

Package ‘palaeoSig’

February 20, 2015

Type Package

Title Significance Tests for Palaeoenvironmental Reconstructions

Version 1.1-3

Date 2015-02-4

Author Richard Telford, Mathias Trachsel

Maintainer Richard Telford <Richard.Telford@bio.uib.no>

Depends vegan

Suggests fields

Imports TeachingDemos, rioja, mgcv, MASS

Description Tests if quantitative palaeoenvironmental reconstructions are statistically significant.

License GPL-3

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2015-02-17 19:05:40

R topics documented:

palaeoSig-package	2
abundances	3
agelme	4
anamorph	5
arctic.pollen	6
centipede.plot	7
cor.mat.fun	8
coverage.plot	9
Hill.N2.core	10
joinsig	11
make.env	13
make.set	14
mat.h	15

multi.mat	16
obs.cor	17
plot.fittedAgelme	20
predict.agelme	20
randomTF	22
rne	24
species	26
STOR	28

Index	29
--------------	-----------

palaeoSig-package	<i>Significance Tests of Quantitative Palaeoenvironmental Reconstructions</i>
-------------------	---

Description

Significance tests for quantitative palaeoenvironmental reconstructions derived from transfer functions. Functions from the **autocorTF** package are now included in **palaeoSig**.

Details

This package includes:

significance tests for quantitative palaeoenvironmental reconstructions ([randomTF](#), [obs.cor](#))

graphical methods to show autocorrelation in transfer functions ([rne](#))

null model test of transfer functions performance in a spatially autocorrelated environment - see vignette.

h-block cross-validation for MAT ([mat.h](#)) - the plan is to move this to **rioja**

Several functions have from **autocorTF** version 1.0 and **palaeoSig** version 1.0 have been rewritten or replaced with more flexible or user friendly functions. See `news(package="palaeoSig")` for details.

See also my blog at <https://quantpalaeo.wordpress.com/>

Package:	palaeoSig
Type:	Package
Version:	1.1-2
Date:	2013-04-17
License:	GPL - 3
LazyLoad:	yes

Author(s)

Richard Telford <richard.telford@bio.uib.no>

References

Telford, R. J. and Birks, H. J. B. (2009) Evaluation of transfer functions in spatially structured environments. *Quaternary Science Reviews* **28**: 1309–1316. DOI: 10.1016/j.quascirev.2008.12.020

Telford, R. J. and Birks, H. J. B. (2011) A novel method for assessing the statistical significance of quantitative reconstructions inferred from biotic assemblages. *Quaternary Science Reviews* **30**: 1272–1278. DOI: 10.1016/j.quascirev.2011.03.002

See Also

[rioja, rda](#)

abundances

Generates abundances

Description

Generates species abundances based on species response functions and environmental variables.

Usage

```
abundances(env, spp, nc)
```

Arguments

env	Environmental variables. Usually generated by make.env . Users may as well supply own environmental variables.
spp	Species parameters. Usually generated by species .
nc	Number of counts to be simulated. If omitted no simulation of the counting process is carried out.

Value

spp	Data frame containing species abundances.
env	Data frame containing environmental variables.

Author(s)

Mathias Trachsel and Richard J Telford

References

Minchin, P.R. (1987) Multidimensional Community Patterns: Towards a Comprehensive Model. *Vegetatio*, 71, 145-156.

See Also

[make.env](#), [species](#)

Examples

```
spec <- species(nspp = 30, ndim = 10, Amax = runif, fun = runif, xpar = c(-50, 150),
  srange = 200, alpha = 4, gamma = 4)
env.var <- make.env(100, elen = rep(100, 10), emean = rep(50, 10), edistr = 'uniform',
  ndim = 10)
spec.abun <- abundances(env.var, spec, 200)
```

agelme

Estimation of the relationship between Calibrated age and depth

Description

Estimates the relationship of Calibrated age and depth for paleorecords. The function uses a smooth spline of the mgcv library by Simon Wood. It produces predicted confidence interval for the relationship approximating a mixed effect model, as there are two levels of uncertainty, i.e. within dated object and between dated objects.

Usage

```
agelme(depup, depdo, bpup, bpdo, use, weights=c(1, rep(0, length(depup)-1)),
  vspan=1, k=length(depup)-1, m=2, diagnostic=FALSE)
```

Arguments

depup	The upper depths of the dated slides
depdo	The lower depths of the dated slides
bpup	The younger calibrated ages of the dated slides
bpdo	The older calibrated ages of the dated slides
use	Logical vector of dates to include in the model. Default is to use all.
weights	Weights to be used for the estimation, default is fixed top-layer followed by inverse variance of within dated object
vspan	The span to be used for the diagnostic plots, default span = 1
k	Number of base function to start the shrinkage in the gam estimation procedure
m	The order of penalty for the term, i.e. the degree of continuity at the knots (default, m = 2 gives cubic smooth spline)
diagnostic	Logical, should diagnostic plots be made.

Details

Note that the fixation of the top layer is done by a weight = 1, whereas the other weights follows inverse variance within object.

