrie and Kipp Factor Analysis, a form of Principal Components Regrssion (Imbrie & Kipp 1971).

Function predict predicts values of the environemntal variable for newdata or returns the fitted (predicted) values from the original modern dataset if newdata is NULL. Variables are matched between training and newdata by column name (if match.data is TRUE). Use compare.datasets to assess conformity of two species datasets and identify possible no-analogue samples.

IKFA has methods fitted and rediduals that return the fitted values (estimates) and residuals for the training set, performance, which returns summary performance statistics (see below), coef which returns the species coefficients, and print and summary to summarise the output. IKFA also has a plot method that produces scatter plots of predicted vs observed measurements for the training set.

Function rand.t.test performs a randomisation t-test to test the significance of the cross-validated components after van der Voet (1994).

Function screeplot displays the RMSE of prediction for the training set as a function of the number of factors and is useful for estimating the optimal number for use in prediction. By default screeplot will also carry out a randomisation t-test and add a line to scree plot indicating percentage change in RMSE with each component annotate with the p-value from the randomisation test.

Value

Function IKFA returns an object of class IKFA with the following named elements:

coefficients	species coefficients (the updated "optima").
fitted.values	fitted values for the training set.
call	original function call.
x	environmental variable used in the model.
standx, meanT sdx	additional information returned for a PLSif model.

Function crossval also returns an object of class IKFA and adds the following named elements:

predicted	predicted values of each training set sample under cross-validation.
residuals.cv	prediction residuals.

If function predict is called with newdata=NULL it returns the fitted values of the original model, otherwise it returns a list with the following named elements:

fit	predicted values for newdata.

If sample specific errors were requested the list will also include:

fit.boot	mean of the bootstrap estimates of newdata.
v1	standard error of the bootstrap estimates for each new sample.
v2	root mean squared error for the training set samples, across all bootstram samples.
SEP	standard error of prediction, calculated as the square root of v1^2 + v2^2.

Function performance returns a matrix of performance statistics for the IKFA model. See performance, for a description of the summary.

Function rand.t.test returns a matrix of performance statistics together with columns indicating the p-value and percentage change in RMSE with each higher component (see van der Veot (1994) for details).

Author(s)

Steve Juggins

References

Imbrie, J. & Kipp, N.G. (1971). A new micropaleontological method for quantitative paleoclimatology: application to a Late Pleistocene Caribbean core. In *The Late Cenozoic Glacial Ages* (ed K.K. Turekian), pp. 77-181. Yale University Press, New Haven.

van der Voet, H. (1994) Comparing the predictive accuracy of models uing a simple randomization test. *Chemometrics and Intelligent Laboratory Systems*, **25**, 313-323.

See Also

WA, MAT, performance, and compare.datasets for diagnostics.

Examples

```
data(IK)
spec <- IK$spec
SumSST <- IK$env$SumSST
core <- IK$core

fit <- IKFA(spec, SumSST)
fit
# cross-validate model
fit.cv <- crossval(fit, cv.method="lgo")
# How many components to use?
screeplot(fit.cv)

#predict the core
```

```
pred <- predict(fit, core, npls=2)

#plot predictions - depths are in rownames
depth <- as.numeric(rownames(core))
plot(depth, pred$fit[, 2], type="b")

# fit using only factors 1, 2, 4, & 5
# and using polynomial terms
# as Imbrie & Kipp (1971)
fit2 <- IKFA(spec, SumSST, ccoef=c(1, 2, 4, 5), IsPoly=TRUE)
fit2.cv <- crossval(fit2, cv.method="lgo")
screeplot(fit2.cv)

## Not run:
# predictions with sample specific errors
# takes approximately 1 minute to run
pred <- predict(fit, core, sse=TRUE, nboot=1000)
pred

## End(Not run)
```

`inkspot` *Two-way ordered bubble plot of a species by sites data table*

Description

Plots a two-way ordered bubble plot of a species by sites data table, where species are rows and sites are columns. The sites can be ordered and the functions will sort species to cluster abundances on the diagonal.

Usage

```
inkspot(data, gradient=1:nrow(data), use.rank=FALSE,
      reorder.species = TRUE, x.axis=c("sites", "gradient",
      "none"), x.axis.top=FALSE, site.names=NULL, spec.names=NULL,
      pch=21, cex.max=3, col="black", bg="darkgrey",
      x.grid=FALSE, y.grid=FALSE, grid.col="grey", grid.lty="dotted",
      legend.values=c(2, 5, 10, 20, 50), ...)
```

Arguments

`data`	data frame to be plotted.
`gradient`	a vector for ordering sites along the x-axis.
`use.rank`	logical to indicate that the rank rather than absolute values of gradient should be used to plot site positions. Defaults to `FALSE`.
`reorder.species`	should species be reordered to reflect pattern in site ordering? Defaults to `TRUE`.

x.axis	controls labelling of x.axis. sites uses site names, gradient plots an axis relecting values of the supplied gradient, and none omits labels and draws ticks at the site positions.
x.axis.top	logical to include an x-axis on the top of the figure labelled with values of the gradient.
site.names, spec.names	character vectors of site or species names to annotate the axes. Defaults to row and column names.
cex.max	maximum size of plotting symbol. Symbols are scaled so maximum species abundance has a symbol of this size.
pch, col, bg	plotting symbol and line / fill colour.
...	additional arguments to plot.
legend.values	if not null, places a legend in the top-left corner displaying the listed values.
x.grid, y.grid	draw grid at x and y ticks.
grid.col, grid.lty	grid colour and line type.

Details

Function inkspot plots a two-way table of species by sites as a bubble plot, with sites ordered along the x-axis, species on the y-axis, and abundance indiacted by scaled symbols ("bubbles"). It is a useful way to visualise species distribution along an envionmental, spatial or temporal gradient. If gradient is not given sites are plotting in the order they appear in the input data. Otherwise sites are plotting according to the values in gradient. If site labels overlap (multiple sites at similar values of gradient), labels can be suppressed x.axis= "none", or replaced with the gradient axis x.axis="gradient". A gradient axis can also be added to the top x.axis (x.axis.top=TRUE. Symbols are scaled so that the maximu abundance has a symbol size of cex.max. If sites are spaced unvenly along the gradient, or if many labels overlap, sites may be plotted evenly spaced using use.rank=TRUE. In this case the function will place top axis labels (if requested) at the appropriate positions along the gradient.

Value

Function inkspot returns a list with two named elements:

spec	index of the species order.
site	index of the site order.

Author(s)

Steve Juggins

See Also

vegemite in package vegan for a tabular alternative.

Examples

```
data(SWAP)
mx <- apply(SWAP$spec, 2, max)
spec <- SWAP$spec[, mx > 10]
#basic plot of data with legend
inkspot(spec, cex.axis=0.6)

#order sites by pH
pH <- SWAP$pH
inkspot(spec, pH, cex.axis=0.6)

# add a top axis
inkspot(spec, pH, x.axis.top=TRUE, cex.axis=0.6)

# order by pH but plot sites at regular intervals to avoid label overlap
inkspot(spec, pH, use.rank=TRUE, x.axis.top=TRUE, cex.axis=0.6)

# or add long taxon names
oldmar <- par("mar")
par(mar=c(3,12,2,1))
nms <- SWAP$names[mx > 10, 2]
inkspot(spec, pH, spec.names=as.character(nms), use.rank=TRUE,
x.axis.top=TRUE, cex.axis=0.6)
par(mar=oldmar)
```

---

interp.dataset *Interpolate a dataset*

Description

Given a data frame of variables measured along a temporal or spatial gradient, interpolate each variable to new values of the gradient. Useful for interpolating sediment core data to the depths ot ages of another sequences, or to evenly spaced intervals.

Usage

```
interp.dataset(y, x, xout, method=c("linear","loess","sspline"),
     rep.negt=TRUE, span=0.25, df=min(20, nrow(y)*.7), ...)
```

Arguments

y	data frame to be interpolated.
x	numeric vector giving ages, depths (ie. x-values(for data frame to be interpolated.
xout	numeric vector of values to interpolate to.
method	interpolation method, should be an unambiguous abbreviation of either linear, loess, sspline or aspline. See details.

rep.negt	logical to indicate whether or not to replace negative values with zero in the interpolated data.
span	span for loess, default=0.25.
df	degress of freedome for smoothing spline, default is the lower of 20 or 0.7 * number of samples.
...	additional arguments to approx, loess and smooth.spline.

Details

Function interp.dataset interpolates the columns of data frame with rows measured at intervals given by x, to new intervals given by xout. This function is useful to interpolation one set of sediment core data to the depth or ages of another, or to a regular set of intervals. Interpolation can be done using linear interpolation between data points in the original series (default) using function 'approx' in package 'stats', using loess locally weighted regression, or by smooth.spline. The latter two methods will also smooth the data and additional arguments may be passed to these functions to control the amount of smoothing.

