Package ‘PCPS’

January 15, 2020

Type Package

Title Principal Coordinates of Phylogenetic Structure

Version 1.0.7

Date 2020-01-15

Author Vanderlei Julio Debastiani

Maintainer Vanderlei Julio Debastiani <vanderleidebastiani@yahoo.com.br>

Depends SYNCSA (&gt;= 1.3.4)

Imports ape, picante, phylobase, vegan, RcppArmadillo, stats, graphics, parallel, nlme

Description Set of functions for analysis of Principal Coordinates of Phylogenetic Structure (PCPS).

License GPL-2

Encoding UTF-8

RoxygenNote 7.0.2

Collate 'matrix.p.sig.R' 'pcps.R' 'pcps.sig.R' 'FUN.ADONIS.R'
'FUN.GLM.R' 'FUN.GLS.marginal.R' 'FUN.GLS.sequential.R'
'FUN.LME.marginal.R' 'FUN.LME.sequential.R' 'FUN.MANTEL.R'
'FUN.RDA.R' 'check.formula.R' 'define.clade.R'
'organize.pcps.R' 'pcoa.sig.R' 'pcps.curve.R'
'pcpc.curve.calc.R' 'plot.pcps.R' 'plot.pcpscurve.R'
'print.pcoasig.R' 'print.pcps.R' 'print.pcpscurve.R'
'print.pcppsig.R' 'print.summarypcoasig.R'
'print.summarypcps.R' 'scores.pcps.R' 'select.pcpsmethod.R'
'self.belonging.R' 'summary.pcoasig.R' 'summary.pcps.R'
'summary.pcpscurve.R' 'wmmdscale.org.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2020-01-15 22:00:02 UTC
R topics documented:

check.formula .......................................................... 2
define.clade ............................................................. 3
matrix.p.null ............................................................ 4
matrix.p.sig ............................................................... 5
mutate.names.matrix.p.null ............................................. 13
organize.pcps ............................................................ 13
pca.sig ................................................................. 15
pcps ................................................................. 17
pcps.curve ............................................................... 20
select.pcpsmethod ....................................................... 23
self.belonging ........................................................... 23
wcmdscale.org .......................................................... 25

Index 26

---

check.formula  

Internal function

Description

Internal function to check the validity of left hand side in a formula object.

Usage

check.formula(formula, vectornames)

Arguments

formula  An object of class formula.
vectornames  A vector with names to check the index.

Value

The index of left side of the formula in vectornames.
**define.clade**

**Define clade**

**Description**

Function to define groups (clades) in a phylogenetic tree.

**Usage**

```r
define.clade(tree, threshold, time, method = c("threshold", "time"))
```

**Arguments**

- `tree`: Phylogenetic tree.
- `threshold`: A threshold value to form the groups.
- `time`: A cutting height (age) to form the groups.
- `method`: Method to define the clades, "threshold" or "time".

**Details**

In the method threshold the total length of phylogenetic tree is used as cutting factor. If threshold is near to zero the cutting is near the root, if threshold near to one cutting is near the tips.

The phylogenetic tree must contain the node labels for the function work. Use the `makeNodeLabel` for defining node labels in a flexible way.

**Value**

- `clades`: Tips and their clades.
- `height`: The cutting height (age).

**Author(s)**

Vanderlei Julio Debastiani <vanderleidebastiani@yahoo.com.br>

**See Also**

- `makeNodeLabel`

**Examples**

```r
require(ape)
trees<-makeNodeLabel(rcoal(10))
clades<-define.clade(tree, threshold = 0.8, method = "threshold")
clades
plot.phylo(tree, show.node.label = TRUE)
abline(v = clades$height)
```
matrix.p.null  

Auxiliar function to generate sets of null P matrix or null PCPS

Description

Auxiliar function to generate sets of null P matrix or null PCPS used in matrix.p.sig or pcps.sig. The result are long lists with permuted matrices.

Usage

matrix.p.null(
  comm,  
  phylodist,  
  runs = NULL,  
  calpcpcs = FALSE,  
  method = "bray",  
  squareroot = TRUE,  
  adjpcps = FALSE,  
  choices = NULL
)

Arguments

comm  
Community data, with species as columns and sampling units as rows. This matrix can contain either presence/absence or abundance data.

phylodist  
Matrix containing phylogenetic distances between species.

runs  
Number of matrix will be generated (Default runs = NULL).

calpcpcs  
Logical argument (TRUE or FALSE) to specify if generate the PCPS (Default calpcpcs = FALSE).

method  
Dissimilarity index, as accepted by vegdist (Default dist = "bray").

squareroot  
Logical argument (TRUE or FALSE) to specify if use square root of dissimilarity index (Default squareroot = TRUE).

adjpcps  
Logical argument (TRUE or FALSE) to specify if return fitted PCPS (Default adjpcps = FALSE).

choices  
Numeric vector to choose the PCPS to adjust (Default pcps.choices = NULL).

Value

call  
The arguments used.

P.obs  
Observed phylogeny-weighted species composition matrix.

pcps.obs  
Observed principal coordinates of phylogenetic structure (PCPS).

permutation.site  
A matrix with sequence of permutation for site shuffle null model, each permutation in one row.
matrix.p.sig

permutation.taxa
A matrix with sequence of permutation for taxa shuffle null model, each permutation in one row.

P.null.site
A list with each permuted P matrix according with site shuffle null model.