The diagnostic plots is used to check the quality of the estimation and to see if there is a need for an assumption of between object variance proportional to mean. The latter however is rarely encountered for paleodata.

Value

tdf	Degrees of freedom used by the cubic smooth spline, a vector with first value for constant variance and second vector for variance equal to mu.
weights	A vector of the weights used by the cubic smooth spline
RES	A vector of the Residual sum of squares
Models	A list with the models from the cubic smooth spline, constant and mu variance, respectively
Data	A data.frame including the data used for the estimation

Author(s)

Einar Heegaard <einar.heegaard@bio.uib.no>

References

Heegaard, E., Birks, HJB. & Telford, RJ. 2005. Relationships between calibrated ages and depth in stratigraphical sequences: an estimation procedure by mixed-effect regression. *The Holocene* 15: 612-618

Examples

```
data(STOR)

fit.mod <- with(STOR, agelme(depthup, depthdo, cageup, cagedo))

#Predicting using the constant variance model,
#for each cm between 70 and 400 cm.
fit.pre <- predict(fit.mod, 1, 70:400)
plot(fit.pre)
```

anamorph

Anamorph

Description

Creates functions that transform arbitrary distributions into a Gaussian distributions, and vice versa.

Usage

```
anamorph(x, k, plot = F)
```

Arguments

x	data
k	number of Hermite polynomials
plot	plot

Details

Increasing k can give a better fit.

Value

Returns two function in a list

xtog	Function to transform arbitrary variable x into a Gaussian distribution
gtox	The back transformation

Author(s)

Richard Telford <Richard.Telford@bio.uib.no>

References

Wackernagel, H. (2003) *Multivariate Geostatistics*. 3rd edition, Springer-Verlag, Berlin.

Examples

```
set.seed(666)
x <- c(rnorm(50, 0, 1), rnorm(50, 6, 1))
hist(x)
ana.fun <- anamorph(x, 30, plot = TRUE)
xg <- ana.fun$xtog(x)
qqnorm(xg)
qqline(xg)
all.equal(x, ana.fun$gtox(xg))
```

arctic.pollen

Arctic Pollen and associated environmental data

Description

Arctic pollen percent data and associated environmental data

Usage

```
data(arctic.pollen)
```

Format

arctic.pollen A data frame with 828 observations on the percentage of 39 pollen taxa

arctic.env Environmental data for the pollen sites

Source

Data extracted from North American Pollen Database and New *et al.* (2002) by Fr chet te *et al.* (2008). Following Fr chet te (Pers. Comm.), three duplicate sites have been deleted.

References

Fréchette, B., de Vernal, A., Guiot, J., Wolfe, A. P., Miller, G. H., Fredskild, B., Kerwin, M. W. and Richard, P. J. H. (2008) Methodological basis for quantitative reconstruction of air temperature and sunshine from pollen assemblages in Arctic Canada and Greenland. *Quaternary Science Reviews* **27**, 1197–1216 doi:10.1016/j.quascirev.2008.02.016

Examples

```
data(arctic.pollen)
data(arctic.env)
```

centipede.plot	<i>Plots species optima and tolerances</i>
----------------	--

Description

Plots species optima and tolerances as calculated by a [WA](#) model.

Usage

```
centipede.plot(x,keep=TRUE, xlab="", xlim, ...)
```

Arguments

x	A WA model.
keep	A logical vector of which taxa to include in the plot. Defaults to all taxa.
xlab	The xlab of the plot
xlim	xlims for the plot. Defaults to a sensible range.
...	Extra arguments to plot, for example of the cex, col, and pch of the points.

Details

Extracts, sorts and plots [WA](#) optima and tolerances. Tends only to work well when there are a reasonable number of taxa, otherwise it is difficult to read the names on the axis. If rare taxa are not excluded via the keep argument, the tol.cut argument in [WA](#) may need to be set to prevent very small tolerances of these taxa.

Value

None

Author(s)

Richard Telford <Richard.Telford@bio.uib.no>

Examples

```
require(rioja)
data(SWAP)
mod<-WA(SWAP$spec, SWAP$pH, toLDW=TRUE)
keep<-colSums(SWAP$spec)>40#only common taxa
opar<-par(no.readonly = TRUE)
par(mar=c(5.1,6.1,2.1,1.1))#extra space on left side of plot for names
centipede.plot(x=mod, keep=keep, pch=20, cex.axis=.7)
par(opar)
```

cor.mat.fun	<i>Generates correlation matrix</i>
-------------	-------------------------------------

Description

Generates a correlation matrix for the environmental variables generated in [make.env](#) and for correlated species optima in [species](#). Only used when correlated environmental variables or optima are generated.

Usage

```
cor.mat.fun(ndim, cors)
```

Arguments

ndim	Number of environmental variables that are subsequently generated with make.env .
cors	List of correlations between environmental variables. Each element of the list consists of three numbers, the first two numbers indicate the variables that are correlated, the third number is the correlation coefficient. If correlations between two variables are omitted the correlation remains 0.