Value

Function interp.datasets returns a data frame of the input data interpolated to the values given in xout. Values of xout outside the range of the original data are replaced by NA.

Author(s)

Steve Juggins

See Also

loess, and smooth.spline for details of interpolation methods.

Examples

```
data(RLGH)
spec <- RLGH$spec
depth <- RLGH$depths$Depth

# interpolate new dataset to every 0.5 cm
# using default method (linear)
x.new <- seq(0, 20, by=0.5)
sp.interp <- interp.dataset(y=spec, x=depth, xout=x.new)
## Not run:
# examine the results and compare to original data
strat.plot.simple(spec, depth, sp.interp, x.new)

## End(Not run)
```

LWR *Weighted averaging (LWR) regression and calibration*

Description

Functions for reconstructing (predicting) environmental values from biological assemblages using weighted averaging (LWR) regression and calibration.

Usage

```
LWR(y, x, FUN=WA, dist.method="sq.chord", k=30, lean=TRUE,
       fit.model=TRUE, check.data=TRUE, verbose=TRUE, ...)

## S3 method for class 'LWR'
 predict(object, newdata=NULL, k = object$k, sse=FALSE,
      nboot=100, match.data=TRUE, verbose=TRUE, lean=TRUE, ...)

## S3 method for class 'LWR'
crossval(object, k=object$k, cv.method="lgo", verbose=TRUE,
      ngroups=10, nboot=100, h.cutoff=0, h.dist=NULL, ...)

## S3 method for class 'LWR'
performance(object, ...)

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

## S3 method for class 'LWR'
summary(object, full=FALSE, ...)

## S3 method for class 'LWR'
residuals(object, cv=FALSE, ...)

## S3 method for class 'LWR'
fitted(object, ...)
```

Arguments

y	a data frame or matrix of biological abundance data.
x, object	a vector of environmental values to be modelled or an object of class LWR.
dist.method	distance measure used to derfine closest analogues.
k	number of close analogues to use in calibration function.
FUN	calibration function (e.g. WA, WAPLS etc).
newdata	new biological data to be predicted.
fit.model	TRUE fits model to training set. FALSE omist this step and builds a LWR object than can be used for prediction.

check.data	logical to perform simple checks on the input data.
full	logical to show head and tail of output in summaries.
match.data	logical indicate the function will match two species datasets by their column names. You should only set this to FALSE if you are sure the column names match exactly.
lean	logical to exclude some output from the resulting models (used when cross-validating to speed calculations).
cv.method	cross-validation method, either "lgo" or "bootstrap".
verbose	logical to show feedback during cross-validaton.
nboot	number of bootstrap samples.
ngroups	number of groups in leave-group-out cross-validation.
h.cutoff	cutoff for h-block cross-validation. Only training samples greater than h.cutoff from each test sample will be used.
h.dist	distance matrix for use in h-block cross-validation. Usually a matrix of geographical distances between samples.
sse	logical indicating that sample specific errors should be calculated.
cv	logical to indicate model or cross-validation residuals.
...	additional arguments.

Details

Function LWR performs ... To do.

Value

Function LWR returns an object of class LWR with the following named elements:

Author(s)

Steve Juggins

See Also

WAPLS, MAT, and compare.datasets for diagnostics.

MAT *Palaeoenvironmental reconstruction using the Modern Analogue Technique (MAT)*

Description

Functions for reconstructing (predicting) environmental values from biological assemblages using the Modern Analogue Technique (MAT), also know as k nearest neighbours (k-NN).

Usage

```
MAT(y, x, dist.method="sq.chord", k=5, lean=TRUE)

## S3 method for class 'MAT'
predict(object, newdata=NULL, k=object$k, sse=FALSE,
        nboot=100, match.data=TRUE, verbose=TRUE, lean=TRUE,
        ...)

## S3 method for class 'MAT'
performance(object, ...)

## S3 method for class 'MAT'
crossval(object, k=object$k, cv.method="lgo",
        verbose=TRUE, ngroups=10, nboot=100, h.cutoff=0, h.dist=NULL, ...)

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

## S3 method for class 'MAT'
summary(object, full=FALSE, ...)

## S3 method for class 'MAT'
plot(x, resid=FALSE, xval=FALSE, k=5, wMean=FALSE, xlab="",
      ylab="", ylim=NULL, xlim=NULL, add.ref=TRUE,
      add.smooth=FALSE, ...)

## S3 method for class 'MAT'
residuals(object, cv=FALSE, ...)

## S3 method for class 'MAT'
fitted(object, ...)

## S3 method for class 'MAT'
screeplot(x, ...)

paldist(y, dist.method="sq.chord")
```

```
paldist2(y1, y2, dist.method="sq.chord")
```

Arguments

y, y1, y2	data frame containing biological data.
newdata	data frame containing biological data to predict from.
x	a vector of environmental values to be modelled, matched to y.
dist.method	dissimilarity coefficient. See details for options.
match.data	logical indicate the function will match two species datasets by their column names. You should only set this to FALSE if you are sure the column names match exactly.
k	number of analogues to use.
lean	logical to remove items form the output.
object	an object of class MAT.
resid	logical to plot residuals instead of fitted values.
xval	logical to plot cross-validation estimates.
wMean	logical to plot weighted-mean estimates.
xlab, ylab, xlim, ylim	additional graphical arguments to plot.wa.
add.ref	add 1:1 line on plot.
add.smooth	add loess smooth to plot.
cv.method	cross-validation method, either "lgo", "bootstrap" or "h-block".
verbose	logical to show feedback during cross-validation.
nboot	number of bootstrap samples.
ngroups	number of groups in leave-group-out cross-validation, or a vector contain leave-out group menbership.
h.cutoff	cutoff for h-block cross-validation. Only training samples greater than h.cutoff from each test sample will be used.
h.dist	distance matrix for use in h-block cross-validation. Usually a matrix of geographical distances between samples.
sse	logical indicating that sample specific errors should be calculated.
full	logical to indicate a full or abbreviated summary.
cv	logical to indicate model or cross-validation residuals.
...	additional arguments.

Details

MAT performs an environmental reconstruction using the modern analogue technique. Function MAT takes a training dataset of biological data (species abundances) y and a single associated environmental variable x, and generates a model of closest analogues, or matches, for the modern data data using one of a number of dissimilarity coefficients. Options for the latter are: "euclidean", "sq.euclidean", "chord", "sq.chord", "chord.t", "sq.chord.t", "chi.squared", "sq.chi.squared", "bray".

"chord.t" are true chord distances, "chord" refers to the the variant of chord distance using in palaeoecology (e.g. Overpeck et al. 1985), which is actually Hellinger's distance (Legendre & Gallagher 2001). There are various help functions to plot and extract information from the results of a MAT transfer function. The function predict takes MAT object and uses it to predict environmental values for a new set of species data, or returns the fitted (predicted) values from the original modern dataset if newdata is NULL. Variables are matched between training and newdata by column name (if match.data is TRUE). Use compare.datasets to assess conformity of two species datasets and identify possible no-analogue samples.

MAT has methods fitted and rediduals that return the fitted values (estimates) and residuals for the training set, performance, which returns summary performance statistics (see below), and print and summary to summarise the output. MAT also has a plot method that produces scatter plots of predicted vs observed measurements for the training set.

Function screeplot displays the RMSE of prediction for the training set as a function of the number of analogues (k) and is useful for estimating the optimal value of k for use in prediction.

paldist and paldist1 are helper functions though they may be called directly. paldist takes a single data frame or matrix returns a distance matrix of the row-wise dissimilarities. paldist2 takes two data frames of matrices and returns a matrix of all row-wise dissimilarities between the two datasets.

Value

Function MAT returns an object of class MAT which contains the following items:

call	original function call to MAT.
fitted.vales	fitted (predicted) values for the training set, as the mean and weighted mean (weighed by dissimilarity) of the k closest analogues.
diagnostics	standard deviation of the k analogues and dissimilarity of the closest analogue.
dist.n	dissimilarities of the k closest analogues.
x.n	environmental values of the k closest analogues.
match.name	column names of the k closest analogues.
x	environmental variable used in the model.
dist.method	dissimilarity coefficient.
k	number of closest analogues to use.
y	original species data.
cv.summary	summary of the cross-validation (not yet implemented).
dist	dissimilarity matrix (returned if lean=FALSE).

If function predict is called with newdata=NULL it returns a matrix of fitted values from the original training set analysis. If newdata is not NULL it returns list with the following named elements:

fit	predictions for newdata.
diagnostics	standard deviations of the k closest analogues and distance of closest analogue.
dist.n	dissimilarities of the k closest analogues.
x.n	environmental values of the k closest analogues.

match.name	column names of the k closest analogues.
dist	dissimilarity matrix (returned if lean=FALSE).