P.null.taxa
A list with each permuted P matrix according with taxa shuffle null model.

pcps.null.site
A list with each permuted PCPS according with site shuffle null model.

pcps.null.taxa
A list with each permuted PCPS according with taxa shuffle null model.

pcps.null.taxa.adj
A list with each permuted PCPS (adjusted) according with taxa shuffle null model.

Author(s)
Vanderlei Julio Debastiani <vanderleidebastiani@yahoo.com.br>

See Also

matrix.p, pcps, matrix.p.sig, pcps.sig

matrix.p.sig

Association between phylogeny-weighted species composition and environmental predictors

Description
Analyses to relate an environmental gradient to the phylogenetic assembly of species across a meta-community by means of phylogenetic fuzzy weighting.

Usage

matrix.p.sig(
  comm,
  phylodist,
  envir,
  checkdata = TRUE,
  FUN,
  runs = 999,
  parallel = NULL,
  newname = "pcps",
  ...
)

pcps.sig(
  comm,
  phylodist,
  envir,
checkdata = TRUE,
method = "bray",
squareroot = TRUE,
FUN,
choices,
runs = 999,
parallel = NULL,
newname = "pcps",
...
)

FUN.ADONIS(x, envir, method.p, sqrt.p = TRUE, formula, return.model = FALSE)

FUN.GLM(x, envir, formula, ..., return.model = FALSE)

FUN.GLS.marginal(x, envir, formula, ..., return.model = FALSE)

FUN.GLS.sequential(x, envir, formula, ..., return.model = FALSE)

FUN.LME.marginal(x, envir, formula, ..., return.model = FALSE)

FUN.LME.sequential(x, envir, formula, ..., return.model = FALSE)

FUN.MANTEL(
  x,
  envir,
  method.p,
  method.envir,
  sqrt.p = TRUE,
  ..., 
  return.model = FALSE
)

FUN.RDA(x, envir, return.model = FALSE)

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

**Arguments**

**comm**  Community data, with species as columns and sampling units as rows. This matrix can contain either presence/absence or abundance data. Alternatively, comm can be an object of class metacommunity.data, an alternative way to set all data.frames/matrices. When you use the class metacommunity.data the arguments phylodist and envir must not be specified. See details.

**phylodist**  Matrix containing phylogenetic distances between species.

**envir**  A matrix or data.frame with environmental variables for each community, with variables as columns and sampling units as rows. See Details and Examples.
matrix.p.sig

checkdata  Logical argument (TRUE or FALSE) to check if species sequence in the community data follows the same order as the one in the phylodist matrix and if sampling units in the community data follows the same order as the one in the environmental data (Default checkdata = TRUE).

FUN  An object of class function to perform the analysis. See Details and Examples.

runs  Number of permutations for assessing significance.

parallel  Number of parallel processes or a predefined socket cluster done with parallel package. Tip: use detectCores() (Default parallel = NULL).

newname  New name to be replaced in object returned by matrix.p.null (Default newname = 'pcps').

...  Other arguments passed to FUN function. See Details and Examples.

method  Dissimilarity index, as accepted by vegdist (Default dist = "bray").

squareroot  Logical argument (TRUE or FALSE) to specify if use square root of dissimilarity index (Default squareroot = TRUE).

choices  Numeric vector to choose the PCPS used in analysis. See Details and Examples.

x  An object of class pcps sig or other object to apply the function passed by FUN. See Details.

method.p  Resemblance index between communities based on P matrix, as accepted by vegdist. Used in FUN.MANTEL, FUN.ADONIS, FUN.ADONIS2.global and FUN.ADONIS2.margin analysis. See Details and Examples.

sqrt.p  Logical argument (TRUE or FALSE) to specify if use square root of dissimilarity P matrix. Used in FUN.MANTEL, FUN.ADONIS, FUN.ADONIS2.global and FUN.ADONIS2.margin analysis. See Details and Examples (Default sqrt.p = TRUE).

formula  An object of class formula. Used in FUN.GLM, FUN.ADONIS, FUN.ADONIS2.global, FUN.ADONIS2.margin, FUN.GLS.marginal, FUN.GLS.sequential, FUN.LME.marginal and FUN.LME.sequential analysis. See Details and Examples.

return.model  Must not be specified. See Details.

method.envir  Resemblance index between communities based on environmental variables, as accepted by vegdist. Used in FUN.MANTEL analysis. See Details and Examples.

Details

Each metacommunity is submitted to phylogenetic fuzzy weighting, generating a matrix that describing the phylogeny-weighted species composition of the communities (matrix.p). The function matrix.p.sig test directly the association this matrix with the environmental predictors. The pairwise dissimilarities are submitted to Mantel test (mantel) or ADONIS test (adonis or adonis2) to evaluate the influence of an environmental gradient on species dispersion across the communities. The function pcps.sig generates principal coordinates of phylogenetic structure (pcps) and use a single axis for run a generalized linear model (GLM, glm), linear model using generalized least squares (GLS, gls), linear mixed-effects models (LME, lme) or use set of axis for run a distance-based redundancy analysis (db-RDA, rda).

The sequence species show up in the community data matrix must be the same as they show up in the phylogenetic distance matrix and, similarly, the sequence of communities in the community data
matrix must be the same as that in the environmental data. The function `organize.pcps` organizes the data, placing the matrices of community, phylogenetic distance and environmental data in the same order. The function use of function `organize.pcps` is not required for run the functions, but is recommended. In this way the arguments `comm` and `phylodist` can be specified them as normal arguments or by passing them with the object returned by the function `organize.pcps` using, in this case only the argument `comm`. Using the object returned by `organize.pcps`, the `comm` argument is used as an alternative way of entering to set all data.frames/matrices, and therefore the arguments `phylodist` and `envir` must not be specified.

