Package 'adiv'

February 19, 2024

Type Package

Title Analysis of Diversity

Version 2.2.1

Date 2024-02-05

Author Sandrine Pavoine

Maintainer Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

Description Functions, data sets and examples for the calculation of various indices of biodiversity including species, functional and phylogenetic diversity. Part of the indices are expressed in terms of equivalent numbers of species. The package also provides ways to partition biodiversity across spatial or temporal scales (alpha, beta, gamma diversities). In addition to the quantification of biodiversity, ordination approaches are available which rely on diversity indices and allow the detailed identification of species, functional or phylogenetic differences between communities.

Depends R (>= 4.0)

Imports ade4, adegraphics, ape, cluster, graphics, grDevices, lpSolve, methods, phylobase, phytools, rgl, stats, utils

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2024-02-19 13:30:02 UTC

R topics documented:

Index **140**

---

adiv-package   *Analysis of Diversity*

Description

Package adiv focuses on the analysis of biodiversity

Details

Package adiv is dedicated to biodiversity indices used in ecology and conservation biology (see e.g. functions divparam, optimEH, QE). It also treats the concepts of species' originality (e.g. distinctDis, distinctTree, distinctUltra) and of dissimilarities between species and communities (e.g. dissABC, dissRicotta, dsimcom). Focus is placed on species, functional and phylogenetic diversity and on the apportionment of biodiversity across spatial and temporal scales (see for notably functions EqRao, EqRS, EqRSintra). Ordination methods are also available to depict in detail biodiversity patterns in space and/or time (e.g. crossdpcoa, dspca, evoCA, evoNSCA, evopca).

Author(s)

Sandrine Pavoine

Maintainer: Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

---

abgdivparam   *Apportionment of Parametric Indices of Diversity*

Description

Function abgdivparam calculates alpha, beta and gamma components of species diversity using parametric indices derived from Tsallis (HCDT) and Hill compositional indices. Alpha is for within-community diversity, beta for between-community diversity and gamma for the diversity of all combined communities.

Usage

```
abgdivparam(comm, w = c("speciesab", "even"),
method = c("hillCJC", "hillR", "tsallis"), q = 2,
option = c("multiplicative", "additive", "proportional",
"C", "U", "V", "S", "Renyi"), tol = 1e-08)

## S3 method for class 'abgdivparam'
plot(x, legend = TRUE,
legendposi = "topright", type = "b",
col = if (is.numeric(x)) NULL else 1:nrow(x$div),
lty = if (is.numeric(x)) NULL else rep(1, nrow(x$div)),
pch = if (is.numeric(x)) NULL else 1:nrow(x$div),
ylim1 = range(x$div[c("Alpha", "Gamma"), ]), ylim2 = NULL, ...)
```

Arguments

comm	a data frame or a matrix typically with communities as rows, species as columns and an index of abundance as entries.
w	either a numeric vector giving weights for communities (same order as in comm), or a code: one of "even" and "speciesab". If several codes are given, only the first one is used. See details.
method	a string with one of the following codes: "tsallis", "hillR", or "hillCJC". See details.
q	a vector with nonnegative value(s) for parameter q. See details.
option	a string code: either "multiplicative", "additive" or "proportional". If several codes are given, only the first one is used. Only, with method="hillCJC", other options are possible: "C", "U", "V", "S", "Renyi". See details.
tol	numeric tolerance threshold: values between -tol and tol are considered equal to zero.
x	an object of class abgdivparam obtained with function abgdivparam.
legend	a logical. If TRUE a legend is given with the colour, the type of line (etc.) used to define the diversity curve of each diversity level (gamma, alpha, beta).
legendposi	a string that gives the position of the legend to be passed to function legend of the base of R.
type	a string to be passed to the graphic argument type of functions plot and lines used to draw the diversity curve of each diversity level (gamma, alpha, beta).
col	vector of colours to be passed to the graphic argument col of functions plot and lines to define the colour of the diversity curve of each diversity level (gamma, alpha, beta, in that order).
lty	vector of types of line (plain, broken etc.) to be passed to the graphic argument lty of functions plot and lines used to draw the diversity curve of each diversity level (gamma, alpha, beta, in that order).
pch	vector of types of point (open circle, close circle, square etc.) to be passed to the graphic argument pch of functions plot and lines used to draw the diversity level (gamma, alpha, beta, in that order).

ylim1	a vector with two numerics indicating the range to be used to display alpha and gamma diversity.
ylim2	a vector with two numerics indicating the range to be used to display beta diversity.
...	other arguments can be added and passed to the functions plot and lines used to draw the graphic.

Details

Consider *j* a community (*j*=1,...,*m*), a_{jk} the abundance of species *k* in community *j*. *q* is the parameter that increases with the importance given to abundant species compared to rare species in diversity.

The methods available are: tsallis (decomposition of Tsallis or HCDT entropy (Harvda and Charvat 1967; Daroczy 1970; Tsallis 1988) into alpha, beta, gamma components):

$$^q\gamma_{Tsallis} = \left[1 - \sum_k \left(\sum_{j=1}^m w_j \frac{a_{jk}}{\sum_k a_{jk}}\right)^q\right] / (q-1)$$

$$^q\alpha_{Tsallis} = \sum_{j=1}^m w_j \left[1 - \sum_k \left(\frac{a_{jk}}{\sum_k a_{jk}}\right)^q\right] / (q-1)$$

hillR (Routledge decomposition of Hill diversity into alpha, beta, gamma components):

$$^q\gamma_{Hill} = \left[\sum_k \left(\sum_{j=1}^m w_j \frac{a_{jk}}{\sum_k a_{jk}}\right)^q\right]^{1/(1-q)}$$

$$^q\alpha_{Hill-R} = \left[\sum_{j=1}^m w_j \sum_k \left(\frac{a_{jk}}{\sum_k a_{jk}}\right)^q\right]^{1/(1-q)}$$

hillCJC (Chiu et al. (2014) decomposition of species diversity into alpha, beta, gamma components, see Supplementary material Appendix 2 in Pavoine (2016) for a justification of the formulas):

$$^q\gamma_{Hill} = \left[\sum_k \left(\sum_{j=1}^m w_j \frac{a_{jk}}{\sum_k a_{jk}}\right)^q\right]^{1/(1-q)}$$

$$^q\alpha_{Hill-CJC} = \frac{1}{m}\left[\sum_k \sum_{j=1}^m (w_j)^q \left(\frac{a_{jk}}{\sum_k a_{jk}}\right)^q\right]^{1/(1-q)}$$

Then option "additive" calculates β diversity as $\gamma - \alpha$. Option "proportional" calculates β as $(\gamma-\alpha)/\gamma$. Option "multiplicative" calculates β diversity as γ/α. Only for method="hillCJC",

options "C", "U", "V", "S", use the multiplicative option and also calculate one of the transformations introduced by Chiu et al. (2014): indices $1-C_{qm}$, $1-U_{qm}$, $1-V_{qm}$, and $1-S_{qm}$, respectively. "Renyi" calculates β diversity as $ln(\gamma/\alpha)/ln(m)$.