If sample specific errors were requested the list will also include:

fit.boot	mean of the bootstrap estimates of newdata.
v1	standard error of the bootstrap estimates for each new sample.
v2	root mean squared error for the training set samples, across all bootstram samples.
SEP	standard error of prediction, calculated as the square root of v1^2 + v2^2.

Functions paldist and paldist2 return dissimilarity matrices. performance returns a matrix of performance statistics for the MAT model, with columns for RMSE, R2, mean and max bias for each number of analogues up to k. See performance for a description of the output.

Author(s)

Steve Juggins

References

Legendre, P. & Gallagher, E. (2001) Ecologically meaningful transformations for ordination of species. *Oecologia*, **129**, 271-280.

Overpeck, J.T., Webb, T., III, & Prentice, I.C. (1985) Quantitative interpretation of fossil pollen spectra: dissimilarity coefficients and the method of modern analogs. *Quaternary Research*, **23**, 87-108.

See Also

WAPLS, WA, performance, and compare.datasets for diagnostics.

Examples

```
# pH reconstruction of the RLGH, Scotland, using SWAP training set
# shows recent acidification history
data(SWAP)
data(RLGH)
fit <- MAT(SWAP$spec, SWAP$pH, k=20)  # generate results for k 1-20
#examine performance
performance(fit)
print(fit)
# How many analogues?
screeplot(fit)
# do the reconstruction
pred.mat <- predict(fit, RLGH$spec, k=10)
# plot the reconstruction
plot(RLGH$depths$Age, pred.mat$fit[, 1], type="b", ylab="pH", xlab="Age")

#compare to a weighted average model
fit <- WA(SWAP$spec, SWAP$pH)
```

```
pred.wa <- predict(fit, RLGH$spec)
points(RLGH$depths$Age, pred.wa$fit[, 1], col="red", type="b")
legend("topleft", c("MAT", "WA"), lty=1, col=c("black", "red"))
```

`Merge` *Merges two or more data frames on the basis of common column names.*

Description

Merges two or more data frames on the basis of common column names.

Usage

```
Merge(..., join="outer", fill=0, split=FALSE, verbose=TRUE)
```

Arguments

`...`	two or more data frames to merge.
`join`	type of join to perform. Should be an unambiguous abbreviation of either "outer", "inner", or "leftouter". An outer join produces a data frame that contains all the unique column names of the input data, ie, the union of all input column names. An inner join produces a data frame containing only column names that are common across the input data, ie. the intersection of the input column names. A left outer join produces a data frame containing all column names of the first data frame only: column names that occur in subsequent data frames are omitted.
`fill`	value to use to fill non-matched columns. Defaults to zero which is appropriate for species abundance data.
`split`	logical to return a single data frame (TRUE) or a named list containing separate (original) data frames with a common set of merged columns (FALSE). Defaults to TRUE (a single data frame).
`verbose`	logical to suppress warning messages.

Details

`Merge` is a utilty function for combining separate datasets of biological count data that have only a subset of taxa (column names) in common. The outer join is appropriate for merging prior to a joint ordination or for merging a training set and core data prior to environmental reconstruction using the modern analogue technique (MAT). A left outer join should be used to prepare data for an ordination of a training set and subsequent projection of a second onto the ordination axes. The function is capitalised to distinguish it from `merge` in the base R.

Value

If split is set to FALSE the function returns a single data frame with the number of rows equal to the combined rows of the input data and columns sorted alphabetically according to the join type. Otherwise returns a named list of the merged data frames.

Author(s)

Steve Juggins

See Also

merge.

Examples

```
data(RLGH)
data(SWAP)
# Merge RLGH core data with SWAP training set
# Extract species data from datasets
SWAPsp <- SWAP$spec
RLGHsp <- RLGH$spec
# full outer join for joint ordination of both datasets
comb <- Merge(SWAPsp, RLGHsp)

## Not run:
# superimpose core trajectory on ordination plot
library(vegan) # decorana
ord <- decorana(comb, iweigh=1)
par(mfrow=c(1,2))
plot(ord, display="sites")
sc <- scores(ord, display="sites")
sc <- sc[(nrow(SWAPsp)+1):nrow(comb), ]
lines(sc, col="red")
title("Joint DCA ordination of surface and core")

# Do the same but this time project core passively
# Note we cannot use data from the outer join since decorana
# will delete taxa only present in the core - the resulting
# ordination model will then not match the taxa in the core
comb2 <- Merge(SWAPsp, RLGHsp, join="leftouter", split=TRUE)
ord2 <- decorana(comb2$SWAPsp, iweigh=1)
sc2 <- predict(ord2, comb2$RLGHsp, type="sites")
plot(ord2, display="sites")
lines(sc2, col="red")
title("DCA with core added \"passively\"")

## End(Not run)
```

MLRC *Palaeoenvironmental reconstruction using Maximum Likelihood Response Surfaces*

Description

Functions for reconstructing (predicting) environmental values from biological assemblages using Maximum Likelihood response Surfaces.

Usage

```
MLRC(y, x, check.data=TRUE, lean=FALSE, n.cut=5, verbose=TRUE, ...)

MLRC.fit(y, x, n.cut=2, use.glm=FALSE, max.iter=50, lean=FALSE, verbose=FALSE, ...)

## S3 method for class 'MLRC'
 predict(object, newdata=NULL, sse=FALSE, nboot=100,
      match.data=TRUE, verbose=TRUE, ...)

## S3 method for class 'MLRC'
crossval(object, cv.method="loo", verbose=TRUE, ngroups=10,
      nboot=100, h.cutoff=0, h.dist=NULL, ...)

## S3 method for class 'MLRC'
performance(object, ...)

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

## S3 method for class 'MLRC'
summary(object, full=FALSE, ...)

## S3 method for class 'MLRC'
plot(x, resid=FALSE, xval=FALSE, xlab="", ylab="",
      ylim=NULL, xlim=NULL, add.ref=TRUE, add.smooth=FALSE, ...)

## S3 method for class 'MLRC'
residuals(object, cv=FALSE, ...)

## S3 method for class 'MLRC'
coef(object, ...)

## S3 method for class 'MLRC'
fitted(object, ...)
```

Arguments

y	a data frame or matrix of biological abundance data.
x, object	a vector of environmental values to be modelled or an object of class wa.
n.cut	cutoff value for number of occurrences. Species with fewer than n.cut occurrences will be excluded from the analysis.
use.glm	logical to use glm to fit responses rather than internal code. Defaults to FALSE.
newdata	new biological data to be predicted.
max.iter	maximum iterations of the logit regression algorithm.
check.data	logical to perform simple checks on the input data.

match.data	logical indicate the function will match two species datasets by their column names. You should only set this to FALSE if you are sure the column names match exactly.
lean	logical to exclude some output from the resulting models (used when cross-validating to speed calculations).
full	logical to show head and tail of output in summaries.
resid	logical to plot residuals instead of fitted values.
xval	logical to plot cross-validation estimates.
xlab, ylab, xlim, ylim	additional graphical arguments to plot.wa.
add.ref	add 1:1 line on plot.
add.smooth	add loess smooth to plot.
cv.method	cross-validation method, either "loo", "lgo", "bootstrap" or "h-block".
verbose	logical to show feedback during cross-validation.
nboot	number of bootstrap samples.
ngroups	number of groups in leave-group-out cross-validation, or a vector contain leave-out group menbership.
h.cutoff	cutoff for h-block cross-validation. Only training samples greater than h.cutoff from each test sample will be used.
h.dist	distance matrix for use in h-block cross-validation. Usually a matrix of geographical distances between samples.
sse	logical indicating that sample specific errors should be calculated.
cv	logical to indicate model or cross-validation residuals.
...	additional arguments.

Details

Function MLRC Maximim likelihood reconstruction using response curves.

Function predict predicts values of the environemntal variable for newdata or returns the fitted (predicted) values from the original modern dataset if newdata is NULL. Variables are matched between training and newdata by column name (if match.data is TRUE). Use compare.datasets to assess conformity of two species datasets and identify possible no-analogue samples.

MLRC has methods fitted and rediduals that return the fitted values (estimates) and residuals for the training set, performance, which returns summary performance statistics (see below), coef which returns the species coefficients, and print and summary to summarise the output. MLRC also has a plot method that produces scatter plots of predicted vs observed measurements for the training set.

Value

Function MLRC returns an object of class MLRC with the following named elements:

Function crossval also returns an object of class MLRC and adds the following named elements:

predicted	predicted values of each training set sample under cross-validation.

`residuals.cv` prediction residuals.

If function `predict` is called with `newdata=NULL` it returns the fitted values of the original model, otherwise it returns a list with the following named elements:

`fit` predicted values for `newdata`.

If sample specific errors were requested the list will also include:

`fit.boot` mean of the bootstrap estimates of newdata.

`v1` standard error of the bootstrap estimates for each new sample.