The significance is obtained via two null models, one that shuffles sites across the environmental gradient and another that shuffles terminal tips (taxa) across the phylogenetic tree. The first null model (site shuffle) shuffles the site position across the environmental gradient and rerun the same model, generating a null F value (or r value in Mantel test). The second null model (taxa shuffle), shuffles terminal tips across the phylogenetic tree and generates a null matrix containing phylogeny-weighted species composition and rerun the same model, generating another null F value. In the `pcps.sig` function are generated set of null PCPS and each null PCPS (or set of PCPS in RDA) is submitted to a procrustean adjustment (see `procrustes`), and the fitted values between observed PCPS and null PCPS is obtained. The adjusted null PCPS is used to rerun the model, generating another null F value. The observed F value (or r value) is compared independently with both null sets of F values (or r value) to generate a probability value of the original F value being generated merely by chance according to each null model.

**The argument FUN**

The type of analysis performed by this function is specified using the argument `FUN`. The current version of package includes ten predefined function, however additional small functions can be easy specify. All this function uses the environmental variables to analyze the association between phylogeny-weighted species composition and environmental predictors. For matrix P analysis, in `matrix.p.sig` function, the predefined functions available are `FUN.MANTEL`, `FUN.ADONIS`, `FUN.ADONIS2.global` and `FUN.ADONIS2.margin`. For PCPS analysis, in `pcps.sig` function, the predefined functions available are `FUN.GLM`, `FUN.RDA`, `FUN.GLS.marginal`, `FUN.GLS.sequential`, `FUN.LME.marginal` and `FUN.LME.sequential`. The significance for each null model is performed as described here, NOT using p value of basic functions.

**FUN.MANTEL**

Mantel test that can be used in matrix P analysis. The arguments `method.p` and `sqrt.p` are specified for determine resemblance index between communities based on P matrix. The argument `method.envir` is specified to determine resemblance index between communities based on environmental variables. The significance is assess using r value, see more in `mantel`.

**FUN.ADONIS**

Multivariate analysis of variance that can be used in matrix P analysis. The arguments `method.p` and `sqrt.p` are specified for determine resemblance index between communities based on P matrix. The argument `formula` is specified, where the left hand side gives the resemblance data, right hand side gives the variables. The resemblance data is internally named `p.dist`, thus formula is an expression of the form `p.dist ~ model` (see Examples). The significance is assess using overall F value, see more in `adonis`.

**FUN.ADONIS2.global and FUN.ADONIS2.margin**

Multivariate analysis of variance that can be used in matrix P analysis. The arguments `method.p` and `sqrt.p` are specified for determine resemblance index between communities based on P matrix. The argument `formula` is specified, where the left hand side gives the resemblance data, right hand side
matrix.p.sig

gives the variables. The resemblance data is internally named \( p_{dist} \), thus formula is an expression of the form \( p_{dist} \sim \text{model} \) (see Examples). The significance is assessed using F value and the difference between function is due to the argument \( \text{by} \) in \texttt{adonis2}. The function \texttt{FUN.ADONIS2.global} use as default \( \text{by} = \text{NULL} \) to assess the overall significance of all terms together whereas the function \texttt{FUN.ADONIS2.margin} use as default \( \text{by} = \text{margin} \) to assess the marginal effects of the terms and return F and p value for each term. See more in \texttt{adonis2}.

The function \texttt{adonis2} evaluate the formula argument in the global environment, however CRAN do not allow assignments to the global environment. As a temporary workaround, copy and run the lines below to make the functions \texttt{FUN.ADONIS2.global} and \texttt{FUN.ADONIS2.margin} available.

\begin{verbatim}
FUN.ADONIS2.global <- function(x, envir, method.p, formula, sqrt.p = TRUE, return.model = FALSE){
p.dist <- vegdist(x, method = method.p)
if(sqrt.p){
p.dist <- sqrt(p.dist)
}
assign("p.dist", p.dist, envir = globalenv())
mod.obs <- adonis2(formula, data = data.frame(envir), permutations = 0, by = NULL, parallel = NULL)
rm(p.dist, envir = globalenv())
statistic.obs <- mod.obs$F[1]
if(return.model){
  res <- list()
  res$mod.obs <- mod.obs
  res$statistic.obs <- statistic.obs
} else{
  res <- statistic.obs
}
return(res)
}

FUN.ADONIS2.margin <- function(x, envir, method.p, formula, sqrt.p = TRUE, return.model = FALSE){
p.dist <- vegdist(x, method = method.p)
if(sqrt.p){
p.dist <- sqrt(p.dist)
}
assign("p.dist", p.dist, envir = globalenv())
mod.obs <- adonis2(formula, data = data.frame(envir), permutations = 2, by = "margin", parallel = NULL)
rm(p.dist, envir = globalenv())
nf <- length(mod.obs$F)-2
statistic.obs <- mod.obs$F[seq_len(nf)]
if(return.model){
  res <- list()
  res$mod.obs <- mod.obs
  res$statistic.obs <- statistic.obs
} else{
  res <- statistic.obs
}
return(res)
}\end{verbatim}
FUN.GLM
Generalized linear models that can be used in PCPS analysis. The argument `formula` is specified, where the left hand side gives the PCPS used, right hand side gives the variables. The PCPS are internally named sequentially `pcps.1`, `pcps.2`, `pcps.3` and so on. Thus, formula is an expression of the form `pcps.1 ~ model` (see Examples). The type of environmental variables are extracted directly from `envir` argument, thus variables of class `factor` can be already specified in `envir data.frame` or through `formula` argument. The significance is assess using overall F value, see more in `glm`.