The weights of the sites (argument w) can be "even" (even weights), or "speciesab" (proportional to the summed abundances of all species).

Value

If only one value of q is given, abgdivparam returns a vector with alpha, beta, and gamma diversities. If more than one value of q is given, it returns a list of two objects:

q   the numeric vector of values for q.

div   a data frame with alpha, beta, gamma calculated for all values of q.

Only if method="hillCJC" and option= "C", "U", "V", "S", or "Renyi", the index $1 - C_{qm}$ (for "C"), $1 - U_{qm}$ (for "U"), $1 - V_{qm}$ (for "V"), $1 - S_{qm}$ (for "S") or the Renyi transformation (see above, for "Renyi") is also provided in the div data frame under the name "transformed.beta".

The function plot.abgdivparam returns a graphic.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Chiu, C.-H., Jost, L., Chao, A. (2014) Phylogenetic beta diversity, similarity, and differentiation measures based on Hill numbers. *Ecological Monographs*, **84**, 21–44.

Daroczy, Z. (1970) Generalized information functions. *Information and Control*, **16**, 36–51.

Havrda, M., Charvat F. (1967) Quantification method of classification processes: concept of structural alpha- entropy. *Kybernetik*, **3**, 30–35

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

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

Rao, C.R. (1986) Rao's axiomatization of diversity measures. In: Kotz S, Johnson NL, editors. *Encyclopedia of Statistical Sciences*. New York: Wiley and Sons. pp. 614–617.

Routledge, R.D. (1979) Diversity indices: which ones are admissible? *Journal of Theoretical Biology*, **76**, 503–515.

See Also

divparam, abgevodivparam

Examples

```
data(batcomm)
abgdivparam(batcomm$ab)
plot(abgdivparam(batcomm$ab))
abgdivparam(batcomm$ab, q=0:4)
plot(abgdivparam(batcomm$ab, q=0:4))
```

---

abgevodivparam	*Apportionment of Parametric Indices of Phylogenetic Diversity*

Description

Function `abgevodivparam` calculates alpha, beta and gamma components of phylogenetic diversity using parametric indices derived from Tsallis (HCDT) and Hill compositional indices. Alpha is for within-community diversity, beta for between-community diversity and gamma for the diversity of all combined communities.