`v2` root mean squared error for the training set samples, across all bootstram samples.

`SEP` standard error of prediction, calculated as the square root of v1^2 + v2^2.

Function `performance` returns a matrix of performance statistics for the MLRC model. See `performance`, for a description of the summary.

Author(s)

Steve Juggins

References

Birks, H.J.B., Line, J.M., Juggins, S., Stevenson, A.C., & ter Braak, C.J.F. (1990) Diatoms and pH reconstruction. *Philosophical Transactions of the Royal Society of London*, **B, 327**, 263-278.

Juggins, S. (1992) Diatoms in the Thames Estuary, England: Ecology, Palaeoecology, and Salinity Transfer Function. *Bibliotheca Diatomologica*, **Band 25**, 216pp.

Oksanen, J., Laara, E., Huttunen, P., & Merilainen, J. (1990) Maximum likelihood prediction of lake acidity based on sedimented diatoms. *Journal of Vegetation Science*, **1**, 49-56.

ter Braak, C.J.F. & van Dam, H. (1989) Inferring pH from diatoms: a comparison of old and new calibration methods. *Hydrobiologia*, **178**, 209-223.

See Also

`WA`, `MAT`, `performance`, and `compare.datasets` for diagnostics.

Examples

```
data(IK)
spec <- IK$spec / 100
SumSST <- IK$env$SumSST
core <- IK$core / 100

fit <- MLRC(spec, SumSST)
fit

#predict the core
pred <- predict(fit, core)
```

```
#plot predictions - depths are in rownames
depth <- as.numeric(rownames(core))
plot(depth, pred$fit[, 1], type="b")

## Not run:
# this is slow!
# cross-validate model
fit.cv <- crossval(fit, cv.method="loo", verbose=5)

# predictions with sample specific errors
pred <- predict(fit, core, sse=TRUE, nboot=1000, verbose=5)

## End(Not run)
```

MR *Multiple regression*

Description

Functions for reconstructing (predicting) environmental values from biological assemblages using multiple regression.

Usage

```
MR(y, x, check.data=TRUE, lean=FALSE, ...)

MR.fit(y, x, lean=FALSE)

## S3 method for class 'MR'
 predict(object, newdata=NULL, sse=FALSE, nboot=100,
      match.data=TRUE, verbose=TRUE, ...)

## S3 method for class 'MR'
crossval(object, cv.method="loo", verbose=TRUE, ngroups=10,
      nboot=100, h.cutoff=0, h.dist=NULL, ...)

## S3 method for class 'MR'
performance(object, ...)

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

## S3 method for class 'MR'
summary(object, full=FALSE, ...)

## S3 method for class 'MR'
plot(x, resid=FALSE, xval=FALSE, xlab="",
      ylab="", ylim=NULL, xlim=NULL, add.ref=TRUE,
```

```
      add.smooth=FALSE, ...)

## S3 method for class 'MR'
residuals(object, cv=FALSE, ...)

## S3 method for class 'MR'
coef(object, ...)

## S3 method for class 'MR'
fitted(object, ...)
```

Arguments

y	a data frame or matrix of biological abundance data.
x, object	a vector of environmental values to be modelled or an object of class wa.
newdata	new biological data to be predicted.
check.data	logical to perform simple checks on the input data.
match.data	logical indicate the function will match two species datasets by their column names. You should only set this to FALSE if you are sure the column names match exactly.
lean	logical to exclude some output from the resulting models (used when cross-validating to speed calculations).
full	logical to show head and tail of output in summaries.
resid	logical to plot residuals instead of fitted values.
xval	logical to plot cross-validation estimates.
xlab, ylab, xlim, ylim	additional graphical arguments to plot.wa.
add.ref	add 1:1 line on plot.
add.smooth	add loess smooth to plot.
cv.method	cross-validation method, either "loo", "lgo", "bootstrap" or "h-block".
verbose	logical to show feedback during cross-validation.
nboot	number of bootstrap samples.
ngroups	number of groups in leave-group-out cross-validation, or a vector contain leave-out group menbership.
h.cutoff	cutoff for h-block cross-validation. Only training samples greater than h.cutoff from each test sample will be used.
h.dist	distance matrix for use in h-block cross-validation. Usually a matrix of geographical distances between samples.
sse	logical indicating that sample specific errors should be calculated.
cv	logical to indicate model or cross-validation residuals.
...	additional arguments.

Details

Function MR performs multiple regrssion. It is a wrapper to lm.

Function predict predicts values of the environmental variable for newdata or returns the fitted (predicted) values from the original modern dataset if newdata is NULL. Variables are matched between training and newdata by column name (if match.data is TRUE). Use compare.datasets to assess conformity of two species datasets and identify possible no-analogue samples.

MR has methods fitted and rediduals that return the fitted values (estimates) and residuals for the training set, performance, which returns summary performance statistics (see below), coef which returns the species coefficients, and print and summary to summarise the output. MR also has a plot method that produces scatter plots of predicted vs observed measurements for the training set.

Value

Function MR returns an object of class MR with the following named elements:

coefficients	species coefficients (the updated "optima").
fitted.values	fitted values for the training set.
call	original function call.
x	environmental variable used in the model.

Function crossval also returns an object of class MR and adds the following named elements:

predicted	predicted values of each training set sample under cross-validation.
residuals.cv	prediction residuals.

If function predict is called with newdata=NULL it returns the fitted values of the original model, otherwise it returns a list with the following named elements:

fit	predicted values for newdata.

If sample specific errors were requested the list will also include:

fit.boot	mean of the bootstrap estimates of newdata.
v1	standard error of the bootstrap estimates for each new sample.
v2	root mean squared error for the training set samples, across all bootstram samples.
SEP	standard error of prediction, calculated as the square root of v1^2 + v2^2.

Function performance returns a matrix of performance statistics for the MR model. See performance, for a description of the summary.

Author(s)

Steve Juggins

See Also

WA, MAT, performance, and compare.datasets for diagnostics.

Examples

```
data(IK)
spec <- IK$spec
SumSST <- IK$env$SumSST
core <- IK$core

# Generate a MR model using taxa with max abun > 20%

mx <- apply(spec, 2, max)
spec2 <- spec[, mx > 20]

fit <- MR(spec2, SumSST)
fit
# cross-validate model
fit.cv <- crossval(fit, cv.method="lgo")
fit.cv

#predict the core
pred <- predict(fit, core)

#plot predictions - depths are in rownames
depth <- as.numeric(rownames(core))
plot(depth, pred$fit[, 1], type="b")

## Not run:
# predictions with sample specific errors
# takes approximately 1 minute to run
pred <- predict(fit, core, sse=TRUE, nboot=1000)
pred

## End(Not run)
```

Ponds *Southeast England ponds and pools diatom and water chemistry dataset.*

Description

Diatom and associated water chemistry data for 30 small ponds & pools from SE England collected by, and described in Bennion (1994). Dataset is a list with the following named elements: (spec) diatom relative abundances for 48 selected common taxa, (env) lake names, UK GB grid references, lake depth (m) and mean lake-water chemistry. Units are ueq/l except pH, conductivity (uS/cm), alkalinity (meq/l), total phoshporus and chlorophyll-a (ug/l), and nitrate (mg/l). Column names in spec are short, 6-character alphanumeric codes for each diatom taxon. Ponds$names contains the full names for each taxon, in the correct order).

Usage

```
data(Ponds)
```

Source

Bennion, H. (1994) A diatom-phosphorus transfer function for shallow, eutrophic ponds in southeast England. *Hydrobiologia*, **275/276**, 391-410.

Examples

```
data(Ponds)
names(Ponds$spec)
hist(Ponds$env$TP)
```

PTF *Palaeoecological transfer functions*

Description

Functions for diagnosing and interpreting palaeoecological transfer functions.

Usage

```
## Default S3 method:
performance(object, ...)

## Default S3 method:
crossval(object, ...)
```

Arguments

`object`	a transfer function model from `wa`, `wapls` etc.
`...`	additional arguments.

Details

Package `rioja` implements a number of numerical methods for inferring the value of an environmental variable from a set of sepecies abundances, given a modern training set of species data and associated environmental values. In palaeoecology these are known as "transfer functions" or "inference models" and are used to hindcast or "reconstruct" past environmental conditions from sub-fossil species assemblages preserved in sediment cores. The techniques included are weighted averaging (`WA`), partial least squares (PLS) and weighted average partial least squared (`WAPLS`), Imbrie and Kipp Factor Analysis (`IKFA`) a form of principal components regression, Maximum Likelihood Response Curves (`MLRC`), and the Modern Analogue Technique (`MAT`, a form of k-NN non-parametric regression (see Juggins & Birks (2010) for a review).