FUN.RDA
Redundancy analysis that can be used in PCPS analysis. The RDA analysis is performed using all PCPS specified with choices argument and all environmental variables specified by `envir` argument. The significance is assess using overall F value, see more in `rda`.

FUN.GLS.marginal and FUN.GLS.sequential
Linear model using generalized least squares that can be used in PCPS analysis. The argument `formula` is specified, where the left hand side gives the PCPS used, right hand side gives the variables. The PCPS are internally named sequentially `pcps.1`, `pcps.2`, `pcps.3` and so on. Thus, formula is an expression of the form `pcps.1 ~ model` (see Examples). The type of environmental variables are extracted directly from `envir` argument, thus variables of class `factor` can be already specified in `envir data.frame` or through `formula` argument. The significance is assess using F value and the difference between function is due to the argument `type` in `anova.gls`. The function `FUN.GLS.marginal` use as default `type = marginal` to assess the marginal significance of all terms whereas the function `FUN.GLS.sequential` use as default `type = sequential` to assess the sequential effects of the terms. Those functions return all F values calculed by `anova.gls`, including the intercept if it is in the model. Additional arguments as `correlation` can be passed by `...` argument. See more in `gls` and `anova.gls`.

FUN.LME.marginal and FUN.LME.sequential
Linear mixed-effects models that can be used in PCPS analysis. The argument `formula` is specified, where the left hand side gives the PCPS used, right hand side gives the variables. The PCPS are internally named sequentially `pcps.1`, `pcps.2`, `pcps.3` and so on. Thus, formula is an expression of the form `pcps.1 ~ model` (see Examples). The type of environmental variables are extracted directly from `envir` argument, thus variables of class `factor` can be already specified in `envir data.frame` or through `formula` argument. The significance is assess using F value and the difference between function is due to the argument `type` in `anova.lme`. The function `FUN.LME.marginal` use as default `type = marginal` to assess the marginal significance of all terms whereas the function `FUN.LME.sequential` use as default `type = sequential` to assess the sequential effects of the terms. Those function return all F values calculed by `anova.lme`, including the intercept if it is in the model. Additional arguments as `correlation` and `random` can be passed by `...` argument. See more in `lme` and `anova.lme`.

Additional function
The functions `matrix.p.sig` and `pcps.sig` only perform permutation following null models and apply the functions in all permuted matrices. Additional functions can be easy specify and passed via `FUN` argument. A skeleton of this function is slowed below. In this function the argument `x` will be always the matrix P or one matrix with PCPS choose, when additional arguments as `envir` will specify statistical analysis performed in matrix P ou PCPS. This function must return the observed
statistical in addition the \textit{return.model} argument must not be specified because it specify the return options used for observed and null statistics.

\begin{verbatim}
FUN.X <- function(x, envir, ..., return.model = FALSE){
  mod.obs <- # Function to perform analysis using x, envir and any additional argument
  statistic.obs <- # Extract only the numeric values of observed statistical
  # Next lines are mandatory
  if(return.model){
    res <- list()
    res$mod.obs <- mod.obs
    res$statistic.obs <- statistic.obs
  } else{
    res <- statistic.obs
  }
  return(res)
}
\end{verbatim}

\textbf{Value}

\begin{itemize}
  \item \textbf{call} \hspace{1cm} The arguments used.
  \item \textbf{P.obs} \hspace{1cm} Phylogeny-weighted species composition matrix.
  \item \textbf{PCPS.obs} \hspace{1cm} The principal coordinates of phylogenetic structure (PCPS)
  \item \textbf{model} \hspace{1cm} The observed model returned by FUN, an object of class glm, gls, lme, rda, adonis, adonis2 or mantel to predefined function.
  \item \textbf{fun} \hspace{1cm} The function used.
  \item \textbf{statistic.null.site} \hspace{1cm} A matrix with null statistic for site shuffle null model.
  \item \textbf{statistic.null.taxa} \hspace{1cm} A matrix with null statistic for taxa shuffle null model.
  \item \textbf{obs.statistic} \hspace{1cm} Observed statistic, F value or r value to predefined function.
  \item \textbf{p.site.shuffle} \hspace{1cm} The p value for the site shuffle null model.
  \item \textbf{p.taxa.shuffle} \hspace{1cm} The p value for the taxa shuffle null model.
\end{itemize}

\textbf{Note}

\emph{IMPORTANT}: The sequence of species in the community data matrix MUST be the same as that in the phylogenetic distance matrix and, similarly, the sequence of communities in the community data matrix MUST be the same as that in the environmental data. See details and \texttt{organize.pcps}.