Usage

```
abgevodivparam(phyl, comm, w = c("evoab", "even", "speciesab"),
method = c("hillCJC", "hillR", "tsallis"), q = 2,
option = c("multiplicative", "additive", "proportional",
"C", "U", "V", "S", "Renyi"), tol = 1e-08)

## S3 method for class 'abgevodivparam'
plot(x, legend = TRUE,
legendposi = "topright", type = "b",
col = if (is.numeric(x)) NULL else 1:nrow(x$div),
lty = if (is.numeric(x)) NULL else rep(1, nrow(x$div)),
pch = if (is.numeric(x)) NULL else 1:nrow(x$div),
ylim1 = range(x$div[c("Alpha", "Gamma"), ]), ylim2 = NULL, ...)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase), or hclust.
comm	a data frame or a matrix typically with communities as rows, species as columns and an index of abundance as entries. Species should be labeled as in the phylogenetic tree where they are the tips.
w	either a numeric vector giving weights for communities (same order as in comm), or a code: one of "even", "evoab", and "speciesab". If several codes are given, only the first one is used. See details.
method	a string with one of the following codes: "tsallis", "hillR", or "hillCJC". See details.
q	a vector with nonnegative value(s) for parameter q. See details.

option	a string code: either "multiplicative", "additive" or "proportional". If several codes are given, only the first one is used. Only, with method="hillCJC", other options are possible: "C", "U", "V", "S", "Renyi". See details.
tol	numeric tolerance threshold: values between -tol and tol are considered equal to zero.
x	an object of class abgevodivparam obtained with function abgevodivparam.
legend	a logical. If TRUE a legend is given with the colour, the type of line (etc.) used to define the diversity curve of each diversity level (gamma, alpha, beta).
legendposi	a string that gives the position of the legend to be passed to function legend of the base of R.
type	a string to be passed to the graphic argument type of functions plot and lines used to draw the diversity curve of each diversity level (gamma, alpha, beta).
col	vector of colours to be passed to the graphic argument col of functions plot and lines to define the colour of the diversity curve of each diversity level (gamma, alpha, beta, in that order).
lty	vector of types of line (plain, broken etc.) to be passed to the graphic argument lty of functions plot and lines used to draw the diversity curve of each diversity level (gamma, alpha, beta, in that order).
pch	vector of types of point (open circle, close circle, square etc.) to be passed to the graphic argument pch of functions plot and lines used to draw the diversity level (gamma, alpha, beta, in that order).
ylim1	a vector with two numerics indicating the range to be used to display alpha and gamma diversity.
ylim2	a vector with two numerics indicating the range to be used to display beta diversity.
...	other arguments can be added and passed to the functions plot and lines used to draw the graphic.

Details

Consider a phylogenetic tree T, b_T the set of branches in T, k a branch, L_k the length of branch k, j a community (j=1,...,m), a_{jk} the abundance associated with branch k in community j (sum of abundance of all species descending from the branch). q is the parameter that increases with the importance given to abundant species compared to rare species in diversity.

The methods available are: tsallis (decomposition of Tsallis or HCDT entropy (Harvda and Charvat 1967; Daroczy 1970; Tsallis 1988) into alpha, beta, gamma components adapted here to phylogenetic diversity):

$$^q\gamma_{evoTsallis} = \left[1 - \sum_{k \in b_T} L_k \left(\sum_{j=1}^{m} w_j \frac{a_{jk}}{\sum_{k \in b_T} L_k a_{jk}} \right)^q \right] / (q-1)$$

$$^q\alpha_{evoTsallis} = \sum_{j=1}^{m} w_j \left[1 - \sum_{k \in b_T} L_k \left(\frac{a_{jk}}{\sum_{k \in b_T} L_k a_{jk}} \right)^q \right] / (q-1)$$

hillR (Routledge decomposition of Hill diversity into alpha, beta, gamma components adapted hete to phylogenetic diversity):

$$^q\gamma_{evoHill} = \left[\sum_{k \in b_T} L_k \left(\sum_{j=1}^{m} w_j \frac{a_{jk}}{\sum_{k \in b_T} L_k a_{jk}} \right)^q \right]^{1/(1-q)}$$

$$^q\alpha_{evoHill-R} = \left[\sum_{j=1}^{m} w_j \sum_{k \in b_T} L_k \left(\frac{a_{jk}}{\sum_{k \in b_T} L_k a_{jk}} \right)^q \right]^{1/(1-q)}$$

hillCJC (Chiu et al. (2014) decomposition of phylogenetic diversity into alpha, beta, gamma components, see Supplementary material Appendix 2 in Pavoine (2016) for a justification of the formulas):

$$^q\gamma_{evoHill} = \left[\sum_{k \in b_T} L_k \left(\sum_{j=1}^{m} w_j \frac{a_{jk}}{\sum_{k \in b_T} L_k a_{jk}} \right)^q \right]^{1/(1-q)}$$

$$^q\alpha_{evoHill-CJC} = \frac{1}{m} \left[\sum_{k \in b_T} L_k \sum_{j=1}^{m} (w_j)^q \left(\frac{a_{jk}}{\sum_{k \in b_T} L_k a_{jk}} \right)^q \right]^{1/(1-q)}$$

Then option "additive" calculates β diversity as $\gamma - \alpha$. Option "proportional" calculates β as $(\gamma - \alpha)/\gamma$. Option "multiplicative" calculates β diversity as γ/α. Only for method="hillCJC", options "C", "U", "V", "S", use the multiplicative option and also calculate one of the transformations introduced by Chiu et al. (2014): indices $1 - C_{qm}$, $1 - U_{qm}$, $1 - V_{qm}$, and $1 - S_{qm}$, respectively. "Renyi" is the $^q evoD_{Renyi}$ index introduced in Pavoine (2016), see also Supplementary material Appendix 1 in Pavoine (2016).

The weights of the sites (argument w) can be "even" (even weights), "evoab" (proportional to the summed abundances of all evolutionary units), or "speciesab" (proportional to the summed abundances of all species). Note that if the phylogenetic tree is ultrametric (the distance from any species to the root is constant), then options "evoab" and "speciesab" are equivalent.

Value

If only one value of q is given, abgevodivparam returns a vector with alpha, beta, and gamma diversities. If more than one value of q is given, it returns a list of two objects:

q   the numeric vector of values for q.

div   a data frame with alpha, beta, gamma calculated for all values of q.

Only if method="hillCJC" and option= "C", "U", "V", "S", or "Renyi", the index $1 - C_{qm}$ (for "C"), $1 - U_{qm}$ (for "U"), $1 - V_{qm}$ (for "V"), $1 - S_{qm}$ (for "S") or the Renyi transformation (see above, for "Renyi" is also provided in the div data frame under the name "transformed.beta".

The function plot.abgevodivparam returns a graphic.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

The methodologies and scripts were presented in

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

using earlier work by

Daroczy, Z. (1970) Generalized information functions. *Information and Control*, **16**, 36–51.

Havrda, M., Charvat F. (1967) Quantification method of classification processes: concept of structural alpha- entropy. *Kybernetik*, **3**, 30–35

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

Routledge, R.D. (1979) Diversity indices: which ones are admissible? *Journal of Theoretical Biology*, **76**, 503–515.

Rao, C.R. (1986) Rao's axiomatization of diversity measures. In: Kotz S, Johnson NL, editors. *Encyclopedia of Statistical Sciences*. New York: Wiley and Sons. pp. 614–617.

Chiu, C.-H., Jost, L., Chao, A. (2014) Phylogenetic beta diversity, similarity, and differentiation measures based on Hill numbers. *Ecological Monographs*, **84**, 21–44.

See Also

evodiss, divparam, evodivparam

Examples

```
## Not run:
if(require(ape)){
data(batcomm)
phy <- read.tree(text=batcomm$tre)
ab <- batcomm$ab[,phy$tip.label]
abgevodivparam(phy, ab)
plot(abgevodivparam(phy, ab))
abgevodivparam(phy, ab, q=0:4)
plot(abgevodivparam(phy, ab, q=0:4))
}

## End(Not run)
```

| aptree | *Hierarchical Partitioning of Evolutionary and Ecological Patterns in the Organization of Phylogenetically-Structured Species Assemblages* |

Description

apd performs Hardy (2008)'s test for phylogenetic structure in species abundance distribution;

aptree apportions the diversity (according to index I_a by Pavoine et al. 2009) within one or several communities between evolutionary periods;

plot.aptree displays the phylogenetic tree with vertical lines at each speciation event (limits of the evolutionary periods), the first period starts at the tips and the last one ends at the root node; the phylogenetic tree is pruned retaining only the species present in at least one of the observed communities;

abgaptree provides the apportionment of alpha, beta and gamma diversities between evolutionary periods, according to index I_a by Pavoine et al. (2009);

rtestaptree performs the test of phylogenetic signal in the differences between communities at each evolutionary periods;

plot.rtestaptree displays the phylogenetic tree with vertical lines at each speciation event (limits of the evolutionary periods; see above); colours are used to highlight the periods where the dissimilarities between communities are different from that expected at random;

tecAptree provides technical information for the apportionment of diversity between evolutionary periods;

pIa calculates the index I_a by Pavoine et al. (2009) within each community.

Usage

```
apd(phyl, comm, wcom = c("even", "speciesab"), nrep = 99,
alter = "two-sided", tol = 1e-08)

aptree(phyl, comm, exponent = 2, tol = 1e-08)