Value

Correlation matrix

Author(s)

Mathias Trachsel

See Also

[make.env](#), [species](#)

Examples

```
correlations <- list(c(1,2,0.5),c(1,4,0.1),c(2,5,0.6))
cor.mat <- cor.mat.fun(5,correlations)
```

`coverage.plot`*Coverage of fossil taxa in modern calibration set*

Description

A simple diagnostic plot showing the coverage of fossil taxa in modern calibration set

Usage

```
coverage.plot(mod, fos, rare=5, identify=FALSE)
```

Arguments

<code>mod</code>	Modern calibration set species data
<code>fos</code>	Fossil species data
<code>rare</code>	Value of Hill's N2 below which species are highlighted
<code>identify</code>	Whether to identify selected taxa.

Details

Finds the maximum abundance of fossil taxa and plots this against the maximum abundance the taxa in the modern calibration set. Taxa with a Hill's N2 less than `rare` in the calibration set are highlighted in blue. Taxa absent from the calibration set are highlighted in red. If there are many taxa above the 1:1 line, or important fossil taxa have a low N2 in the calibration set, reconstructions should be interpreted with caution.

Value

An invisible data.frame with the modern and fossil maximum abundances and N2 in the calibration set.

Author(s)

Richard Telford <Richard.Telford@bio.uib.no>

Examples

```
require(rioja)
data(SWAP)
data(RLGH)
coverage.plot(mod=SWAP$spec, fos=RLGH$spec, identify=FALSE)
```

Hill.N2.core

Calculate the effective number of species in the fossil data

Description

Gives a measure of the species diversity in the fossil data.

Usage

```
Hill.N2.core(spp)
```

Arguments

spp Species data

Details

Uses [Hill.N2](#) from the rioja package

Value

Minimum, first quartile and median effective number of species

Note

If the effective number of species is small, WA based reconstructions are unlikely to be significant, and MAT based reconstructions should be tested instead.

Author(s)

Richard Telford

References

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

See Also

[Hill.N2](#)

Examples

```
require(rioja)
data(RLGH)

Hill.N2.core(RLGH$spec)
```

jointsig

Test if two variables jointly control changes in fossil data

Description

Generates synthetic variables with different proportion of two environmental variables, and tests how much variance in the fossil data reconstructions of these synthetic variables explain.

Usage

```
jointsig(spp, fos, var1, var2, method = "randomTF", n = 99, r = 32, ...)
## S3 method for class 'js'
plot(x, names.v1, names.v2, ...)
```

Arguments

spp	Data frame of modern training set species data, transformed as required for example with sqrt
fos	Data frame of fossil species data, with same species codes and transformations as spp
var1	Training set environmental variable 1.
var2	Training set environmental variable 2.
n	number of random training sets used to generate the null model
method	Which significance test to use. Current options are randomTF and obs.cor. The latter may give strange results - use with caution.
r	How many synthetic variables to make. More is better but slower.
x	Output from jointsig
names.v1	Vector length 2 with names of the end members of the first environmental variable, e.g., c("cold", "warm") for temperature.
names.v2	Ditto for the second variable.
...	Other arguments to the significance test (some of these are required) or plot.

Details

With method="randomTF", the function calculates the proportion of variance in the fossil data explained by transfer function reconstructions of synthetic variables. The synthetic variables are composed of two environmental variables, weighted between -1 and +1, so to represent a circle. This is compared with a null distribution of the proportion of variance explained by reconstructions based on random environmental variables. Any transfer function in the rioja library can be used. With method="obs.cor", the aim is the same, but the function reports the correlation between the species weighted average optima on the synthetic variables and the species first axis scores. This option has some pathological behaviour and should probably be avoided.

Value

A list with components

PCA	The unconstrained ordination of the fossil data.
preds	A list of the containing the reconstructions for each environmental variable.
MAX	Proportion of the variance explained by the first axis of the unconstrained ordination. This is the maximum amount that a reconstruction of a single variable can explain.
EX	The proportion of the variance in the fossil data explained by each reconstruction.
sim.ex	The proportion of variance explained by each of the random environmental variables.
sig	The p-value of each reconstruction.

Author(s)

Richard Telford <richard.telford@bio.uib.no>

References

Unpublished method - use with caution. Can give spurious results with weighted averaging.

See Also

[randomTF,obs.cor](#)

Examples

```
require(rioja)
data(SWAP)
data(RLGH)

rlgh.js <- jointsig(spp = sqrt(SWAP$spec), fos = sqrt(RLGH$spec), var1 = SWAP$pH,
  var2 = sample(SWAP$pH), method = "randomTF", n = 99, r = 32, fun = WA, col = 1)
#nonsense second variable

plot(rlgh.js, c("acid", "alkaline"), c("down", "up"))
```

make.env	<i>Generates environmental variables</i>
----------	--

Description

Generates environmental variables used for generating species abundances. Environmental variables may be correlated, and may follow different distributions.