\textbf{Author(s)}

Vanderlei Julio Debastiani \texttt{<vanderleidebastiani@yahoo.com.br>}

\textbf{References}


See Also

matrix.p, pcps, procrustes, glm, rda, adonis, adonis2, mantel

Examples

## Not run:
data(flona)

# MANTEL
res <- matrix.p(sig(flona$community, flona$phylo, FUN = FUN.MANTEL, method.p = "bray", 
                   method.envir = "euclidean", envir = flona$environment[, 2, drop = FALSE], runs = 99))
res

# ADONIS
res <- matrix.p(sig(flona$community, flona$phylo, FUN = FUN.ADONIS, method.p = "bray", 
                   formula = p.dist~temp, envir = flona$environment[, 2, drop = FALSE], runs = 99))
res

# ADONIS2
res <- matrix.p(sig(flona$community, flona$phylo, FUN = FUN.ADONIS2.global, 
                   envir = flona$environment, formula = p.dist-temp+alt, 
                   method.p = "bray", runs = 99))
res
res <- matrix.p(sig(flona$community, flona$phylo, FUN = FUN.ADONIS2.margin, 
                   envir = flona$environment, formula = p.dist-temp+alt, 
                   method.p = "bray", runs = 99))
res

# GLM
res <- pcps.sig(flona$community, flona$phylo, FUN = FUN.GLM, method = "bray", 
                 formula = pcps.1~temp, envir = flona$environment, choices = 1, runs = 99)
res
summary.lm(res$model)

# RDA
res <- pcps.sig(flona$community, flona$phylo, FUN = FUN.RDA, envir = flona$environment, 
                 choices = 1:2, runs = 99)
res

# GLS
res <- pcps.sig(flona$community, flona$phylo, FUN = FUN.GLS.marginal, 
                 formula = pcps.1~temp, envir = flona$environment, choices = 1, runs = 99)
res
anova(res$model, type = "marginal")
res <- pcps.sig(flona$community, flona$phylo, FUN = FUN.GLS.marginal, 
                 formula = pcps.1~temp, envir = flona$environment, 
                 correlation = nlme::corCAR1(form = ~1:39), choices = 1, runs = 99)
res
anova(res$model, type = "marginal")
mutate.names.matrix.p.null

mutate.names.matrix.p.null

*Internal function*

**Description**

Internal function to perform replacement names in object returned by `matrix.p.null`.

**Usage**

```r
mutate.names.matrix.p.null(x, replacement, newname)
```

**Arguments**

- `x` An object returned by `matrix.p.null`.
- `replacement` A replacement name to matched in object returned by `matrix.p.null`.
- `newname` New name to be replaced in object returned by `matrix.p.null`.

organize.pcps

*Function for organize data for Package PCPS*

**Description**

Package `PCPS` requires that the species and community sequence in the data.frame or matrix must be the same for all data.frame/matrices. This function use the function `organize.syncsa` to organize the data.

**Usage**

```r
organize.pcps(comm, phylodist = NULL, envir = NULL, check.comm = TRUE, ...)
```
Arguments

comm Community data, with species as columns and sampling units as rows.

phylodist Matrix containing phylogenetic distance between species. Must be a complete matrix (not a half diagonal matrix). This matrix can be larger than community data (more species) as long as it has at least all species that are in community data (Default phylodist = NULL).

envir Environmental variables for each community, with variables as columns and sampling units as rows (Default envir = NULL).

check.comm Logical argument (TRUE or FALSE) to remove sampling units and species with total sums equal or less than zero (Default check.comm = TRUE).

... Other parameters for the organize.syncsa function.

Details

The function, as well as organize.syncsa, organizes the data for the functions of the package PCPS, placing the matrices of community, phylogenetic distance and environmental variables in the same order.

Essentially this function is the same as function organize.syncsa. This use as reference the community data for organize all data.frame or matrices in the same order that the sampling units names and species names found in community data set. For this all data sets entered must be correctly named, with rows and columns named. The matrices phylodist and envir can be larger than community data (more species and/or more sampling units) as long as it has at least all species and/or sampling units that are in community data. The function organizes the data despite the absence of one of the data.frames or matrices, provided that the community data had been entered. Unspecified data will appear as NULL. All arguments this function will be passed to organize.syncsa, see more details in organize.syncsa.

Value

A object of class metacommunity.data (also of the class list) with all result returned by organize.syncsa. Featured for:

call The arguments used.

community Community data.

phylodist Phylogenetic distance.

environmental Environmental variables.

Author(s)

Vanderlei Julio Debastiani <vanderleidebastiani@yahoo.com.br>

See Also

organize.syncsa
**Examples**

```r
data(ADRS)
organize.pcps(ADRS$community, phylodist = ADRS$phylo)
```

---

**pcoa.sig**

*Significant dimensions in principal coordinate analysis*

**Description**

Function for determine the number of significant dimensions in principal coordinate analysis (PCoA).

**Usage**

```r
pcoa.sig(
  data,
  method = "gower",
  squareroot = FALSE,
  axis = 6,
  n.start = NULL,
  by = 1,
  iterations = 1000,
  parallel = NULL
)
```

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

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

## S3 method for class 'pcoasig'
summary(object, choices = c(1, 2), ...)

**Arguments**

- **data** Community data matrix.
- **method** Method for dissimilarity index, as accepted by `vegdist` (Default method = "gower").
- **squareroot** Logical argument (TRUE or FALSE) to specify if use square root of dissimilarity index (Default squareroot = FALSE).
- **axis** Maximum number of ordination principal axes to be monitored (Default axis = 6).
- **n.start** Initial sample size. If n.start = NULL initial sample size is equal to total sample size (Default n.start = NULL).
- **by** Sampling unit is added at each sampling step (Default by = 1).
iterations
Number of permutations to assess significance (Default iterations = 1000).

parallel
Number of parallel processes or a predefined socket cluster done with parallel package. Tip: use detectCores() (Default parallel = NULL).

x
An object of class pcoasig.

... Other parameters for the respective functions.

object
An object of class pcoasig.

choices
Axes for re-scaling. Choices must have length equal to two (Default choices = c(1, 2)).