## S3 method for class 'aptree'
plot(x, col.line = 'blue', ...)

abgaptree(phyl, comm, exponent = 2,
wcom = c("even", "speciesab"), tol = 1e-08)

rtestaptree(phyl, comm, nrep = 99, alter = "two-sided",
exponent = 2, wcom = c("even", "speciesab"), tol = 1e-08)

## S3 method for class 'rtestaptree'
plot(x, col.line = c("blue", "red"),
alpha = 0.05, ...)
```

```
tecAptree(phyl, v = NULL, tol = 1e-08)

pIa(phyl, comm, exponent = 2, tol = 1e-08)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase), or hclust.
comm	a data frame or a matrix typically with communities as rows, species as columns and presence/absence or an index of abundance as entries. Species should be labeled as in the phylogenetic tree where they are the tips. In function aptree, comm can be a vector that provides the presence/absence or an index of abundance within a single community.
wcom	a numeric vector that gives the weight attributed to the community. The weights must be positive and their sum equals 1.
nrep	a numeric that gives the number of permutations.
alter	a string specifying the alternative hypothesis; it must be one of "greater", "less" or "two-sided".
tol	a numeric. If the absolute value of a statistic is less than tol, this statistic is considered equal to zero.
exponent	a numeric that gives the value of parameter a in index I_a.
x	in plot.aptree, x is an object inheriting class aptree obtained with function aptree. In plot.rtestaptree, x is an object inheriting class rtestaptree obtained with function rtestaptree.
col.line	in plot.aptree, col.line is a string which attributes a colour to the vertical lines placed at each speciation event and defining the evolutionary periods. In plot.rtestaptree, it is a vector with two strings. These strings give colours to the vertical lines placed at each speciation event. The first colour is used when the differences between communities at the evolutionary period are not significantly different from random; the second colour is used when they are.
alpha	a numeric: the nominal alpha level for significancy (the p-values calculated with function rtestaptree are compared to alpha to determine the result of the test).
...	further arguments passed to or from other methods.
v	either NULL or a vector that provides the presence/absence or an index of abundance of species within a single community.

Details

The approaches developed in these functions rely on a parametric index of phylogenetic diversity named I_a. The parameter a controls the importance given to rare versus abundant species in communities. Index I_a generalizes Rao's quadratic entropy (QE) applied to phylogenetic distances between species (when a=2) and Faith's Phylogenetic Diversity index (PD) (when a=0). When a tends towards 1, the index is a generalization of the Shannon index of diversity applied to phylogenetic data in addition to abundance data. In Pavoine et al. (2009), we developed this index and demonstrated how it can be used to partition diversity simultaneously across evolutionary periods in the phylogeny and across spatial (e.g. local communities in a region) and/or time units (e.g. a community investigated yearly).

Value

The function apd returns an object of class randtest with the results of the test (see function randtest in package ade4).

The function aptree returns a data frame with the evolutionary periods as rows, the communities as columns and the diversity values as entries.

The function plot.aptree returns a graph.

The function abgaptree returns a data frame with the evolutionary periods as rows, alpha diversity, beta diversity and gamma diversity as columns and the diversity values as entries.

The function rtestaptree returns an object of class krandtest with the results of the permutation tests. (see function krandtest in package ade4)

The function plot.rtestaptree returns a graph.

The function tecAptree returns a list. If v is NULL, the values of the list are:

h	the height at which each evolutionary period ends;
plength	period length;
ngroups	number of monophyletic groups per evolutionary period;
list	list of the species per monophyletic group at each evolutionary period;
call	original call.

If v contains a vector of presence/absence or abundance, the following object is added in the output:

relab	the relative abundance (sum of species' presences or abundances depending on v) of each monophyletic group at each evolutionary period.

The function pIa returns a data frame with communities as rows and only one column. This column gives, for each community, the value taken by index I_a of phylogenetic diversity developed by Pavoine et al. (2009).

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr> with contributions of Stephane Dray.

References

Pavoine, S., Love, M., Bonsall, M.B. (2009) Hierarchical partitioning of evolutionary and ecological patterns in the organization of phylogenetically-structured species assemblages: application to rockfish (genus: Sebastes) in the Southern California Bight. *Ecology Letters*, **12**, 898–908.

See Also

[QE](QE)

Examples

```
## Not run:
if(require(ape)){
data(rockfish)
phy <- read.tree(text=rockfish$tre)
ABG <- abgaptree(phy, rockfish$fau, wcom="speciesab")
colSums(ABG)
A <- aptree(phy, rockfish$fau)
colSums(A)
plot(A, cex=0.5)
P <- pIa(phy, rockfish$fau)
P
T <- apd(phy, rockfish$fau)
plot(T)
#R <- rtestaptree(phy, rockfish$fau, nrep=999, wcom="speciesab")
#plot(R)
TA <- tecAptree(phy)
TA$h
}

## End(Not run)
```

---

batcomm   *Bat Abundance and Phylogeny Along a Disturbance Gradient in a Neotropical Rainforest*

Description

The data were collected by Medellin et al. (2000) on bats in four habitats in the Selva Lacandona of Chiapas, Mexico. Two phylogenetic trees were obtained, one with Fritz et al. (2009) phylogeny pruned for retaining only the species present in Medellin et al. data set; the other one using a consensus tree obtained from 9999 credible trees developed by Upham et al. (2019) and obtained from vertlife.org (see Pavoine and Ricotta (2021) for more details).

Usage

```
data("batcomm")
```

Format

`batcomm` is a list of two components:

ab, a data frame with habitats as rows, species as columns and abundance of species in habitats as entries.

tre, a phylogenetic tree in newick format for all bat species in ab.

ab2, a data frame with habitats as rows, species as columns and abundance of species in habitats as entries.

tre2, a phylogenetic tree in newick format for all bat species in ab.

Details

In `ab` and `ab2`, four habitat types were analyzed: F=rainforest; P=cacao plantations; O=old fields; C=cornfields (Pavoine 2016).

`ab` and `tre` contain species names as in Medellin et al. (2000). `ab2` and `tre2` contain species names as in Upham et al. (2019).

Synonyms used are: Pteronotus_parnelii / Pteronotus_parnellii; Phyllostomus_stenops / Phylloderma_stenops; Tonatia_brasiliense / Lophostoma_brasiliense; Tonatia_evotis / Lophostoma_evotis; Micronycteris_brachyotis / Lampronycteris_brachyotis; Vampyrodes_major / Vampyrodes_caraccioli.