The techniques are implemented in a consistent way and include functions for fitting a model to a training set of species and environmental data, with the function name named after the technique: that is, `WA` fits a weighted averaging model. Any model can be cross-validated using the `crossval` function, which allows internal cross-validation using leave-one-out, leave-n-out, bootstrapping or h-block cross-validation. There are a number of generic functions that can be used to summarise and diagnose the models: (`print`, `summary`, `performance` and `plot`. Some techniques have additional

diagnostic functions such as `screeplot` and `rand.t.test` to help estimate the approproate number of components (WAPLS), factors (IKFA) or number of analogues (IKFA).

Predictions for new species data can be made using `predict`, with an option to calculate sample-specific errors using bootstrapping, after the method described in Birks et al. (1990).

Value

Function `performance` returns a list with a named matrix `object` which contains the following columns:

`RMSE`	root mean squared error, defined as the square root of the average sqaured error between the observed and predicted values for the training set.
`R2`	squared correlation betweenn observed and predicted values.
`Avg.Bias`	mean bias (mean of the residuals between measured and predicted values).
`Max.Bias`	maximum bias, calculated by dividing the environmental gradient into a number of equal spaced segments (10 by default) and calculating the average bias for each segment. The maximum bias is maximum of these 10 values and quantifies the tendendy for the model to over- or under-estimate at particular part of the gradient (ter Braak & Juggins 1993).

If the transfer function object has been cross-validated, (ie. is the output of `crossval`, the list returned by `performance` also contains a matrix named crossval, which contains the above statistics calculated for the cross-validation predictions.

Function `crossval` returns an object of the original class and adds the following named elements:

`predicted`	predicted values of each training set sample under cross-validation.
`residuals.cv`	prediction residuals.

Function `rand.t.test` is a generic function that performs a randomisation t-test to test the significance of a cross-validated model, after van der Voet (1994). Methods exist for `WA`, `WAPLS` and `IKFA`.

Author(s)

Steve Juggins

References

Birks, H.J.B., Line, J.M., Juggins, S., Stevenson, A.C., & ter Braak, C.J.F. (1990) Diatoms and pH reconstruction. *Philosophical Transactions of the Royal Society of London*, **B, 327**, 263-278.

Juggins, S., & Birks, HJB. (2010) Environmental Reconstructions. In Birks et al. (eds) *Tracking Environmental Change using Lake Sediments: Data Handling and Statistical Techniques*., Kluwer Academic Publishers.

van der Voet, H. (1994) Comparing the predictive accuracy of models uing a simple randomization test. *Chemometrics and Intelligent Laboratory Systems*, **25**, 313-323.

randomPTF *Random transfer functions to calculate variable importance*

Description

Function for calculating the important of each taxon (predictor) in palaeoecological transfer functions

Usage

```
randomPTF(spec, env, fun, ncol = 1, nVar, nTF = 500, verbose = TRUE,
          do.parallel = FALSE, ...)

## S3 method for class 'randomPTF'
plot(x, use.pointLabel=TRUE, ...)

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

Arguments

spec	a data frame or matrix of biological abundance data.
env	a vector of environmental values to be modelled.
fun	a transfer function method. Additional arguments can be passed with . . .)
ncol	some transfer functions return more than one column of results, for example with different WAPLS components. col selects which column to use. See the relevant transfer function method help file.
nVar	number of variables (ie. species) to use in each randomisation (defaults to nsp/3).
nTF	number of random transfer functions to create (default=500).
verbose	logical show feedback during cross-validation.
do.parallel	logical to run in parallel on multi-core machines. If true a suitable parallel backend should be installed (see examples).
...	additional parameters to the transfer function call.
x	an object of class randomPTF.
use.pointLabel	logical to label points using function labelPoints in package maptools.

Details

Function randomPTF calculates taxon importance values using a method analogous to that used in random forests and described in Juggins et al. (2015).

The parallel version can give c. 3 times speed-up on a quad-core machine.

Value

Function randomPTF returns an object of class randomPTF with the following named elements:

VI	taxon importance values, ordered form high to low.
spec	original species data frame.
env	original vector of environmental values.

Author(s)

Steve Juggins

References

Juggins S, Simpson GL, Telford RJ. Taxon selection using statistical learning techniques to improve transfer function prediction. The Holocene 2015; 25: 130-136.

Examples

```
## Not run:
data(SWAP)
result <- randomPTF(SWAP$spec, SWAP$pH, fun=WA)
plot(result, cex=0.6)
print(result)
# parallel version
if (.Platform$OS.type=='windows') {
  library(doParallel)
  registerDoParallel(cores=4)
} else {
  library(doMC)
  registerDoMC(cores=4)
}
system.time(result <- randomPTF(SWAP$spec, SWAP$pH, fun=WA, do.parallel=TRUE, nTF=5000))
## End(Not run)
```

RLGH *Diatom stratigraphic data from the Round Loch of Glenhead, Galloway, Southwest Scotland*

Description

Diatom stratigraphic data from the Round Loch of Glenhead, Galloway, Southwest Scotland from core K05, first published in Allott et al. (1992) and re-analysed in Juggins et al. (1996) and Battarbee et al. (2005). Data are relative abundances (percentages) of a subset of 41 diatom taxa in 20 samples, and includes all taxa with a maximum abundance of 1 percent in any core sample. Dataset is a list with the following named elements: spec diatom relative abundances, depths associated sediment core depths and 210Pb ages. Column names in RLGH$spec are short, 6-character alphanumeric codes for each diatom taxon. RLGH$names contains the full names for each taxon, in the correct order). Note that some rare and low abundance taxa have been removed so the percentages do not sum to 100.

Usage

```
data(RLGH)
```

References

Battarbee, R.W., Monteith, D.T., Juggins, S. Evans, C.D., Jenkins, A. & Simpson, G.L. (2005) Reconstructing pre-acidification pH for an acidified Scottish loch: A comparison of palaeolimnological and modelling approaches. *Environmental Pollution*, **137**, 135-149.

Allott, T.E.H., Harriman, R., & Battarbee, R.W. (1992) Reversibility of acidification at the Round Loch of Glenhead, Galloway, Scotland. *Environmental Pollution*, **77**, 219-225.

Juggins, S., Flower, R., & Battarbee, R. (1996) Palaeolimnological evidence for recent chemical and biological changes in UK Acid Waters Monitoring Network sites. *Freshwater Biology*, **36**, 203-219.

Examples

```
data(RLGH)
names(RLGH$spec)
names(RLGH$depths)
```

---

strat.plot *Plot a stratigraphic diagram*

Description

Plots a diagram of multiple biological, physical or chemical parameters agains depth or time, as used in geology & palaeoecology.

Usage

```
strat.plot (d, yvar=NULL, scale.percent=FALSE, graph.widths=1, minmax=NULL,
            scale.minmax=TRUE, xLeft=0.07, xRight=1, yBottom=0.07,
            yTop=0.8, title="", cex.title=1.8, y.axis=TRUE, x.axis=TRUE,
            min.width=5, ylim=NULL, y.rev=FALSE, y.tks=NULL, y.tks.labels=NULL,
            ylabel="", cex.ylabel=1, cex.yaxis=0.8, xSpace=0.01, x.pc.inc=10,
            x.pc.lab=TRUE, x.pc.omit0=TRUE, wa.order="none", plot.line=TRUE,
            col.line="black", lwd.line=1, col.symb="black", plot.bar=TRUE,
            lwd.bar=1, col.bar="grey", sep.bar=FALSE, bar.back=FALSE,
            plot.poly=FALSE, col.poly="grey", col.poly.line=NA, lwd.poly=1,
         plot.symb=FALSE, symb.pch=19, symb.cex=1, x.names=NULL, cex.xlabel=1.1,
        srt.xlabel=90, mgp=NULL, ylabPos=2, cex.axis=.8, clust=NULL, clust.width=0.1,
        orig.fig=NULL, exag=FALSE, exag.mult=5, col.exag="grey90", exag.alpha=0.2,
           col.bg=NULL, fun1=NULL, fun2=NULL, add=FALSE, omitMissing=TRUE, ...)

addZone (x, upper, lower=NULL, ...)