Usage

```
make.env(n, elen, emean, edistr, ecor, ndim)
```

Arguments

n	Number of samples to be generated.
elen	Range of the environmental variables. Single number or vector of length ndim.
emean	Mean of the environmental variables. Single number or vector of length ndim.
edistr	Distribution of the environmental variables. Currently 'uniform' and 'Gaussian' are supported.
ecor	Correlation matrix of the environmental variables supplied by user. Object generated by cor.mat.fun . If omitted environmental variables are not correlated.
ndim	Number of environmental variables to generate.

Value

Matrix of environmental variables. n rows and ndim columns.

Author(s)

Mathias Trachsel and Richard J. Telford

References

Minchin, P.R. (1987) Multidimensional Community Patterns: Towards a Comprehensive Model. *Vegetatio*, 71, 145-156.

See Also

[cor.mat.fun](#)

Examples

```
env.vars <- make.env(100, elen = rep(100, 10), emean = rep(50, 10), edistr = 'uniform', ndim = 10)
```

`make.set`*Function to simulate species data*

Description

Function to simulate species data following Minchin (1987). This functions generates species response functions, simulates environmental variables and simulates species assemblages based on species response functions and environmental variables. Users can as well supply own species parameters (e.g. when simulating calibration and fossil datasets) and own environmental variables.

Usage

```
make.set(ndim, n, elen, emean, edistr, ecor, cnt, spec, env,...)
```

Arguments

<code>ndim</code>	Number of environmental variables to generate.
<code>n</code>	Number of samples to be generated.
<code>elen</code>	Range of the environmental variables. Single number or vector of length <code>ndim</code> .
<code>emean</code>	Mean of the environmental variables. Single number or vector of length <code>ndim</code> .
<code>edistr</code>	Distribution of the environmental variables. Currently 'uniform' and 'Gaussian' are supported.
<code>ecor</code>	Correlation matrix of the environmental variables supplied by user. Object generated by <code>cor.mat.fun</code> . If omitted environmental variables are uncorrelated.
<code>cnt</code>	Number of counts to be simulated.
<code>spec</code>	Users may supply their own species paramters.
<code>env</code>	Users may supply their own environmental variables.
<code>...</code>	Arguments passed to <code>species</code>

Value

<code>spp</code>	Species abundance data.
<code>env</code>	Environmental variables used to simulate species abundance data.
<code>spec</code>	Species parameters.

Author(s)

Mathias Trachsel and Richard J. Telford

References

Minchin, P.R. (1987) Multidimensional Community Patterns: Towards a Comprehensive Model. *Vegetatio*, 71, 145-156.

See Also

[make.env](#), [species](#), [cor.mat.fun](#)

Examples

```
calib <- make.set(nspp = 90, ndim = 3, Amax = runif, fun = runif, xpar = c(-50, 150),
  srange = 400, alpha = 4, gamma = 4, n = 100, elen = rep(100, 3), emean = rep(50, 3),
  edistr = 'uniform', cnt = 1000)

# Provide species parameters generated above, so that the fossil data use the
# same species parameters.
fos <- make.set(ndim = 3, n = 100, elen = rep(100, 3), emean = rep(50, 3), edistr = 'uniform',
  cnt = 1000, spec = calib$spec)

# Supplying own environmental variables and species parameters.
env.vars <- make.env(100, elen = rep(100, 3), emean = rep(50, 3), edistr = 'uniform', ndim = 3)
fos <- make.set(cnt = 1000, spec = calib$spec, env = env.vars)
```

 mat.h

MAT with minimum geographic distance for analogues

Description

h-block cross-validation for MAT.

Usage

```
mat.h(y, x, noanalogues, geodist, thresh)
```

Arguments

y	Community data or distance matrix
x	Environmental variable
noanalogues	Number of analogues
geodist	Geographical distances
thresh	Threshold distance below which sites are excluded during cross validation

Details

Analogues closer than the geographic distance specified by thresh are not excluded, reducing the effect of spatial autocorrelation on the performance statistics. H-block cross-validation is also implemented in the **rioja** package. This function can be faster if used many times as it can take a pre-calculated distance matrix.

Value

Predictions for each site.

Author(s)

Richard Telford <Richard.Telford@bio.uib.no>

References

Telford, R. J. and Birks, H. J. B. (2009) Evaluation of transfer functions in spatially structured environments. *Quaternary Science Reviews* **28**: 1309–1316. DOI: 10.1016/j.quascirev.2008.12.020

Examples

```
require(fields)
data(arctic.env)
data(arctic.pollen)
arctic.dist <- rdist.earth(arctic.env[,c("Longitude", "Latitude")], miles=FALSE)

h0 <- mat.h(arctic.pollen, arctic.env$tjul, noanalogues = 10, arctic.dist, thresh=-1)
h100 <- mat.h(arctic.pollen, arctic.env$tjul, noanalogues = 10, arctic.dist, thresh=100)

cor(h0, arctic.env$tjul)
cor(h100, arctic.env$tjul)

sqrt(mean((h0 - arctic.env$tjul)^2))
sqrt(mean((h100 - arctic.env$tjul)^2))
```

multi.mat

MAT for multiple variables

Description

MAT for many environmental variables simultaneously. More efficient than calculating them separately for each variable.