`tre` was obtained using Fritz et al. (2009) and `tre2` using Upham et al. (2019). The resolution of the bat phylogeny obtained from Fritz is uncertain especially at the older nodes (Pavoine 2016).

Source

<http://www.oikosjournal.org/appendix/oik-03262>

References

Fritz, S.A., Bininda-Emonds, O.R.P., Purvis, A. (2009) Geographic variation in predictors of mammalian extinction risk: big is bad, but only in the tropics. *Ecology Letters*, **12**, 538–549.

Medellin, R., Equihua M., Amin, M.A. (2000) Bat diversity and abundance as indicators of disturbance in Neotropical rainforest. *Conservation Biology*, **14**, 1666–1675.

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

Pavoine, S., Ricotta, C. (2021) On the relationships between rarity, uniqueness, distinctiveness, originality and functional/phylogenetic diversity. *Contact authors for information on this study*

Upham, N.S., Esselstyn, J.A., Jetz, W. (2019) Inferring the mammal tree: Species-level sets of phylogenies for questions in ecology, evolution, and conservation. *PloS Biology*, **17**, e3000494.

Examples

```
## Not run:
if(require(ape)){
data(batcomm)
phy <- read.tree(text=batcomm$tre)
plot(phy)
ab <- batcomm$ab[,phy$tip.label]
plot(abgevodivparam(phy, ab, q=0:4))

phy2 <- read.tree(text=batcomm$tre2)
plot(phy2)
ab2 <- batcomm$ab2[,phy2$tip.label]
plot(abgevodivparam(phy2, ab2, q=0:4))
}

## End(Not run)
```

---

betastat   *Multiple-Site Dissimilarity Measure for Species Presence/Absence*
                                     Data

Description

Functions `betastatjac` and `betastatsor` calculate multiple-site dissimilarity (beta diversity). The first one is derived from Jaccard coefficient of similarity and the second from Sorensen coefficient. These proposed dissimilarity indices are additively partitioned into species nestedness and turnover.

Usage

```
betastatjac(comm)

betastatsor(comm)
```

Arguments

comm   a data frame typically with communities as rows, species as columns and pres-
                ence/absence (1/0) as entries.

Value

The two functions return a vector of 4 values:

beta   Ricotta and Pavoine (2015) β^+ relative measure of additive beta diversity (multiple-
                site dissimilarity);

betaT   Ricotta and Pavoine (2015) β_T contribution of species turnover to multiple-site
                dissimilarity;

betaN   Ricotta and Pavoine (2015) β_N contribution of species nestedness to multiple-
                site dissimilarity;

sim   Ricotta and Pavoine (2015) $\bar{\beta}^\times$ relative measure of multiple-site similarity.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C. and Pavoine, S. (2015) A multiple-site dissimilarity measure for species presence/absence data and its relationship with nestedness and turnover. *Ecological Indicators*, **54**, 203–206.

Examples

```
data(RP15EI)
# Scripts used in Figure 1 of Ricotta and Pavoine (2015)
betastatjac(RP15EI$M1)
betastatjac(RP15EI$M2)
betastatjac(RP15EI$M3)
betastatjac(RP15EI$M4)

#see also
betastatsor(RP15EI$M1)
betastatsor(RP15EI$M2)
betastatsor(RP15EI$M3)
betastatsor(RP15EI$M4)
```

---

betaTreeUniqueness   *Plot-to-plot functional or phylogenetic dissimilarity and uniqueness*

Description

The function betaTreeUniqueness calculates Ricotta et al. (2020) plot-to-plot functional or phylogenetic beta uniqueness (index named U_F for functional data and U_P for phylogenetic data in Ricotta et al. 2020).

Usage

```
betaTreeUniqueness(mtree, comm, height = NULL, tol = 0.001)
```

Arguments

mtree	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase), or hclust. The tree must be ultrametric: equal distance from any tip to the root.
comm	a matrix containing the relative or absolute abundance of all species in plots. Columns are species and plots are rows. Column labels (species names) should be assigned as in mtree.
height	either NULL or a numeric. See details.
tol	a tolerance threshold. A value between -tol and tol is considered as zero. See details.

Details

Object mtree defines a tree with species as tips. If argument height is NULL, then the root of the tree will be placed at the most recent common ancestor of all species occurring in the set of plots (given in object comm). An alternative position for the root can be given by specifying the height of the tree (argument height). In that case, height must be higher than the distance between tips and the most recent common ancestor of all species.

The tolerance threshold `tol` is particularly important if your tree is not exactly ultrametric due to approximation problems. In that case, the distance from tip to root varies according to the tip considered, although it should not (variations are due to approximation problems). A difference smaller than `tol` in the distance to root for two species will thus be considered as null.

Value

The function returns a matrix with the values of the functional or phylogenetic beta uniqueness for each pair of plots.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C., Laroche, F., Szeidl, L., Pavoine, S. (2020) From alpha to beta functional and phyloge-netic redundancy. *Methods in Ecology and Evolution*, **11**, 487–493.

See Also

`DP` for plot-to-plot dissimilarities, `treeUniqueness` for alpha uniqueness

Examples

```
## Not run:
if(require(ape)){
data(RutorGlacier)
phy <- read.tree(text=RutorGlacier$TreeNW)
plot(phy)

ab <- RutorGlacier$Abund[, phy$tip.label]

# Phylogenetic beta Uniqueness between plots
# (Ricotta et al. 2020)
Up <- betaTreeUniqueness(phy, ab, tol=0.00001)
}

## End(Not run)
```

---

betaUniqueness   *Algorithmic index of plot-to-plot functional (or phylogenetic) dissimi-larity and uniqueness*

Description

The function `betaUniqueness` calculates uniqueness and redundancy taking account of functional dissimilarities between species using equation 5 and 6 in Ricotta et al. (2021). Note that functional dissimilarities could be replaced by any other type of dissimilarities between species, including phylogenetic dissimilarities.

Usage

```
betaUniqueness(comm, dis, Nind = 10000)
```

Arguments

comm	a matrix containing the relative or absolute abundance of all species in plots. Columns are species and plots are rows. Column labels (species names) should be assigned as in dis.
dis	a matrix or an object of class dist providing the functional dissimilarities between species (dissimilarities are nonnegative, symmetric, and the dissimilarity between a species and itself is zero). **Species here must be in the same order as in the columns of comm**.
Nind	an integer. The algorithmic index will be applied by assuming that each plot contains Nind individuals. The highest Nind, the most precise the index value will be (see Gregorius et al. 2003, for more details).