addClustZone(x, clust, nZone, ...)
```

Arguments

d	a matrix or data frame of variables to plot.
yvar	a vector of depths or ages to use for the y-axis (defaults to sample number).
scale.percent	logical to scale x-axes for (biological) percentage data.
graph.widths	a vector of relative widths for each curve, used if scale.percent=FALSE.
minmax	2 * nvar matrix of min and max values to scale each curve if scale.percent=FALSE.
scale.minmax	logical to show only min and max values on x-axes (to avoid label crowding).
xLeft, xRight, yBottom, yTop	x, y position of plot on page, in relative units.
title	main title for plot.
x.names	character vector of names for each graph, of same length as ncol(d).
cex.title	size of label for title.
y.axis	logical to control drawing of left-hand y-axis scale. Defaults to TRUE.
x.axis	logical or logical vector to control drawing of x-axes. Defaults to TRUE.
min.width	minimum upper value of x-axis when scaled for percent data.
ylim	numeric vector of 2 values to control limist of y-axis. Defaults to data range.
y.rev	logical to reverse y-axis. Defaults to FALSE.
y.tks	numerical vector listing values of y-axis ticks.
y.tks.labels	character vector listing values of y-axis labels.
ylabel	label for y-axis.
ylabPos	position for y-axis label.
cex.ylabel, cex.yaxis	text size for y-axis labels and values.
xSpace	space between graphs, in relative units.
x.pc.inc	increment for x-axis values when scale.percent is TRUE.
x.pc.lab	logical to control drawing of x-axis values when scale.percent is TRUE.
x.pc.omit0	logical to omit initial zero x-axis label when scale.percent is TRUE.
wa.order	"none", "topleft" or "bottomleft", to sort variables according to the weighted average with y.
plot.line, plot.poly, plot.bar, plot.symb	logical flags to plot graphs as lines, silhouettes, bars or symbols.
col.line, col.poly.line	colour of lines and silhouette outlines. Can be a single colour or a vector of colours, one for each graph.
col.poly	silhouette fill colour. Can be a single colour or a vector of colours, one for each graph.
lwd.line, lwd.poly, lwd.bar	line widths for line, silhouette or bar graphs.
col.bar	colour of bars in a bar graph. col.bar can be a vector to specify colours of individual bars or graphs.

col.symb	symbol colour.
sep.bar	If true, colours in col.bar are applied to individual bars, otherwise individual graphs.
bar.back	logical to plot bars behind (TRUE) or on top (FALSE: default) of curves.
cex.xlabel	size of label for variable names.
srt.xlabel	rotation angle for variable names.
symb.pch, symb.cex	symbol type / size.
exag	logical to add exaggerated curves when plot.poly=TRUE. Can be a single value or a vector to add exaggeration to individual curves.
exag.mult	multiplier for exaggerated curves. Can be a single value or a vector to control exaggeration to individual curves.
col.exag	colour for exaggerated curves. Can be a single value, a vector to control colour of individual curves, or "auto" for transparent version of main curve.
exag.alpha	alpha channel for transparent exaggerated curves when col.exag="auto".
mgp	value of mgp for x-axes. See par for details.
cex.axis	text size for x-axis labels. See par for details.
clust	an constrained classification object of class chclust to add to plot.
fun1, fun2	custom functions to add additional features to curve. Can be a single function applied to all curves or a vector to apply individual functions to individual curves. fun1 draws behind curves, fun2 draws on top of curves.
clust.width	width of dendrogram to add to right of plot, in relative units.
orig.fig	fig values to specify area of window in which to place diagram. See par for details. Defaults to whole window.
add	logical to contol drawing of new page. See par for details. Defaults to FALSE in which a call to strat.plot will start a new diagram. Set to TRUE to add a diagram to an existing plot.
x	a stratigraphic diagram object produced by strat.plot.
upper, lower	upper and (optional) lower limits of a zone to add to an existing stratigraphic diagram.
nZone	number of zones to draw.
omitMissing	remove missing values before plotting. Defaults to TRUE.
col.bg	background colour for each curve.
...	further graphical arguments.

Details

strat.plot plots a series of variables in a stratigraphic diagram. Diagrams can be plotted as line graphs and / or bar charts. Samples are plotted on the y-axis by sample number by default but may be plotted against sample age or depth by specifying a variable for yvar. Margins of the plotting area can be changed using xLeft, xRight, yBottom and yTop. A dendrogram produced by chclust can be added to the right of the diagram.

The function addZone can be used to add a horizontal line or box to an existing plot, and addClustZone will add a specified number of zones from a dendrogram (see examples).

The function uses fig to split the screen and may be incompatible with par(mfrow) and split.screen.

Value

Returns a list containing the following objects:

box	Vector of 4 values giving the coordinates of the left, right, bottom and top of the plotting area, in relative units.
usr	Ranges of the plotting area, in data units.
yvar	Variable used for the y-axis.
ylim	Limits of the y-axis.

Author(s)

Steve Juggins

See Also

chclust.

Examples

```
library(vegan) ## decorana
data(RLGH)
## Not run:
# create appropriately sized graphics window
windows(width=12, height=7) # quartz() on Mac, X11 on linux

## End(Not run)
# remove less abundant taxa
mx <- apply(RLGH$spec, 2, max)
spec <- RLGH$spec[, mx > 3]
depth <- RLGH$depths$Depth
#basic stratigraphic plot
strat.plot(spec, y.rev=TRUE)
#scale for percentage data
strat.plot(spec, y.rev=TRUE, scale.percent=TRUE)
# plot by sample depth
strat.plot(spec, yvar = depth, y.rev=TRUE, scale.percent=TRUE,
title="Round Loch of Glenhead", ylabel="Depth (cm)")
# add a dendromgram from constrained cluster analysis
diss <- dist(sqrt(RLGH$spec/100)^2)
clust <- chclust(diss, method="coniss")
# broken stick model suggest 3 significant zones
bstick(clust)
x <- strat.plot(spec, yvar = depth, y.rev=TRUE,
scale.percent=TRUE, title="Round Loch of Glenhead", ylabel="Depth (cm)",
clust=clust)
# add zones
addClustZone(x, clust, 3, col="red")
# use fig to contol diagram size and position
x <- strat.plot(spec, xRight = 0.7, yvar = depth, y.rev=TRUE,
scale.percent=TRUE, title="Round Loch of Glenhead", ylabel="Depth (cm)")
# add curves for first two DCA components of diatom data
```

```
dca <- decorana(spec, iweigh=1)
sc <- scores(dca, display="sites", choices=1:2)
strat.plot(sc, xLeft = 0.7, yvar = depth, y.rev=TRUE, xRight=0.99,
y.axis=FALSE, clust=clust, clust.width=0.08, add=TRUE)

# Use custom function to add smooth to curve

sm.fun <- function(x, y, i, nm) {
  tmp <- data.frame(x=y, y=x)
  tmp <- na.omit(tmp)
  lo <- lowess(tmp, f=0.3)
  lines(lo$y, lo$x, col="red", lwd=1)
}

x <- strat.plot(spec, yvar = depth, y.rev=TRUE, scale.percent=TRUE,
title="Round Loch of Glenhead", ylabel="Depth (cm)", fun1=sm.fun)

# Pollen diagram using built-in Abernethy Forest dataset
data(aber)
depth <- aber$ages$Age
spec <- aber$spec

# basic silhouette plot
strat.plot(spec, yvar = depth, y.rev=TRUE, scale.percent=TRUE, ylabel="Depth (cm)",
plot.poly=TRUE, col.poly="darkgreen", col.poly.line=NA)

# now with horizontal lines at sample positions
strat.plot(spec, yvar = depth, y.rev=TRUE, scale.percent=TRUE, ylabel="Depth (cm)",
plot.poly=TRUE, col.poly="darkgreen", plot.bar="Full", col.poly.line=NA)

# add exaggerated curves
strat.plot(spec, yvar = depth, y.rev=TRUE, scale.percent=TRUE, ylabel="Depth (cm)",
plot.poly=TRUE, col.poly="darkgreen", plot.bar="Full", col.poly.line=NA, exag=TRUE)

# use different colours for trees
xx <- 1:ncol(spec)
cc <- ifelse(xx < 8, "darkgreen", "darkred")
strat.plot(spec, yvar = depth, y.rev=TRUE, scale.percent=TRUE, ylabel="Depth (cm)",
plot.poly=TRUE, col.poly=cc, plot.bar="Full", col.poly.line=NA, exag=TRUE, col.exag="auto")
```

SWAP *SWAP surface sediment diatom data and lake-water pH.*

Description

SWAP (Surface Water Acidification Programme) surface sediment diatom data from Birks et al. (1990) and Stevenson et al. (1990). Dataset is a list with the following named elements: (spec) diatom relative abundances for 277 taxa in 167 surface samples, (pH) associated lake-water pH. Column names in spec are short, 6-character alphanumeric codes for each diatom taxon. SWAP$names contains the full names for each taxon, in the correct order).

Usage

```
data(SWAP)
```

Source

Birks, H.J.B., Line, J.M., Juggins, S., Stevenson, A.C., & ter Braak, C.J.F. (1990) Diatoms and pH reconstruction. *Philosophical Transactions of the Royal Society of London*, **B 327**, 263-278.

Stevenson, A.C., Juggins, S., Birks, H.J.B., Anderson, D.S., Anderson, N.J., Battarbee, R.W., Berge, F., Davis, R.B., Flower, R.J., Haworth, E.Y., Jones, V.J., Kingston, J.C., Kreiser, A.M., Line, J.M., Munro, M.A.R., & Renberg, I. (1991) *The Surface Waters Acidification Project Palaeolimnology Programme: Modern Diatom / Lake-Water Chemistry Data-Set* ENSIS Ltd, London.

Examples

