Package 'palaeoSig'

October 8, 2026

Type Package

Title Significance Tests for Palaeoenvironmental Reconstructions

Version 2.1-5

URL https://richardjtelford.github.io/palaeoSig/, https://github.com/richardjtelford/palaeoSig

BugReports https://github.com/richardjtelford/palaeoSig/issues

Depends R (>= 4.1)

Suggests units, knitr, rmarkdown, gstat, sf, analogue, testthat (>= 3.0.0)

Imports TeachingDemos, rioja, mgcv, MASS, tibble, dplyr, ggplot2, forcats (>= 1.0.0), assertr, vegan, rlang, ggrepel, tidyr, purrr (>= 1.0.0)

Description Several tests of quantitative palaeoenvironmental reconstructions from microfossil assemblages, including the null model tests of the statistically significant of reconstructions developed by Telford and Birks (2011) <doi:10.1016/j.quascirev.2011.03.002>, and tests of the effect of spatial autocorrelation on transfer function model performance using methods from Telford and Birks (2009) <doi:10.1016/j.quascirev.2008.12.020> and Trachsel and Telford (2016) <doi:10.5194/cp-12-1215-2016>. Age-depth models with generalized mixed-effect regression from Heegaard et al (2005) <doi:10.1191/0959683605hl836rr> are also included.

License GPL-3

LazyLoad yes

VignetteBuilder knitr

Encoding UTF-8

Language en-GB

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Richard Telford [aut, cre, cph],
Mathias Trachsel [ctb]

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

Repository CRAN

Date/Publication 2026-10-07 22:50:02 UTC

Contents

abundances . . . 2
agelme . . . 3
anamorph . . . 5
arctic.pollen . . . 6
Atlantic . . . 7
centipede_plot . . . 7
cor.mat.fun . . . 8
coverage_plot . . . 9
Hill.N2.core . . . 10
jointsig . . . 11
make.env . . . 13
make.set . . . 14
multi.mat . . . 15
obs.cor . . . 16
plot.fittedAgelme . . . 20
plot.RNE . . . 21
predict.agelme . . . 23
randomTF . . . 24
species . . . 27
STOR . . . 29

Index **31**

---

abundances *Generates simulated species 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	Count sum to be simulated. If omitted no simulation of the counting process is carried out.

Value

A list with two elements:

spp	A data frame with species abundances.
env	A data frame with 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. doi:10.1007/BF00039167

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 palaeorecords. The function uses a smooth spline from 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 palaeodata.

Value

A list of class 'agelme' with the following items:

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
Constant	A matrix with the numerical results for the dated points using a constant variance

Muvar	A matrix with the numerical results for the dated points using variance equal to mu
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

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 doi:10.1191/0959683605hl836rr

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 Gaussian distributions, and vice versa.

Usage

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

Arguments

x	vector of data to transform
k	number of Hermite polynomials to use
plot	logical; plot the transformation?

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. doi:10.1007/9783662052945

Examples

```
set.seed(42)
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)
data(arctic.env)
```

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 Fréchette *et al.* (2008). Following Fréchette (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

`Atlantic` *Atlantic core-top foram assemblages*

Description

A dataset containing over 1000 foram assemblages from the Atlantic from Kucera et al (2005) and the 50m SST for the warmest season. Rare taxa and co-located assemblages are removed.