Details
At each iteration step a bootstrap sample is subjected to PCoA ordination, the scores are submitted to a procrustean adjustment, and the correlation between observed and bootstrap ordination scores is computed. It compares such correlations to the same parameter generated in a parallel bootstrapped ordination of randomly permuted data. The number of axes in bootstrap or null PCoA with eigenvectors corresponding to positive eigenvalues may be smaller than the number of axes monitored, in this case, axes with values equal to 0 are created. The number of iterations with original values for each axis is shown in n.permut.bootstrap and n.permut.null.

The function scores.pcoasig re-scales the correlation values for biplot graphics.

Value
value
The eigenvalues, relative eigenvalues and cumulative relative eigenvalues.

vectors
The principal coordinates.

correlations
Correlations between axis and original data.

mean.cor.null
Mean correlations, for axis, between null and reference scores.

mean.cor.bootstrap
Mean correlations, for axis, between bootstrap and reference scores.

n.permut.bootstrap
Number of iterations for each axis in bootstrap step.

n.permut.null
Number of iterations for each axis in null step.

probabilities
Probabilities for each axis.

Note
Principal Component Analysis (PCA)
You can use the same function to determine the number of significant dimensions in principal component analysis (PCA). For this, standardize each variable for zero mean and univariate (function decostand and method standardize) and use euclidean distance as dissimilarity index.

Interpretation
If the higher dimension is significant, then all lower dimensions will also be significant.

Author(s)
Vanderlei Julio Debastiani <vanderleidebastiani@yahoo.com.br>
References


See Also

pcoa, procrustes

Examples

```r
## Not run:
data(flona)
res<-pcoa.sig(flona$community, method = "bray", squareroot = TRUE, axis = 6, iterations = 100)
res
summary(res)$scores

## End(Not run)
```

---

**pcps**

*Principal Coordinates of Phylogenetic Structure*

**Description**

Function to generate Principal Coordinates of Phylogenetic Structure (PCPS).

**Usage**

```r
pcps(
  comm,
  phylodist,
  checkdata = TRUE,
  method = "bray",
  squareroot = TRUE,
  correlations = TRUE
)
```

```r
## S3 method for class 'pcps'
plot(
  x,
  groups = NULL,
  choices = c(1, 2),
  display = "text",
  showlabel = TRUE,
  ...
)
```
Arguments

comm  Community data, with species as columns and sampling units as rows. This matrix can contain either presence/absence or abundance data. Alternatively comm can be an object of class metacommunity.data, an alternative way to set all data.frames/matrices. When you use the class metacommunity.data the argument phylodist must not be specified. See details.

phylodist  Matrix containing phylogenetic distances between species.

checkdata  Logical argument (TRUE or FALSE) to check if species sequence in the community data follows the same order as the one in the phylodist matrix (Default checkdata = TRUE).

method  Dissimilarity index, as accepted by vegdist (Default dist="bray").

squareroot  Logical argument (TRUE or FALSE) to specify if use square root of dissimilarity index (Default squareroot = TRUE).

correlations  Logical argument (TRUE or FALSE) to specify if are calculed the correlations between each PCPS and each species in matrix P (Default correlations = TRUE).

x  An object of class pcps.

groups  Factor giving the groups (Clades) for each species (Default groups = NULL).

choices  Axes for re-scaling. Choices must have length equal to two (Default choices = c(1, 2)).

display  Display text or points for the sampling units, partial match to "text" or "points" (Default display = "text").

showlabel  Label the groups by their names in the centroid of the object.

...  Other parameters for the respective functions.

Details

The function obtains a matrix containing phylogeny-weighted species composition (matrix.p) and is submitted to principal coordinates analysis (PCoA). This method generates the principal coordinates of phylogenetic structure (PCPS) (Duarte, 2011).

The sequence species show up in the community data matrix must be the same as they show up in the phylogenetic distance matrix. The function organize.pcps organizes the data, placing the matrices of community and phylogenetic distance in the same order. The use of organize.pcps
is not required for run this function, but is recommended. In this way the arguments comm and phylodist can be specified them as normal arguments or by passing them with the object returned by the function `organize.pcps` using, in this case only the argument comm. Using the object returned by organize.pcps, the comm argument is used as an alternative way of entering to set all data.frames/matrices, and therefore the phylodist argument must not be specified.

The function summary or the function scores.pcps re-scales the correlation values for obtain the scores for biplot graphics. The function plot draws a simple biplot and represent clades as "spider" graphs (see `ordispider`).

**Value**

- `P` Phylogeny-weighted species composition matrix.
- `values` The eigenvalues, relative eigenvalues and cumulative relative eigenvalues.
- `vectors` The principal coordinates of phylogenetic structure (PCPS).
- `correlations` Correlations between a PCPS axis and phylogenetically weighted species abundances or frequencies.
- `scores` Scores for biplot graphics.

**Note**

**IMPORTANT**: The sequence species show up in the community data matrix MUST be the same as they show up in the phylogenetic distance matrix. See details and `organize.pcps`.

**Author(s)**

Vanderlei Julio Debastiani <vanderleidebastiani@yahoo.com.br>

**References**


**See Also**

- `matrix.p`, `wcmdscale`, `ordispider`, `ordilabel`

**Examples**

```r
data(ADRS)
res<-pcps(ADRS$community, ADRS$phylo)
res
summary(res)
summary(res, choices = c(1, 2))$scores
plot(res, display = "text", groups = c(rep("Clade-A", 2), rep("Clade-B", 4)))
```
Description

The function estimate the phylogenetic signal at metacommunity level and draws a representation curve.