Usage

```
multi.mat(training.spp, envs, core.spp, noanalogues = 10, method = "sq-chord",
run = "both")
```


Arguments

training.spp	Community data
envs	Environmental variables - or simulations
core.spp	Optional fossil data to make predictions for
noanalogues	Number of analogues to use
method	distance metric to use
run	Return LOO predictions or predictions for fossil data

Value

Matrix of predictions

Author(s)

Richard Telford <Richard.Telford@bio.uib.no>

References

Telford, R. J. and Birks, H. J. B. (2009) Evaluation of transfer functions in spatially structured environments. *Quaternary Science Reviews* **28**: 1309–1316. DOI: 10.1016/j.quascirev.2008.12.020

Examples

```
data(arctic.env)
data(arctic.pollen)

mMAT <- multi.mat(arctic.pollen, arctic.env[,9:67], noanalogues = 5)
```

obs.cor	<i>Weighted correlation between weighted averaging optima and constrained ordination species scores.</i>
---------	--

Description

obs.cor calculates the weighted correlation between the species weighted average optima and the axis one species scores of an ordination constrained by the WA reconstruction.

Usage

```
obs.cor(spp, env, fos, ord = rda, n = 99, min.occure = 1)
## S3 method for class 'obscor'
plot(x, xlab, ylab, f = 5, which = 1, label = "env", abun="abun.calib", p.val=0.95, ...)
## S3 method for class 'obscor'
identify(x, labels, ...)
```

Arguments

spp	Data frame of modern training set species data, transformed if required, for example with sqrt
env	Vector of a single environmental variable
fos	Data frame of fossil species data. Species codes and transformations should match those in spp.
ord	Constrained ordination. The default is rda. cca will work with the present code. capscale would require the code to be altered.
n	Number of random trials
min.occure	Minimum number of occurrences of species in the species and fossil data.
x	An obscor object.
xlab	X-axis label if the default is unsatisfactory.
ylab	Y-axis label if the default is unsatisfactory.
f	Scale factor for the abundances, the maximum cex of points for the which=1 plot.
which	Which type of plot. which = 1 gives a plot of RDA scores against species optima. which = 2 gives a histogram showing the null distribution of correlations between RDA scores and species optima, together with the observed correlation.
label	Label for the observed correlation with which = 2
abun	Which species weighting required for plots. See details
p.val	P value to draw a line vertical line at (with which=2)
labels	Labels for the points in identify. By default, the species names from intersection of colnames(spp) and colnames(fos) are used.
...	Other arguments to plot or identify

Details

Obs.cor calculates the weighted correlation between the species WA optima in the calibration set and their ordination axis one scores in the fossil data. Seven different weights for the species are implemented. "abun.fos" - weight by the mean abundance in the fossil data. "abun.calib" - weight by the mean abundance in the calibration data "abun.joint" - weight by the product of the mean abundance in the fossil and calibration data "n2.fos" - weight by the effective number of occurrences (Hill's N2) of each species in the fossil data "n2.calib" - weight by the effective number of occurrences (Hill's N2) of each species in the calibration data "n2.joint" - weight by the product of n2.calib and n2.fos "unweighted" - all species receive same weight. This is unlikely to be the best option but is included for completeness. It is unclear which of these weights is likely to be best, research is in progress. A square root transformation of the species data is often useful. n=99 is too small in practice to give a smooth histogram of the null model. n=999 is better.

Value

obs.cor returns an obscor object, which is a list

ob	Observed correlation. Data.frame with columns Optima, RDA1 and abun containing the species optima, ordination axis 1 scores, and abundance used to weight the species respectively and a vector containing the weighted and un-weighted correlations between species optima and ordination scores.
sim	Matrix with the correlation between species weighted average optima and ordination scores in the first column and the weighted correlation in the second column. Each row represents a different random environmental variable.
sig	p-value for the observed correlation between species weighted average optima and ordination scores for each of the weights.

Note

The test of the weighted correlation between species optima and ordination axis scores is more powerful, especially with a small number of fossil observations, than the test of variance explained in [randomTF](#) but is only applicable to WA and will have a large type II error if there are few species.

Author(s)

Richard Telford <richard.telford@bio.uib.no>

References

Telford, R. J. and Birks, H. J. B. (2011) A novel method for assessing the statistical significance of quantitative reconstructions inferred from biotic assemblages. *Quaternary Science Reviews* **30**: 1272–1278. DOI: [10.1016/j.quascirev.2011.03.002](https://doi.org/10.1016/j.quascirev.2011.03.002)

See Also

[randomTF](#), [WA](#), [MAT](#), [WAPLS](#), [rda](#), [cca](#)

Examples

```
require(rioja)
data(SWAP)
data(RLGH)

rlgh.obs <- obs.cor(spp = sqrt(SWAP$spec), env = SWAP$pH, fos = sqrt(RLGH$spec))
rlgh.obs$sim
plot(rlgh.obs, which=1)
plot(rlgh.obs, which=2)
```

plot.fittedAgelme *Plots fitted agelme model and dates*

Description

Plots fitted agelme model and dates

Usage

```
## S3 method for class 'fittedAgelme'  
plot(x, main, xlab = "Depth",  
     ylab = "Calibrated Age", ...)
```

Arguments

x	Fitted agelme model.
main	Title of the plot.
xlab	xlabel of the plot.
ylab	ylabel of the plot.
...	Other arguments to plot.