Value

The function betaUniqueness returns a list with the following objects:

- betaUniqueness: a matrix with the values of the proposed beta uniqueness ($U\beta$=DKG/DR) for each pair of plots (Ricotta et al. (2021), eq. 6);

- betaRedundancy: a matrix with the values of the proposed beta redundancy ($R\beta$=1-DKG/DR) for each pair of plots (Ricotta et al. (2021), eq. 5);

- dissimilarityGap: a matrix with the values of the dissimilarity gap index (DR-DKG) for each pair of plots;

- DR: a matrix with the values of the species-based (Rogers) dissimilarity index (DR) for each pair of plots (Ricotta et al. (2021), eq. 4);

- DKG: a matrix with the values of the algorithmic functional dissimilarity index (DKG) for each pair of plots (Ricotta et al. (2021), eq. 3).

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C., Kosman, E., Laroche, F., Pavoine, S. (2021) Beta redundancy for functional ecology. *Methods in Ecology and Evolution*, **12**, 1062–1069. doi:10.1111/2041210X.13587

Gregorius, H.-R., Gillet, E.M., Ziehe, M. (2003) Measuring differences of trait distributions between populations. *Biometrical Journal*, **8**, 959–973. doi:10.1002/bimj.200390063

See Also

betaTreeUniqueness adapted to the use of phylogenetic trees with species as tips, dislptransport for the algorithmic functional dissimilarity index (DKG in Ricotta et al. 2021), and uniqueness for alpha uniqueness

Examples

```
## Not run:
data(RutorGlacier)
fundis <- dist(scale(RutorGlacier$Traits2[1:6]))
fundis <- fundis/max(fundis)
frameDKG <- betaUniqueness(RutorGlacier$Abund, fundis)

f1 <- unlist(sapply(1:58, function(i) rep(RutorGlacier$Fac[i], 59-i)))
f2 <- unlist(sapply(1:58, function(i) RutorGlacier$Fac[-(1:i)]))
f <- paste(f1, f2, sep="-")
F <- factor(f, levels=c("early-early", "mid-mid", "late-late", "early-mid",
    "mid-late", "early-late"))

vbetaU_A <- as.vector(as.dist(frameDKG$betaUniqueness))

boxplot(vbetaU_A~F, ylab="Beta uniqueness", xlab="Compared successional stages")
# See Ricotta et al. 2021 Electronic Appendix 3 for for details

## End(Not run)
```

---

birdData   *Avian Communities along Successional Forest Gradients*

Description

The density of bird species in 3 locations in the Mediterranean region and 2 locations in the central Europe region were collected by Blondel and Farre (1988) along a gradient of habitats. The densities and a phylogenetic tree are provided for all observed species.

Usage

```
data("birdData")
```

Format

birdData is a list of 4 objects:

fau, a data frame with communities as rows, species as columns and densities as entries. The names of the communities start with the first three letters of the location and end with the seral stage number. See details.

tre, a string. It contains the phylogeny in a newick format.

facA, a factor. It indicates which location each community belongs to.

facB, a factor. It indicates which seral stage each community belongs to.

Details

The locations are Alg = Algeria, Bur = Burgundy, Cor = Corsica Island, Pol = Poland, Pro = Provence.

Three are in the Mediterranean region (Provence, southern France; Corsica Island, southern France; and north east Algeria) and two in the central European region (Burgundy, central France; and Poland).

In each location, a habitat gradient has been conventionally divided into six seral stages (intermediate stages found in forest ecosystems advancing towards their climax stage after a disturbance event) in such a way that all five selected habitat gradients match one another reasonably well in terms of the number, patterns and overall structure of habitats: from low bushy vegetation, less than 1 m height (stage 1), to forests with trees at least 20 m high (stage 6).

The composite phylogenetic tree was obtained based on Davis' supertree: a strict consensus of 2000 trees (see details in Text S5 of Pavoine et al. 2013).

Source

Dataset S1 in Pavoine et al. (2013)

References

Blondel, J. and Farre, H. (1988) The convergent trajectories of bird communities along ecological successions in European forests. *OEcologia (Berlin)*, **75**, 83–93.

Davis, K.E. (2008) *Reweaving the tapestry: a supertree of birds* [PhD thesis]. Glasgow, U.K.: University of Glasgow.

Pavoine, S., Blondel, J., Dufour, A.-B., Gasc, A., Bonsall, M.B. (2013) A new technique for analysing interacting factors affecting biodiversity patterns: crossed-DPCoA. *PloS One*, **8**, e54530.

Examples

```
## Not run:
if(require(ape) && require(adephylo)){
data(birdData)
phy <- read.tree(text=birdData$tre)
phydis <- sqrt(distTips(phy, method="nNodes")+1)

fau <- birdData$fau[, phy$tip.label]
facA <- birdData$facA
facB <- birdData$facB

cd_mainB <- crossdpcoa_maineffect(fau, facB, facA, phydis, w=rep(1/30, 30), scannf = FALSE)
s.label(cd_mainB$l2)
}

## End(Not run)
```

Description

The function `CFprop` calculates the matrix CF of intra- (diagonal) and inter-specific (off-diagonal) similarities as defined in the main text of Pavoine and Izsak (2014), and matrix CwF as defined in Appendix S3 of Pavoine and Izsak (2014) for weighting functional attributes. The function `multiCFprop` calculates matrices CwmF1, CwmF2, CwmF3 when several functional traits are considered (Appendix S3 of Pavoine and Izsak 2014). Traits and the attributes of the traits can be weighted. These two functions consider functional traits expressed as proportion (compositional) vectors. The functions `CFbinary` and `multiCFbinary` are the equivalents of `CFprop` and `multiCFprop` when traits are expressed as binary vectors as shown in Appendix S3 of Pavoine and Izsak (2014).