Usage

```r
pcps.curve(
  comm,
  phylodist,
  trait,
  checkdata = TRUE,
  method = "bray",
  squareroot = TRUE,
  ranks = TRUE,
  null.model.ts = FALSE,
  null.model.bm = FALSE,
  tree,
  runs = 99,
  progresbar = FALSE,
  parallel = NULL
)
```

```r
pcpc.curve.calc(values, vectors, mt)
```

## S3 method for class 'pcpscurve'
plot(
  x,
  draw.model = c("none", "ts", "bm"),
  type = "b",
  probs = c(0.025, 0.975),
  col = "black",
  model.col = "black",
  ...
)

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

## S3 method for class 'pcpscurve'
summary(object, probs = c(0.025, 0.975), ...)
Arguments

comm  Community data, with species as columns and sampling units as rows. This matrix can contain either presence/absence or abundance data. Alternatively comm can be an object of class metacommunity.data, an alternative way to set all data.frames/matrices. When you use the class metacommunity.data the arguments trait and phylodist must not be specified. See details.

phylodist  Matrix containing phylogenetic distances between species.

trait  Matrix data of species described by traits, with traits as columns and species as rows.

checkdata  Logical argument (TRUE or FALSE) to check if species sequence in the community data follows the same order as the one in the trait and in the phylodist matrices (Default checkdata = TRUE).

method  Dissimilarity index, as accepted by \textit{vegdist} (Default dist = "bray").
squareroot  Logical argument (TRUE or FALSE) to specify if use square root of dissimilarity index (Default squareroot = TRUE).
ranks  Logical argument (TRUE or FALSE) to specify if ordinal variables are convert to ranks (Default ranks = TRUE).
null.model.ts  Logical argument (TRUE or FALSE) to specify if use null model that shuffles terminal tips across the phylogenetic tree to generate null curves. See details (Default null.model.ts = FALSE).
null.model.bm  Logical argument (TRUE or FALSE) to specify if use null model that simulate trait evolving under Brownian motion to generate null curves. See details (Default null.model.bm = FALSE).
tree  Phylogenetic tree, as phylo object.
runs  Number of randomizations.
progressbar  Logical argument (TRUE or FALSE) to specify if display a progress bar on the R console (Default progressbar = FALSE).
parallel  Number of parallel processes or a predefined socket cluster done with parallel package. Tip: use detectCores() (Default parallel = NULL).
values  The eigenvalues, relative eigenvalues and cumulative relative eigenvalues returned by \textit{pcps}.
vectors  The principal coordinates of phylogenetic structure returned by \textit{pcps}.
mt  Matrix containing trait average at community level for one trait.
x  An object of class pcpscurve.
draw.model  Type of null model to draw; none (none), taxa shuffle (ts), browian motion model (bm).
type  Type of the plot to be drawn (Default type = "b").
probs  Numeric vector of probabilities used by \textit{quantile}. (Default probs = c(0.025, 0.975)).
col  Plot color.
model.col  Color of lines of null models.
...  Further graphical parameters for points.
object  An object of class pcpscurve.
Details

The sequence species show up in the community data matrix must be the same as they show up in the phylogenetic distance matrix and in traits matrix. The function \texttt{organize.pcps} organizes the data, placing the matrices of community and phylogenetic distance and trait in the same order. The function use of function organize.pcps is not requeried for run the functions, but is recommended. In this way the arguments \texttt{comm} and \texttt{phylodist} can be specified them as normal arguments or by passing them with the object returned by the function \texttt{organize.pcps} using, in this case only the argument \texttt{comm}. Using the object returned by organize.pcps, the \texttt{comm} argument is used as an alternative way of entering to set all data.frames/matrices, and therefore the arguments \texttt{phylodist} and trait must not be specified.

The PCPS are used, in a sequential manner, as predictors in a linear regression to model the trait averages across the metacommunity. The curve is drawn as the percentage of cumulative eigenvalues in the abscissa and as the determination coefficient of regressions in the ordinate.

Two null models are available. The first one (ts), the null curves are generated shuffling terminal tips across the phylogenetic tree, generates a set of random PCPS and recalculates the curves. The second (bm), the null curves are generated with simulate traits evolving under Brownian motion model.

Value

- \texttt{curve.obs}: The cumulative PCPS eigenvalues and the coefficient of determination.
- \texttt{curve.null.ts}: The cumulative PCPS eigenvalues and the coefficient of determination for each randomization using the taxa shuffle null model.
- \texttt{curve.null.bm}: The cumulative PCPS eigenvalues and the coefficient of determination for each randomization using the Brownian motion null model.

Note

**IMPORTANT**: The sequence of species in the community data matrix MUST be the same as that in the phylogenetic distance matrix and in traits matrix. See details and \texttt{organize.pcps}.

Author(s)

Vanderlei Julio Debastiani <vanderleidebastiani@yahoo.com.br>

References


See Also

\texttt{matrix.p,pcps}
select.pcpsmethod

Examples

```r
## Not run:
data(flona)
res<-pcps.curve(flona$community, flona$phylo, flona$trait[,1,drop = FALSE],
             null.model.ts = TRUE, runs = 9)
res
summary(res)
plot(res, draw.model = "ts", type = "b", col = "red")
## End(Not run)
```

---

**select.pcpsmethod**  
*Internal function*

**Description**

Internal function to select a predefined method/function available in this package.