```
data(SWAP)
names(SWAP$spec)
hist(SWAP$pH)
```

---

utils *Utility functions.*

Description

Utility functions to perform simple computations, transformations, formatting etc.

Usage

```
make.dummy(fact)

dummy2factor(x)

Hill.N2(df, margin=2)

site.summ(y, max.cut=c(2, 5, 10, 20))

sp.summ(y, n.cut=c(5, 10, 20))
```

Arguments

fact	a factor to convert to a matrix of dummy variables.
x	a matrix or data frame of dummy variables to convert to a factor.
df	a data frame of species abundance data.
margin	margin to calculate over: 1 = by rows, 2 = by columns.
y	data frame or matrix of species by sites data.
n.cut	cut levels of abundance for species summary (see below).
max.cut	cut levels of occurence for species sumamry.

Details

Function `make.dummy` converts a factor into a matrix of dummy (1/0) variables. `dummy2factor` converts a matrix or data frame of dummy variables into a factor.

Function `Hill.N2` returns Hill's N2 values for species or samples for a given species by sites dataset (Hill 1973).

Value

`make.dummy` returns a matrix of dummay variables. `dummy2factor` returns a factor.

`Hill.N2` returns a numeric vector of N2 values.

`sp.summ` returns a matrix with columns for the number of occurences, Hill's N2 and maximum abundance of each species, and the number of occurences at abundance greater than the cut levels given in `n.cut`.

`sam.summ` returns a matrix with columns for the number of taxa, Hill's N2, maximum value and site total of each site (sample), and the number of taxa in each site with abundance greater than the cut levels given in `max.cut`.

Author(s)

Steve Juggins

References

Hill, M.O. (1973) Diversity and evenness: a unifying notation and its consequences. *Ecology*, **54**, 427-432.

WA *Weighted averaging (WA) regression and calibration*

Description

Functions for reconstructing (predicting) environmental values from biological assemblages using weighted averaging (WA) regression and calibration.

Usage

```
WA(y, x, mono=FALSE, tolDW = FALSE, use.N2=TRUE, tol.cut=.01,
      check.data=TRUE, lean=FALSE)

WA.fit(y, x, mono=FALSE, tolDW=FALSE, use.N2=TRUE, tol.cut=.01,
       lean=FALSE)

## S3 method for class 'WA'
 predict(object, newdata=NULL, sse=FALSE, nboot=100,
      match.data=TRUE, verbose=TRUE, ...)
```

```
## S3 method for class 'WA'
crossval(object, cv.method="loo", verbose=TRUE, ngroups=10,
      nboot=100, h.cutoff=0, h.dist=NULL, ...)

## S3 method for class 'WA'
performance(object, ...)

## S3 method for class 'WA'
rand.t.test(object, n.perm=999, ...)

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

## S3 method for class 'WA'
summary(object, full=FALSE, ...)

## S3 method for class 'WA'
plot(x, resid=FALSE, xval=FALSE, tolDW=FALSE, deshrink="inverse",
      xlab="", ylab="", ylim=NULL, xlim=NULL, add.ref=TRUE,
      add.smooth=FALSE, ...)

## S3 method for class 'WA'
residuals(object, cv=FALSE, ...)

## S3 method for class 'WA'
coef(object, ...)

## S3 method for class 'WA'
fitted(object, ...)
```

Arguments

y	a data frame or matrix of biological abundance data.
x, object	a vector of environmental values to be modelled or an object of class WA.
newdata	new biological data to be predicted.
mono	logical to perform monotonic curvilinear deshrinking.
tolDW	logical to include regressions and predictions using tolerance downweighting.
use.N2	logical to adjust tolerance by species N2 values.
tol.cut	tolerances less than tol.cut are replaced by the mean tolerance.
check.data	logical to perform simple checks on the input data.
lean	logical to exclude some output from the resulting models (used when cross-validating to speed calculations).
full	logical to show head and tail of output in summaries.
match.data	logical indicate the function will match two species datasets by their column names. You should only set this to FALSE if you are sure the column names match exactly.

resid	logical to plot residuals instead of fitted values.
xval	logical to plot cross-validation estimates.
xlab, ylab, xlim, ylim	additional graphical arguments to plot.WA.
deshrink	deshrinking type to show in plot.
add.ref	add 1:1 line on plot.
add.smooth	add loess smooth to plot.
cv.method	cross-validation method, either "loo", "lgo", "bootstrap" or "h-block".
verbose	logical to show feedback during cross-validation.
nboot	number of bootstrap samples.
ngroups	number of groups in leave-group-out cross-validation.
h.cutoff	cutoff for h-block cross-validation. Only training samples greater than h.cutoff from each test sample will be used.
h.dist	distance matrix for use in h-block cross-validation. Usually a matrix of geographical distances between samples.
sse	logical indicating that sample specific errors should be calculated.
n.perm	number of permutations for randomisation t-test.
cv	logical to indicate model or cross-validation residuals.
...	additional arguments.

Details

Function WA performs weighted average (WA) regression and calibration. Weighted averaging has a long history in ecology and forms the basis of many biotic indices. It WAs popularised in palaeolimnology by ter Brakk and van Dam (1989) and Birks et al. (1990) follwoing ter Braak & Barendregt (1986) and ter Braak and Looman (1986) who demonstrated it's theroetical properties in providing a robust and simple alternative to species response modelling using Gaussian logistic regression. Function WA predicts environmental values from sub-fossil biological assemblages, given a training dataset of modern species and envionmental data. It calculates estimates using inverse and classical deshrinking, and, optionally, with taxa downweighted by their tolerances. Prediction errors and model complexity (simple or tolerance downweighted WA) can be estimated by cross-validation using crossval which implements leave-one out, leave-group-out, or bootstrapping. With leave-group out one may also supply a vector of group memberships for more carefully designed cross-validation experiments.

Function predict predicts values of the environemntal variable for newdata or returns the fitted (predicted) values from the original modern dataset if newdata is NULL. Variables are matched between training and newdata by column name (if match.data is TRUE). Use compare.datasets to assess conformity of two species datasets and identify possible no-analogue samples.

Function rand.t.test performs a randomisation t-test to test the significance of the difference in cross-validation RMSE between tolerance-downweighted and simple WA, after van der Voet (1994).

WA has methods fitted and rediduals that return the fitted values (estimates) and residuals for the training set, performance, which returns summary performance statistics (see below), coef

which returns the species coefficients (optima and tolerances), and `print` and `summary` to summarise the output. `WA` also has a `plot` method that produces scatter plots of predicted vs observed measurements for the training set.

Value

Function `WA` returns an object of class `WA` with the following named elements:

`coefficients`	species coefficients ("optima" and, optionally, "tolerances").
`deshrink.coefficients`	deshrinking coefficients.
`tolDW`	logical to indicate tolerance downweighted results in model.
`fitted.values`	fitted values for the training set.
`call`	original function call.
`x`	environmental variable used in the model.

If function `predict` is called with `newdata=NULL` it returns the fitted values of the original model, otherwise it returns a list with the following named elements:

`fit`	predicted values for `newdata`.

If sample specific errors were requested the list will also include:

`fit.boot`	mean of the bootstrap estimates of newdata.
`v1`	standard error of the bootstrap estimates for each new sample.
`v2`	root mean squared error for the training set samples, across all bootstram samples.
`SEP`	standard error of prediction, calculated as the square root of v1^2 + v2^2.

Function `crossval` also returns an object of class `WA` and adds the following named elements:

`predicted`	predicted values of each training set sample under cross-validation.
`residuals.cv`	prediction residuals.

Function `performance` returns a matrix of performance statistics for the WA model. See `performance`, for a description of the summary.

Author(s)

Steve Juggins

References

Birks, H.J.B., Line, J.M., Juggins, S., Stevenson, A.C., & ter Braak, C.J.F. (1990) Diatoms and pH reconstruction. *Philosophical Transactions of the Royal Society of London*, **B, 327**, 263-278.

ter Braak, C.J.F. & Barendregt, L.G. (1986) Weighted averaging of species indicator values: its efficiency in environmental calibration. Mathematical Biosciences, 78, 57-72.

ter Braak, C.J.F. & Looman, C.W.N. (1986) Weighted averaging, logistic regression and the Gaussian response model. *Vegetatio*, **65**, 3-11.

ter Braak, C.J.F. & van Dam, H. (1989) Inferring pH from diatoms: a comparison of old and new calibration methods. *Hydrobiologia*, **178**, 209-223.

van der Voet, H. (1994) Comparing the predictive accuracy of models uing a simple randomization test. *Chemometrics and Intelligent Laboratory Systems*, **25**, 313-323.

See Also

WAPLS, MAT, and compare.datasets for diagnostics.

Examples