Usage

```
CFbinary(df, wA = rep(1, ncol(df)))

multiCFbinary(Ktab, w.attributes = lapply(Ktab, function(x) rep(1, ncol(x))),
w.traits = rep(1/length(Ktab), length(Ktab)),
labels = rownames(Ktab[[1]]), solution = c(2, 1))

CFprop(df, wA = rep(1, ncol(df)))

multiCFprop(Ktab, w.attributes = lapply(Ktab, function(x) rep(1, ncol(x))),
w.traits = rep(1/length(Ktab), length(Ktab)),
labels = rownames(Ktab[[1]]), solution = c(2, 1))
```

Arguments

df	a data frame or a matrix with species (or any entities of interest) as rows, functional attributes as columns, and proportions (for `CFprop`) or 0/1 values (for `CFbinary`) as entries. The sum of the columns will be standardized to equal 1 in function `CFprop` but not in `CFbinary`.
wA	a vector of weights that should be given to the attributes (same order as the columns of df).
Ktab	a list of data frames, each of which represents a trait. For a given trait, the data frame should have species (or any entities of interest) as rows, functional attributes as columns, and proportions (for `multiCFprop`) or 0/1 values (for `multiCFbinary`) as entries. The sum of the columns will be standardized to equal 1 in function `multiCFprop` but not in `multiCFbinary`.
w.attributes	a list of weights that should be given to the attributes of each trait. Traits should be in the same order as they appear in the list of tables `Ktab`. The attributes of a trait should be ordered as the columns of the corresponding data frame in `Ktab`.

w.traits	a numeric vector of weights that should be given to the traits (same order as the tables of Ktab).
labels	a vector of strings that gives the names of the species (or the other entities of interest; same order as the rows of all tables in Ktab).
solution	a numeric value (either 1 or 2) that indicates which equations are used to summarize the information given by several traits among the 2 approaches given in Appendix S3 of Pavoine and Izsak (2014) page 9. If a vector is given, only the first value of the vector is considered.

Value

A matrix with nonnegative values

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S. and Izsak, J. (2014) New biodiversity measure that includes consistent interspecific and intraspecific components. *Methods in Ecology and Evolution*, **5**, 165–172.

See Also

[qHdiv](#)

---

crossdpcoa *Crossed-DPCoA*

Description

Crossed-DPCoA typically analyzes the phylogenetic or functional compositions of communities according to two factors affecting the communities (e.g. space and time; habitat and region)

The function crossdpcoa_maineffect obtains the space of DPCoA (the double principal coordinate analysis) where species are placed according to their functional traits or phylogeny and communities are placed at the center of their species. Next, levels of each factor are placed at the center of their communities. The function crossdpcoa_maineffect determines the principal axes of the positions of the levels of one of the factors in this space and projects species' points on these principal axes. The main effect of the factor named facA is analysed by this process (Pavoine et al. 2013).

The function crossdpcoa_version1 performs version 1 of the crossed DPCoA in Pavoine et al. (2013) where the effect of factor facA on the diversity of communities, given factor facB, is analysed.

The function crossdpcoa_version2 performs version 2 of the crossed DPCoA in Pavoine et al. (2013) where the effect of factor facA on the diversity of communities, given factor facB, is also analysed.

Usage

```
crossdpcoa_maineffect(df, facA, facB, dis = NULL,
scannf = TRUE, nf = 2, w = c("classic", "independence"),
tol = 1e-07)

crossdpcoa_version1(df, facA, facB, dis = NULL,
scannf = TRUE, nf = 2, w = c("classic", "independence"),
tol = 1e-07)

crossdpcoa_version2(df, facA, facB, dis = NULL,
scannf = TRUE, nf = 2, w = c("classic", "independence"),
tol = 1e-07)
```

Arguments

df	a data frame or a matrix of 0/1 or nonnegative values. As an exemple, I consider below a communities x species data frame or matrix with abundances as entries.
facA	a factor with the same length as the number of rows (communities) in df.
facB	another factor with the same length as the number of rows (communities) in df.
dis	an object of class dist that contains the distances among species (e.g. functional or phylogenetic distances). If NULL equidistances are used among species. The distances must have Euclidean properties. Distances are integrated in the Euclidean Diversity Index (Champely and Chessel 2002), which corresponds to a particular formulation of Rao (1982) quadratic entropy. For example the diversity within a community i is $EDI = \sum_{k=1}^{S} \sum_{l=1}^{S} p_{k|i} p_{k|j} \frac{d_{kl}^2}{2}$, where S is the number of species, $p_{k|i}$ is the relative abundance of species k within community i; d_{kl} is the (phylogenetic or functional) dissimilarity between species k and l.
scannf	a logical value indicating whether the screeplot (with eigenvalues) should be displayed.
nf	if scannf is FALSE, an integer indicating the number of kept axes.
w	either a string or a numeric vector of positive values that indicates how the rows of df (the communities) should be weighted. If w="classic", the weights are defined from the sum of the values in each row (e.g. sum of all species abundances within a community). If w="independence", then the weight attributed to a row of df (a community) is the product of the weight attributed to a level of factor A with the weight attributed to a level of factor B. If a vector of strings is given, only the first one is retained. If numeric, values in w must be in the same order as the rows of df (see Pavoine et al. 2013 for details on the definition of these weights).
tol	a numeric tolerance threshold: a value between -tol and tol is considered as null.

Value

The functions crossdpcoa_maineffect, crossdpcoa_version1 and crossdpcoa_version2 return a list containing the following information used for computing the crossed-DPCoA:

l1	coordinates of the columns of df (the species).
l2	coordinates of the levels of factor A.
l3	(for functions `crossdpcoa_version1` and `crossdpcoa_version2` only) coordinates of the rows of df (the communities).
eig	the eigenvalues.
lX	the weights attributed to the columns of df (species).
lA	the weights attributed to the levels of factor A.
lB	the weights attributed to the levels of factor B.
lC	the weights attributed to the rows of df (communities).
div	a numeric vector with the apportionment of Rao's quadratic diversity (APQE).
call	the `call` function.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S., Blondel, J., Dufour, A.-B., Gasc, A., Bonsall, M.B. (2013) A new technique for analysing interacting factors affecting biodiversity patterns: crossed-DPCoA. *PloS One*, **8**, e54530.