Examples

```
data(STOR)  
fit.mod <- with(STOR, agelme(depthup, depthdo, cageup, cagedo))  
#Predicting using the constant variance model,  
#for each cm between 70 and 400 cm.  
fit.pre <- predict(fit.mod, 1, 70:400)  
plot(fit.pre)
```

predict.agelme *Predicts the Calibrated age*

Description

This function uses the output from 'agelme' to predict the Calibrated ages for specified depths.

Usage

```
## S3 method for class 'agelme'  
predict(object, v = 1, depth, ...)
```

Arguments

object	An 'agelme' model
v	Using constant (1) or mu (2) variance
depth	A vector of the depths to be predicted
...	Other arguments, currently unused.

Value

A list with three items

- v Whether constant variance or mu variance used.
- fit A data.frame of the predictions including 95% confidence interval.
 - DepthThe depths for the predicted ages
 - EstagePredicted age
 - LowlimLower 95% confidence interval
 - UpplimUpper 95% confidence interval
 - TsdTotal standard deviation
- data A data.frame containing the age and depth information of the radiocarbon dates.

Author(s)

Einar Heegaard <einar.heegaard@bio.bui.no>

References

Heegaard, E., Birks, HJB. & Telford, RJ. 2005. Relationships between calibrated ages and depth in stratigraphical sequences: an estimation procedure by mixed-effect regression. *The Holocene* 15: 612-618

Examples

```
data(STOR)

fit.mod <- with(STOR, agelme(depthup, depthdo, cageup, cagedo))

#Predicting using the constant variance model,
#for each cm between 70 and 400 cm.
fit.pre <- predict(fit.mod, 1, 70:400)
plot(fit.pre)
```

randomTF	<i>Proportion of variance in the fossil data explained by an environmental reconstruction</i>
----------	---

Description

Calculate the proportion of variance in the fossil data explained by an environmental reconstruction with a constrained ordination. This value is compared with a null distribution calculated as the proportion of variance in the fossil data explained by reconstructions from transfer functions trained on random data.

Usage

```
randomTF(spp, env, fos, n = 99, fun, col, condition, autosim, ord = rda,...)
ModelMaker(spp, env, n=99, fun, autosim, ...)
randomTFmm(fos, modelList, col, condition, ord=rda, ...)
## S3 method for class 'palaeoSig'
plot(x, vnames, top = 0.7, adj = c(0, 0.5), p.val=0.95, ...)
```

Arguments

spp	Data frame of modern training set species data, transformed as required for example with sqrt
env	Data frame of training set environmental variables
fos	Data frame of fossil species data, with same species codes and transformations as spp
n	number of random training sets
fun	Transfer function method. Additional argument can be passed with ...
col	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.
condition	Optional data frame of reconstructions to partial out when testing if multiple independent reconstructions are possible.
autosim	Optional data frame of random values. This is useful if the training set is spatially autocorrelated and the supplied data frame contains autocorrelated random variables. If autosim is missing, the transfer functions are trained on random variables drawn from a uniform distribution.
ord	Constrained ordination method to use. rda is the default, cca should also work. capscale won't work without modifications to the code (or a wrapper).
...	Other arguments to the transfer function. For example to change the distance metric in MAT. Also extra arguments to plot.
modelList	Output of ModelMaker
x	Output from randomTF

vnames	Names of environmental variables
top	Proportion of the figure below the environmental name labels.
adj	Adjust the position that the environmental names are plotted at.
p.val	P value to draw a line vertical line at (with which=2)

Details

The function calculates the proportion of variance in the fossil data explained by the transfer function reconstruction. This is compared with a null distribution of the proportion of variance explained by reconstructions based on random environmental variables. Reconstructions can be partialled out to test if multiple reconstructions are statistically significant. If the environment is spatially auto-correlated, a red-noise null should be used instead of the default white noise null. The red noise environmental variables can be generated with the **gstat** package.

Any transfer function in the **rioja** package can be used. Other methods (e.g. random forests) can be used by making a wrapper function.

If several reconstructions using the same training set are being tested, it can be much faster to make the models once, and use them repeatedly. This can be done with **ModelMaker** and **randomTFmm**. **ModelMaker** does not work with **MAT**.

Value

A list with components

PCA	The unconstrained ordination of the fossil data.
preds	A list of the containing the reconstructions for each environmental variable.
MAX	Proportion of the variance explained by the first axis of the unconstrained ordination. This is the maximum amount that a reconstruction of a single variable can explain.
EX	The proportion of the variance in the fossil data explained by each reconstruction.
sim.ex	The proportion of variance explained by each of the random environmental variables.
sig	The p-value of each reconstruction.

ModelMaker returns a list of models.

Note

If there are only a few fossil levels, [obs.cor](#) might have more power. If there are few taxa, tests on [MAT](#) reconstructions have more statistical power than those based on [WA](#).