Usage

```
data(Atlantic)
```

Format

A data frame with 1093 rows and 33 variables. summ50 is 50m water temperature of the warmest season

Source

- Foraminifera data doi:10.1594/PANGAEA.227322
- Sea surface temperature data from WOA13 https://www.ncei.noaa.gov/data/oceans/woa/WOA13/DATAv2/temperature/csv/decav/1.00/ file 'woa13_decav_t13mn01v2.csv.gz'

`centipede_plot` *centipede_plot*

Description

Plot of species WA optima and tolerance

Usage

```
centipede_plot(x, spp, minN2 = 1, mult = 1)
```

Arguments

`x`	A tolerance weighted weighted-average model from `WA`
`spp`	data.frame of species data used to train the WA model
`minN2`	numeric giving minimum N2 for inclusion in plot
`mult`	numeric multiplier for the tolerances

Details

Extracts and sorts WA optima and tolerances and generates a ggplot. Tends only to work well when there are a reasonable number of taxa, otherwise it is difficult to read the names on the axis. Rare taxa can be excluded with the minN2 argument. The tol.cut argument in WA may need to be set to prevent very small tolerances in rare taxa. This function is very similar to the caterpillar plot, but produces a ggplot

Value

A ggplot object.

Examples

```
library(rioja)
data(SWAP)
mod <- WA(SWAP$spec, SWAP$pH, tolDW = TRUE)
coef(mod)
centipede_plot(mod, spp = SWAP$spec, minN2 = 20)
```

---

cor.mat.fun *Generate correlation matrix*

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

A 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_plot*

Description

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

Usage

```
coverage_plot(spp, fos, n2_rare = 5, label = NULL)
```

Arguments

spp	data.frame of modern species abundances
fos	data.frame of fossil species abundances
n2_rare	numeric value of Hill's N2 below which species are highlighted as rare
label	numeric label taxa where maximum fossil abundance - maximum modern abundance > label. Defaults to NULL which does not add labels

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

A ggplot object.

Examples

```
data("SWAP", package = "rioja")
data("RLGH", package = "rioja")
coverage_plot(spp = SWAP$spec, fos = RLGH$spec, n2_rare = 5, label = 0)
```

`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`	data.frame of Species data

Details

Uses `rioja::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

`rioja::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
method	Which significance test to use. Current option are randomTF and obs.cor. The latter may give strange results - use with caution.
n	number of random training sets used to generate the null model
r	How many synthetic variables to make. More is better but slower
...	Other arguments to plot
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.

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.

Functions

- plot(js): Plot js object

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 = 49, r = 32, fun = WA, col = 1
)
# nonsense second variable

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

make.env *Generates simulated environmental variables*

Description

Simulates 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. Object can be generated with 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. doi:10.1007/BF00039167

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` *Simulate species data sets*

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 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

`ndim`	Number of environmental variables to generate.
`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. Object generated by `cor.mat.fun`. If omitted environmental variables are uncorrelated.
`cnt`	Count sum to be simulated.
`spec`	Users may supply their own species parameters.
`env`	Users may supply their own environmental variables.
`...`	Arguments passed to `species`

Value

List with three elements:

`spp`	Species abundance data.
`env`	Environmental variables used to simulate species abundance data.
`spec`	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. doi:10.1007/BF00039167

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)
```

---

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

If run = "both", a list with two elements:

jack	Matrix of leave-one-out cross-validation predictions for the calibration set
core	Matrix of predictions for the fossil data

Otherwise, one of these matrices is returned.

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 *Weighted correlation between weighted averaging optima and constrained ordination species scores.*

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.occur = 1,
  autosim,
  permute = FALSE
)

## S3 method for class 'obscor'
plot(
  x,
  xlab,
  ylab,
  f = 5,
  which = 1,
  variable_names = "env",
  abun = "abun.calib",
  p_val = 0.05,
  ...
)

## S3 method for class 'obscor'
identify(x, labels, ...)

## S3 method for class 'obscor'
autoplot(
  object,
  which = 1,
  variable_names = "env",
  abun = "abun.calib",
  p_val = 0.05,
  nbins = 20,
  top = 0.7,
  ...
)
```

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 method to use. rda is the default, cca should also work. capscale won't work without modifications to the code (or a wrapper).
n	number of random training sets. More is better.
min.occur	Minimum number of occurrences of species in the species and fossil data.
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, and permute is FALSE, the transfer functions are trained on random variables drawn from a uniform distribution.
permute	logical value. Generate random environmental variables by permuting existing variable. Only possible if there is only one environmental variable and autosim is missing.
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.
variable_names	Name of environmental variable (only 1 currently) for the label on 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)
...	Other arguments to plot or identify
labels	Labels for the points in identify. By default, the species names from intersection of colnames(spp) and colnames(fos) are used.
object	An obscor object.
nbins	integer giving number of bins for the histogram
top	Proportion of the figure below the environmental name labels.