**Usage**

```r
select.pcpsmethod(
  method = c("mantel", "adonis", "glm", "rda", "gls.marginal", "gls.sequential",
            "lme.marginal", "lme.sequential", "none")
)
```

**Arguments**

- `method`  
  A predefined method/function available in PCPS package, partial match to "mantel", "adonis", "glm", "rda", "gls.marginal", "gls.sequential", "lme.marginal", "lme.sequential" and "none".

---

**self.belonging**  
*Degree of self belonging of species*

**Description**

Define the degree of self belonging of species.

**Usage**

```r
self.belonging(dis, standardize = TRUE)
```
Arguments

- **dis**: Matrix containing distance between species.
- **standardize**: Logical argument (TRUE or FALSE) to specify if dis must be standardized into values into range 0 to 1 (Default standardize = TRUE).

Details

For the calculation of self-belonging of a set of species the dissimilarities between the species are transformed into similarities and used to define degrees of belonging to fuzzy sets (Pillar et al. 2009; Pillar & Duarte 2010). Every species among all species specifies a fuzzy set in relation to all other species, with a certain degree of belonging. The self-belonging of a given species i expresses its degree of belonging to the root node of the phylogenetic/functional tree, conditioned to the similarities between i and all other internal nodes connecting it to the root.

Value

The self-belonging for each species.

Author(s)

Vanderlei Julio Debastiani <vanderleidebastiani@yahoo.com.br>

References


See Also

- **belonging**

Examples

```r
data(ADRS)
self.belonging(ADRS$phylo)
```
**Description**

Internal function for organize the results of `wcmdscale` function. The function computes dissimilarity indices using the function `vegdist` and perform Principal Coordinates Analysis (PCoA) using the function `wcmdscale`. If data is of class `dist`, the function do not compute the dissimilarity indices.

**Usage**

```r
wcmdscale.org(data, method, squareroot, eig, correlations, ...)
```

**Arguments**

- `data`  
  Data matrix or dissimilarities of class `dist`.
- `method`  
  Method for dissimilarity index, as accepted by `vegdist`.
- `squareroot`  
  Logical argument (TRUE or FALSE) to specify if use square root of dissimilarity index.
- `eig`  
  Logical argument (TRUE or FALSE) to indicates if eigenvalues are returned.
- `correlations`  
  Logical argument (TRUE or FALSE) to indicates if correlations between axis and original data are returned.
- `...`  
  Other arguments passed to `wcmdscale` function.

**Value**

- `values`  
  The eigenvalues, relative eigenvalues and cumulative relative eigenvalues.
- `vectors`  
  The principal coordinates.
- `correlations`  
  Correlations between axis and original data.

**Author(s)**

Vanderlei Julio Debastiani <vanderleidebastiani@yahoo.com.br>

**See Also**

`vegdist`, `wcmdscale`
Index

*Topic PCPS
  define.clade, 3
  matrix.p.null, 4
  matrix.p.sig, 5
  organize.pcps, 13
  pcoa.sig, 15
  pcps, 17
  pcps.curve, 20
  self.belonging, 23
  wcndscale.org, 25

  adonis, 7, 8, 12
  adonis2, 7, 9, 12
  anova.gls, 10
  anova.lme, 10

  belonging, 24
  biplot, 16, 19

  check.formula, 2

  data.frame, 10
  define.clade, 3

  factor, 10
  formula, 2, 7
  FUN.ADONIS(matrix.p.sig), 5
  FUN.ADONIS2.global(matrix.p.sig), 5
  FUN.ADONIS2.margin(matrix.p.sig), 5
  FUN.GLMMatrix.p.sig, 5
  FUN.GL.S.marginal(matrix.p.sig), 5
  FUN.GL.S.sequential(matrix.p.sig), 5
  FUN.LME.marginal(matrix.p.sig), 5
  FUN.LME.sequential(matrix.p.sig), 5
  FUN.MANTEL(matrix.p.sig), 5
  FUN.RDA(matrix.p.sig), 5

  glm, 7, 10, 12
  gls, 7, 10
  lme, 7, 10

  makeNodeLabel, 3
  mantel, 7, 8, 12
  matrix.p, 5, 7, 12, 18, 19, 22
  matrix.p.null, 4, 7, 13
  matrix.p.sig, 4, 5, 5
  mutate.names.matrix.p.null, 13

  ordilabel, 19
  ordispider, 19
  organize.pcps, 8, 11, 13, 18, 19, 22
  organize.syncsa, 13, 14

  pcoa, 17
  pcoa.sig, 15
  pcpc.curve.calc(pcps.curve), 20
  pcps, 5, 7, 12, 17, 21, 22
  pcps.curve, 20
  pcps.sig, 4, 5
  pcps.sig(matrix.p.sig), 5
  plot.pcps(pcps), 17
  plot.pcpsscurve(pcps.curve), 20
  print.pcoasig(pcoa.sig), 15
  print.pcps(pcps), 17
  print.pcpsscurve(pcps.curve), 20
  print.pcppssig(matrix.p.sig), 5
  print.summarypcosig(pcoa.sig), 15
  print.summarypcps(pcps), 17
  procrustes, 8, 12, 17

  quantile, 21

  rda, 7, 10, 12

  scores.pcps(pcps), 17
  select.pcpsmethod, 23
  self.belonging, 23
  summary.pcoasig(pcoa.sig), 15
  summary.pcps(pcps), 17
  summary.pcppssig(pcps.curve), 20

  vegdist, 4, 7, 15, 18, 21, 25
INDEX

wcmdscale, 19, 25
wcmdscale.org, 25