```
# pH reconstruction of core K05 from the Round Loch of Glenhead,
# Galloway, SW Scotland. This lake has become acidified over the
# last c. 150 years

data(SWAP)
data(RLGH)
spec <- SWAP$spec
pH <- SWAP$pH
core <- RLGH$spec
age <- RLGH$depths$Age

fit <- WA(spec, pH, tolDW=TRUE)
# plot predicted vs. observed
plot(fit)
plot(fit, resid=TRUE)

# RLGH reconstruction
pred <- predict(fit, core)

#plot the reconstructio
plot(age, pred$fit[, 1], type="b")

# cross-validation model using bootstrapping
## Not run:
fit.xv <- crossval(fit, cv.method="boot", nboot=1000)
par(mfrow=c(1,2))
plot(fit)
plot(fit, resid=TRUE)
plot(fit.xv, xval=TRUE)
plot(fit.xv, xval=TRUE, resid=TRUE)

# RLGH reconstruction with sample specific errors
pred <- predict(fit, core, sse=TRUE, nboot=1000)

## End(Not run)
```

WAPLS *Weighted averaging partial least squares (WAPLS) regression and calibration*

Description

Functions for reconstructing (predicting) environmental values from biological assemblages using weighted averaging partial least squares (WAPLS) regression and calibration.

Usage

```
WAPLS(y, x, npls=5, iswapls=TRUE, standx=FALSE, lean=FALSE,
      check.data=TRUE, ...)

WAPLS.fit(y, x, npls=5, iswapls=TRUE, standx=FALSE, lean=FALSE)

## S3 method for class 'WAPLS'
 predict(object, newdata=NULL, sse=FALSE, nboot=100,
      match.data=TRUE, verbose=TRUE, ...)

## S3 method for class 'WAPLS'
crossval(object, cv.method="loo", verbose=TRUE, ngroups=10,
      nboot=100, h.cutoff=0, h.dist=NULL, ...)

## S3 method for class 'WAPLS'
performance(object, ...)

## S3 method for class 'WAPLS'
rand.t.test(object, n.perm=999, ...)

## S3 method for class 'WAPLS'
screeplot(x, rand.test=TRUE, ...)

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

## S3 method for class 'WAPLS'
summary(object, full=FALSE, ...)

## S3 method for class 'WAPLS'
plot(x, resid=FALSE, xval=FALSE, npls=1,
      xlab="", ylab="", ylim=NULL, xlim=NULL, add.ref=TRUE,
      add.smooth=FALSE, ...)

## S3 method for class 'WAPLS'
residuals(object, cv=FALSE, ...)
```

```
## S3 method for class 'WAPLS'
coef(object, ...)

## S3 method for class 'WAPLS'
fitted(object, ...)
```

Arguments

y	a data frame or matrix of biological abundance data.
x, object	a vector of environmental values to be modelled or an object of class wa.
newdata	new biological data to be predicted.
iswapls	logical logical to perform WAPLS or PLS. Defaults to TRUE = WAPLS.
standx	logical to standardise x-data in PLS, defaults to FALSE.
npls	number of pls components to extract.
check.data	logical to perform simple checks on the input data.
match.data	logical indicate the function will match two species datasets by their column names. You should only set this to FALSE if you are sure the column names match exactly.
lean	logical to exclude some output from the resulting models (used when cross-validating to speed calculations).
full	logical to show head and tail of output in summaries.
resid	logical to plot residuals instead of fitted values.
xval	logical to plot cross-validation estimates.
xlab, ylab, xlim, ylim	additional graphical arguments to plot.wa.
add.ref	add 1:1 line on plot.
add.smooth	add loess smooth to plot.
cv.method	cross-validation method, either "loo", "lgo", "bootstrap" or "h-block".
verbose	logical show feedback during cross-validation.
nboot	number of bootstrap samples.
ngroups	number of groups in leave-group-out cross-validation, or a vector contain leave-out group menbership.
h.cutoff	cutoff for h-block cross-validation. Only training samples greater than h.cutoff from each test sample will be used.
h.dist	distance matrix for use in h-block cross-validation. Usually a matrix of geographical distances between samples.
sse	logical indicating that sample specific errors should be calculated.
rand.test	logical to perform a randomisation t-test to test significance of cross validated components.
n.perm	number of permutations for randomisation t-test.
cv	logical to indicate model or cross-validation residuals.
...	additional arguments.

Details

Function WAPLS performs partial least squares (PLS) or weighted averaging partial least squares (WAPLS) regression. WAPLS was first described in ter Braak and Juggins (1993) and ter Braak et al. (1993) and has since become popular in palaeolimnology for reconstructing (predicting) environmental values from sub-fossil biological assemblages, given a training dataset of modern species and envionmental data. Prediction errors and model complexity (number of components) can be estimated by cross-validation using crossval which implements leave-one out, leave-group-out, or bootstrapping. With leave-group out one may also supply a vector of group memberships for more carefully designed cross-validation experiments.

Function predict predicts values of the environemntal variable for newdata or returns the fitted (predicted) values from the original modern dataset if newdata is NULL. Variables are matched between training and newdata by column name (if match.data is TRUE). Use compare.datasets to assess conformity of two species datasets and identify possible no-analogue samples.

WAPLS has methods fitted and rediduals that return the fitted values (estimates) and residuals for the training set, performance, which returns summary performance statistics (see below), coef which returns the species coefficients, and print and summary to summarise the output. WAPLS also has a plot method that produces scatter plots of predicted vs observed measurements for the training set.

Function rand.t.test performs a randomisation t-test to test the significance of the cross-validated components after van der Voet (1994).

Function screeplot displays the RMSE of prediction for the training set as a function of the number of components and is useful for estimating the optimal number for use in prediction. By default screeplot will also carry out a randomisation t-test and add a line to scree plot indicating percentage change in RMSE with each component annotate with the p-value from the randomisation test.

Value

Function WAPLS returns an object of class WAPLS with the following named elements:

coefficients	species coefficients (the updated "optima").
meanY	weighted mean of the environmental variable.
iswapls	logical indicating whether analysis was WAPLS (TRUE) or PLS (FALSE).
T	sample scores.
P	variable (species) scores.
npls	number of pls components extracted.
fitted.values	fitted values for the training set.
call	original function call.
x	environmental variable used in the model.
standx, meanT sdx	additional information returned for a PLS model.

Function crossval also returns an object of class WAPLS and adds the following named elements:

predicted	predicted values of each training set sample under cross-validation.

`residuals.cv` prediction residuals.

If function `predict` is called with `newdata=NULL` it returns the fitted values of the original model, otherwise it returns a list with the following named elements:

`fit` predicted values for `newdata`.

If sample specific errors were requested the list will also include:

`fit.boot` mean of the bootstrap estimates of newdata.

`v1` standard error of the bootstrap estimates for each new sample.

`v2` root mean squared error for the training set samples, across all bootstram samples.

`SEP` standard error of prediction, calculated as the square root of v1^2 + v2^2.

Function `performance` returns a matrix of performance statistics for the WAPLS model. See `performance`, for a description of the summary.

Function `rand.t.test` returns a matrix of performance statistics together with columns indicating the p-value and percentage change in RMSE with each higher component (see van der Veot (1994) for details).

Author(s)

Steve Juggins

References

ter Braak, C.J.F. & Juggins, S. (1993) Weighted averaging partial least squares regression (WA-PLS): an improved method for reconstructing environmental variables from species assemblages. *Hydrobiologia*, **269/270**, 485-502.

ter Braak, C.J.F., Juggins, S., Birks, H.J.B., & Voet, H., van der (1993). Weighted averaging partial least squares regression (WA-PLS): definition and comparison with other methods for species-environment calibration. In *Multivariate Environmental Statistics* (eds G.P. Patil & C.R. Rao), pp. 525-560. Elsevier Science Publishers.

van der Voet, H. (1994) Comparing the predictive accuracy of models uing a simple randomization test. *Chemometrics and Intelligent Laboratory Systems*, **25**, 313-323.

See Also

`WA`, `MAT`, `performance`, and `compare.datasets` for diagnostics.

Examples

```
data(IK)
spec <- IK$spec
SumSST <- IK$env$SumSST
core <- IK$core

fit <- WAPLS(spec, SumSST)
fit
```

```
# cross-validate model
fit.cv <- crossval(fit, cv.method="loo")
# How many components to use?
rand.t.test(fit.cv)
screeplot(fit.cv)

#predict the core
pred <- predict(fit, core, npls=2)

#plot predictions - depths are in rownames
depth <- as.numeric(rownames(core))
plot(depth, pred$fit[, 2], type="b", ylab="Predicted SumSST", las=1)

# predictions with sample specific errors
## Not run:
pred <- predict(fit, core, npls=2, sse=TRUE, nboot=1000)
pred

## End(Not run)
```

Index