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 unweighted 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.
`sigs`	p-value for the observed correlation between species weighted average optima and ordination scores for each of the weights.

Functions

- `plot(obscor)`: Plots for obscor object
- `identify(obscor)`: Identify species on obs.cor plot
- `autoplot(obscor)`: autoplot for obscor object

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, that 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@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

`randomTF`, `WA`, `rda`, `cca`

Examples

```
require(rioja)
data(SWAP)
data(RLGH)
rlgh.obs <- obs.cor(
  spp = sqrt(SWAP$spec),
  env = SWAP$pH,
  fos = sqrt(RLGH$spec),
  n = 49 # low number for speed
)
rlgh.obs$sig
plot(rlgh.obs, which = 1)
plot(rlgh.obs, which = 2)

require(ggplot2)
autoplot(rlgh.obs, which = 1)
autoplot(rlgh.obs, which = 2, variable_names = "pH")
```

---

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	x axis label of the plot.
ylab	y axis label 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)
```

`plot.RNE` *Random, neighbour, environment deletion analysis for transfer function models*

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

```
## S3 method for class 'RNE'
plot(x, which = 1, ylim, ...)

rne(
  y,
  env,
  geodist,
  fun,
  neighbours,
  subsets = c(1, 0.75, 0.5, 0.25, 0.1),
  ...
)
```

Arguments

`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
`...`	Arguments passed to fun
`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

Details

Finds the leave-one-out transfer function performance if sites are deleted at random, from a neighbourhood zone, or by deleting environmentally close sites.

Prior to version 2.1, this function would repeat the random removal 10 times to reduce variance in results. This is no longer done as the variance is small for large training sets, it took a long time, and treats one treatment of the data differently.

Value

Returns an RNE object, list with two components

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

Functions

- `plot(RNE)`: Plot RNE object

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(rioja)
require(sf)
data(arctic.env)
data(arctic.pollen)

# using just the first 100 sites so that code runs quickly
# (about 15 seconds for all 828 sites)

# convert environmental data into an sf object
arctic.env <- st_as_sf(
  x = arctic.env,
  coords = c("Longitude", "Latitude"),
  crs = 4326
)

# find great circle distances and remove units
arctic.dist <- st_distance(arctic.env[1:100, ]) |>
  units::set_units("km") |>
  units::set_units(NULL)

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

plot(arctic.rne)
```

`predict.agelme` *Predicts the Calibrated age for agelme models*

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 with the model predictions, containing columns:

Depth The depths for the predicted ages

Estage Predicted age

Lowlim Lower 95 % confidence interval

Upplim Upper 95 % confidence interval

Tsd Total standard deviation

`data` A data.frame containing the age and depth information of the radiocarbon dates.

Author(s)

Einar Heegaard

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 doi:10.1191/0959683605hl836rr

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 *Proportion of variance in the fossil data explained by an environmental reconstruction*

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 = NULL,
  autosim,
  ord = rda,
  permute = FALSE,
  models,
  make_models = FALSE,
  ...
)