Author(s)

Richard Telford <richard.telford@bio.uib.no>

References

Telford, R. J. and Birks, H. J. B. (2011) A novel method for assessing the statistical significance of quantitative reconstructions inferred from biotic assemblages. *Quaternary Science Reviews* **30**: 1272–1278. DOI: 10.1016/j.quascirev.2011.03.002

See Also

[obs.cor](#), [WA](#), [MAT](#), [WAPLS](#), [rda](#), [cca](#)

Examples

```
require(rioja)
data(SWAP)
data(RLGH)

rlghr <- randomTF(spp = sqrt(SWAP$spec), env = data.frame(pH = SWAP$pH),
  fos = sqrt(RLGH$spec), n = 99, fun = WA, col = 1)
rlghr$sig
plot.palaeoSig(rlghr, "pH")
```

 rne

Random, neighbour, environment deletion analysis for transfer function model

Description

Calculates effect of deleting sites from training set at random, from a geographic neighbourhood, or from an environmental neighbourhood. A simple graphical technique for gauging the effect of spatial autocorrelation on the transfer function model.

Usage

```
rne(y, env, geodist, fun, neighbours, subsets = c(1, 0.75, 0.5, 0.25, 0.1), ...)
## S3 method for class 'RNE'
plot(x, which=1, ylim, ...)
```

Arguments

y	Community data, or distance object, or distance matrix
env	Environmental variable
geodist	Matrix of geographical distances between sites
fun	Transfer function
neighbours	Neighbourhood radii
subsets	Proportion of sites to retain in random deletion
...	Arguments passed to fun or plot

x	RNE object to be plotted
which	Which column of the results to plot eg if more than one WAPLS component is calculated
ylim	Y-limits of the plot

Details

Finds the leave-one-out transfer function performance if sites are deleted at random (repeated 10 times to reduce variance in results), from a neighbourhood zone, or by deleting environmentally close sites.

Value

Returns an RNE object, list with two components

random	Performance with random deletion.
neighbour	Performance with deletion by neighbourhood, or environment

Author(s)

Richard Telford <Richard.Telford@bio.uib.no>

References

Telford, R. J. and Birks, H. J. B. (2009) Evaluation of transfer functions in spatially structured environments. *Quaternary Science Reviews* **28**: 1309–1316. DOI: 10.1016/j.quascirev.2008.12.020

Examples

```
require(fields)
require(rioja)
data(arctic.env)
data(arctic.pollen)
```

```
#using just the first 100 sites so code runs in an reasonable time, but still slow
arctic.d <- rdist.earth(arctic.env[1:100,c("Longitude","Latitude")], miles = FALSE)
```

```
arctic.rne <- rne(y = arctic.pollen[1:100,], env = arctic.env$tjul[1:100],
  geodist = arctic.d, fun = MAT, neighbours = c(0,50,200),
  subsets = c(1,.5,.1), k = 5)
plot(arctic.rne)
```

species *Generates species response parameters for n dimensions*

Description

Generates species response parameters to n environmental variables following Minchin (1987).

Usage

```
species(nspp=30, Amax,fun, xpar, srange, alpha = 4,gamma= 4, ndim, sdistr, ocor, odistr)
```

Arguments

nspp	Number of species to be generated.
Amax	Maximum abundance of a species. Amax currently allows three options: i) a function how to generate maximum abundances (e.g. runif, rgamma) ii) a vector of length nspp iii) a single number that is used as maximum abundance for all the species.
fun	Function to generate species optima (e.g. rnorm, runif). The two parameters in xpar are passed to function fun. If omitted species optima are generated at regular intervals between the two values in xpar.
xpar	Two numbers describing a distribution e.g mu and sigma for a normal distribution, lower and upper bound for a random uniform distribution.
srange	Length of the ecological gradient to which individual species respond. Either one number or a matrix with nspp rows and ndim columns. If srange should be different for different environmental variables a simpler solution is to change argument elen in make.env accordingly. E.g. elen =c(100,50,50) when using three environmental gradients.
alpha	Shape parameter of the beta distribution. One number or a matrix with nspp rows and ndim columns.
gamma	Shape parameter of the beta distribution. One number or a matrix with nspp rows and ndim columns.
ndim	Number of environmental variables to which generated species should respond.
sdistr	Users may supply own distributions of species optima. Matrix with nspp rows and ndim columns (as well in the special case of ndim = 1).
ocor	Correlation matrix of the species optima. May be generated by code cor.mat.fun .
odistr	Distribution of the correlated optima either 'uniform' or 'Gaussian'

Details

Details on the exact generation of species response functions from parameters Amax, m, r, gamma and alpha are given in Minchin (1987). Species response curves are determined by five parameters: a parameter determining the maximum abundance (Amax) and one describing the location (m) of this mode. A parameter determining to which environmental range the species respond (srange in

the input *r* in the output) and two parameters (alpha, gamma) describing the shape of the species response function. If alpha = gamma the response curve is symmetric (alpha = gamma = 4, yields Gaussian distributions). Additionally, species optima for several environmental variables may be correlated. Currently this is only possible for gaussian or uniform distributions of species optima. Users may as well supply previously generated optima (e.g. optima similar to a real dataset).