## S3 method for class 'palaeoSig'
plot(x, variable_names, top = 0.7, adj = c(0, 0.5), p_val = 0.05, ...)

## S3 method for class 'palaeoSig'
autoplot(object, variable_names, nbins = 20, top = 0.7, p_val = 0.05, ...)
```

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 or vector with single environmental variable
fos	Data frame of fossil species data, with same species codes and transformations as spp
n	number of random training sets. More is better.
fun	Transfer function method. Additional arguments to fun 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 of the reconstructions 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, and permute is FALSE, 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).
permute	logical value. Generate random environmental variables by permuting existing variable. Only possible if there is only one environmental variable and autosim is missing.
models	list of models made by randomTF with argument make_models = TRUE
make_models	logical, should a list of transfer functions trained on random data be returned
...	Other arguments to the transfer function. For example to change the distance metric in MAT. Also extra arguments to plot.
x	Output from randomTF
variable_names	Names of environmental variables. If missing, taken from env data.frame.
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)
object	Output from randomTF
nbins	integer giving number of bins for the histogram

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 autocorrelated, a red-noise null should be used instead of the default white noise null. 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 reconstructions from several sites are to be tested using the same training set it can be much faster to train the models on random environmental data once and then use them repeatedly. This can be done with `make_models = TRUE` and then running `randomTF` again giving the resultant models to the `models` argument. `make_models` does not work with MAT.

For some transfer function methods, including `WA`, the code can be made somewhat faster by coercing the modern and fossil species data to matrices (`spp <- as.matrix(spp)`), otherwise `WA` has to do this repeatedly. With `MAT`, this should not be done as it might cause an error.

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.

If `make_models = TRUE`, a list of transfer function models is returned.

`autoplot.palaeoSig` returns a `ggplot2` object

Functions

- `plot(palaeoSig)`: Plot palaeoSig object
- `autoplot(palaeoSig)`: autoplot function for palaeoSig object

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@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$spec), n = 49, fun = WA, col = "WA.inv"
)
rlghr$sig
plot(rlghr, "pH")

require("ggplot2")
autoplot(rlghr, "pH")
```

---

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 approximately 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. doi:10.1007/BF00039167

See Also

make.env, abundances

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 are 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 doi:10.1191/0959683605hl836rr

Index

∗ **attribute**
Hill.N2.core, 10
∗ **datagen**
abundances, 2
cor.mat.fun, 8
make.env, 13
make.set, 14
species, 27
∗ **datasets**
arctic.pollen, 6
Atlantic, 7
STOR, 29
∗ **hplot**
jointsig, 11
obs.cor, 16
randomTF, 24
∗ **htest**
jointsig, 11
obs.cor, 16
randomTF, 24
∗ **manip**
anamorph, 5
∗ **multivariate**
jointsig, 11
multi.mat, 15
obs.cor, 16
plot.RNE, 21
randomTF, 24

abundances, 2, *29*
agelme, 3
anamorph, 5
arctic.env *(*arctic.pollen*)*, 6
arctic.pollen, 6
Atlantic, 7
autoplot.obscor *(*obs.cor*)*, 16
autoplot.palaeoSig *(*randomTF*)*, 24

capscale, *18*, *25*
caterpillar, *8*
cca, *18, 19*, *25*, *27*
centipede_plot, 7
cor.mat.fun, 8, *13–15*, *28*
coverage_plot, 9

ggplot, *8, 9*

Hill.N2.core, 10

identify.obscor *(*obs.cor*)*, 16

jointsig, 11

make.env, *3*, *8, 9*, 13, *15*, *28, 29*
make.set, 14
MAT, *25–27*
multi.mat, 15

obs.cor, *12*, 16, *26, 27*

plot.fittedAgelme, 20
plot.js *(*jointsig*)*, 11
plot.obscor *(*obs.cor*)*, 16
plot.palaeoSig *(*randomTF*)*, 24
plot.RNE, 21
predict.agelme, 23

randomTF, *12*, *19*, 24
rda, *18, 19*, *25*, *27*
rioja::Hill.N2(), *10*
rne *(*plot.RNE*)*, 21

species, *3*, *8, 9*, *14, 15*, 27
STOR, 29

WA, *7, 8*, *19*, *26, 27*
WAPLS, *25*, *27*