Value

List with *ndim* elements. Each list contains the species response parameters to one environmental gradient.

Author(s)

Mathias Trachsel and Richard J. Telford

References

Minchin, P.R. (1987) Multidimensional Community Patterns: Towards a Comprehensive Model. *Vegetatio*, 71, 145-156.

Examples

```
spec.par <- species(nspp = 30, Amax = runif, srange = 200, fun = runif, xpar = c(-50,150),
ndim = 5, alpha = 4, gamma = 4)
spec.par <- species(nspp = 30, ndim = 3, Amax = runif, xpar = c(-50,150),
srange = 200, alpha = 4, gamma = 4)

# example where srange, alpha and gamma are different for each species and environmental gradient.
spec.par <- species(nspp = 30, ndim = 3, Amax = runif, xpar = c(-50,150),
srange = matrix(ncol = 3, runif(90,100,200)), alpha = matrix(ncol = 3, runif(90,1,5)),
gamma = matrix(ncol = 3, runif(90,1,5)))

# example where species optima are correlated
correlations <- list(c(1,2,0.5),c(1,3,0.3),c(2,3,0.1))
spec.cor.mat <- cor.mat.fun(3,correlations)
spec.par <- species(nspp = 30, ndim = 3, Amax = runif, xpar = c(50,50), srange = 200,
alpha = 4, gamma = 4, ocor = spec.cor.mat, odistr = 'Gaussian')

# example for species response curves (users should alter alpha and gamma)
spec.par <- species(nspp = 1, Amax = 200, srange = 200, fun = runif, xpar = c(50,50),
ndim = 1, alpha = 3, gamma = 1)
env <- -50:150
response <- palaeoSig::make.abundances(env = -50:150, param = spec.par[[1]]$spp)
plot(env, response, type='l')
```

STOR

Storsandsvatnet

Description

Storsandsvatnet is a lake in western Norway. From the sediments a core was obtained, and 11 samples was submitted for radiocarbon dating. The data contain the depths of the slides dated and the younger and older calibrated ages for each slide.

Usage

```
data(STOR)
```

Format

A data frame with 11 observations on the following 4 variables.

depthup The upper border of the dated slide

depthdo The lower border of the dated slide

cageup The younger calibrated age of the dated slide

cagedo The older calibrated age of the dated slide

Details

The calibrated ages is obtained by calibration of the radiocarbon dates. The borders represent mean calibrated age +/- 1 SD of calibrated age.

Source

The data is unpublished and provided by H. John B. Birks <john.birks@bio.uib.no> and Sylvia M. Peglar

References

Heegaard, E., Birks, HJB. & Telford, RJ. 2005. Relationships between calibrated ages and depth in stratigraphical sequences: an estimation procedure by mixed-effect regression. *The Holocene* 15: 612-618

Examples

```
data(STOR)
```

Index

- *Topic **attribute**
 - Hill.N2.core, 10
- *Topic **datagen**
 - abundances, 3
 - cor.mat.fun, 8
 - make.env, 13
 - make.set, 14
 - species, 26
- *Topic **datasets**
 - arctic.pollen, 6
 - STOR, 28
- *Topic **hplot**
 - centipede.plot, 7
 - coverage.plot, 9
 - jointsig, 11
 - obs.cor, 17
 - randomTF, 22
- *Topic **htest**
 - jointsig, 11
 - obs.cor, 17
 - randomTF, 22
- *Topic **manip**
 - anamorph, 5
- *Topic **multivariate**
 - jointsig, 11
 - mat.h, 15
 - multi.mat, 16
 - obs.cor, 17
 - randomTF, 22
 - rne, 24
- *Topic **package**
 - palaeoSig-package, 2
- abundances, 3
- agelme, 4
- anamorph, 5
- arctic.env (arctic.pollen), 6
- arctic.pollen, 6
- cca, 19, 24
- centipede.plot, 7
- cor.mat.fun, 8, 13–15, 26
- coverage.plot, 9
- Hill.N2, 10
- Hill.N2.core, 10
- identify.obscor (obs.cor), 17
- jointsig, 11
- make.env, 3, 8, 13, 15, 26
- make.set, 14
- MAT, 19, 23, 24
- mat.h, 2, 15
- ModelMaker (randomTF), 22
- multi.mat, 16
- obs.cor, 2, 12, 17, 23, 24
- palaeoSig (palaeoSig-package), 2
- palaeoSig-package, 2
- plot.fittedAgelme, 20
- plot.js (jointsig), 11
- plot.obscor (obs.cor), 17
- plot.palaeoSig (randomTF), 22
- plot.RNE (rne), 24
- predict.agelme, 20
- randomTF, 2, 12, 19, 22
- randomTFmm (randomTF), 22
- rda, 3, 19, 24
- rioja, 3
- rne, 2, 24
- species, 3, 8, 14, 15, 26
- STOR, 28
- WA, 7, 19, 23, 24
- WAPLS, 19, 24