Examples

```
## Not run:
if(require(ape) && require(phylobase) && require(adephylo)
   && require(adegraphics)){
O <- adegpar()$plabels$optim
adegpar("plabels.optim" = TRUE)

data(birdData)
phy <- read.tree(text=birdData$tre)
phydis <- sqrt(distTips(phy, method="nNodes")+1)

fau <- birdData$fau[, phy$tip.label]
facA <- birdData$facA
facB <- birdData$facB

#Here factor B is put first to analyze
#the main effect of the strata:
cd_mainB <- crossdpcoa_maineffect(fau, facB, facA, phydis, w=rep(1/30, 30), scannf = FALSE)
barplot(cd_mainB$eig)
cd_mainB$eig[1:2]/sum(cd_mainB$eig)

#Positions of the levels of factor B on its principal axes:
s.label(cd_mainB$l2)
# The "d" value on graphs indicates the length of the edge of a grid cell (scale of the graphic).

#The coordinates of the species on the same axes
# can be displayed in front of the phylogeny
```

```
# (several possibilities are provided below,
# the last one use package adephylo)):
mainBl1.4d <- phylo4d(phy, as.matrix(cd_mainB$l1))
dotp4d(mainBl1.4d, center = FALSE, scale = FALSE)
barp4d(mainBl1.4d, center = FALSE, scale = FALSE)
gridp4d(mainBl1.4d, center = FALSE, scale = FALSE)
parmar <- par()$mar
par(mar=rep(.1,4))
table.phylo4d(mainBl1.4d, show.node=FALSE, symbol="squares",
    center=FALSE, scale=FALSE, cex.label=0.5, ratio.tree=0.7)
par(mar=parmar)

#If factor A is put first, the analysis focus
#on the main effect of the region:
cd_mainA <- crossdpcoa_maineffect(fau, facA, facB, phydis, w=rep(1/30, 30), scannf = FALSE)
barplot(cd_mainA$eig)
cd_mainA$eig[1:2]/sum(cd_mainA$eig)

#Positions of the levels of factor A on its principal axes:
s.label(cd_mainA$l2)
# The "d" value on graphs indicates the length of the edge of a grid cell (scale of the graphic).

#The coordinates of the species on the same axes
# can be displayed in front of the phylogeny
# (several possibilities are provided below,
# the last one use package adephylo)):
mainAl1.4d <- phylo4d(phy, as.matrix(cd_mainA$l1))
dotp4d(mainAl1.4d, center = FALSE, scale = FALSE)
barp4d(mainAl1.4d, center = FALSE, scale = FALSE)
gridp4d(mainAl1.4d, center = FALSE, scale = FALSE)
parmar <- par()$mar
par(mar=rep(.1,4))
table.phylo4d(mainAl1.4d, show.node=FALSE, symbol="squares",
   center=FALSE, scale=FALSE, cex.label=0.5, ratio.tree=0.7)
par(mar=parmar)

#Crossed DPCoA Version 1
cd_v1 <- crossdpcoa_version1(fau, facA, facB, phydis, w=rep(1/30, 30), scannf = FALSE)
#Proportion of SS(A) expressed by the two first axes:
cd_v1$eig[1:2]/sum(cd_v1$eig)
#To view the positions of the locations on the first two axes, write:
s.label(cd_v1$l2)
#To view the positions of all communities on the first two axes, write:
s.label(cd_v1$l3)
#To view the positions of the species on the first two axes in front of the phylogeny, write:
v1l1.4d <- phylo4d(phy, as.matrix(cd_v1$l1))
# (then several functions can be used as shown below,
# the last function, table.phylo4d, is from package adephylo)):
dotp4d(v1l1.4d, center = FALSE, scale = FALSE)
barp4d(v1l1.4d, center = FALSE, scale = FALSE)
gridp4d(v1l1.4d, center = FALSE, scale = FALSE)
parmar <- par()$mar
par(mar=rep(.1,4))
```

```
table.phylo4d(v1l1.4d, show.node=FALSE, symbol="squares",
   center=FALSE, scale=FALSE, cex.label=0.5, ratio.tree=0.7)
par(mar=parmar)

#Crossed DPCoA Version 2
#Crossed DPCoA version 2 can now be performed as follows:
cd_v2 <- crossdpcoa_version2(fau, facA, facB, phydis, w=rep(1/30, 30), scannf = FALSE)
#Proportion of variation among levels of factor A
#in the subspace orthogonal to the principal axes of B
#expressed by the two first axes:
cd_v2$eig[1:2]/sum(cd_v2$eig)
#To view the positions of the locations on the first two axes, write:
s.label(cd_v2$l2)
#To view the positions of all communities on the first two axes, write:
s.label(cd_v2$l3)
#To view the positions of the species on the first two axes in front of the phylogeny, write:
v2l1.4d <- phylo4d(phy, as.matrix(cd_v2$l1))
# (then several functions can be used as shown below,
# the last function, table.phylo4d, is from package adephylo)):
dotp4d(v2l1.4d, center = FALSE, scale = FALSE)
barp4d(v2l1.4d, center = FALSE, scale = FALSE)
gridp4d(v2l1.4d, center = FALSE, scale = FALSE)
parmar <- par()$mar
par(mar=rep(.1,4))
table.phylo4d(v2l1.4d, show.node=FALSE, symbol="squares",
   center=FALSE, scale=FALSE, cex.label=0.5, ratio.tree=0.7)
par(mar=parmar)

adegpar("plabels.optim" = 0)
}

## End(Not run)
```

---

dbMANOVAspecies   *Indicator species analysis in the framework of multivariate analysis of
                                variance*

Description

"Consider N plots distributed among K groups; plots are composed of species whose abundances
within each plot are known. The QT, QB and QW statistics defined in [Ricotta et al. 2021] aim
at evaluating the average difference, in terms of species identity and abundance, between any two
plots (QT), between two plots within a group (QW), and the gap between QT and QW (=QB) due
to compositional differences between groups of plots. The function dbMANOVAspecies calculates
QT, QB and QW as in [Ricotta et al. 2021] and the species-centered components QTs, QBs and
QWs. It also calculates the SES values (equation 7 [in Ricotta et al. 2021]) associated with QB
and QBs and allows tests for the significance of the SES values (H0 = the SES value is similar
as that expected by randomly permuting plots among groups of plots and H1 = the SES value

of is greater than that expected by randomly permuting plots among groups of plots). The function dbMANOVAspecies_pairwise complements function dbMANOVAspecies by performing post-hoc tests for all pairs of groups. It must be executed in the same R environment (workspace) as function dbMANOVAspecies. A third function named summary.dbMANOVAspecies_pairwise provides a short summary of the results of function dbMANOVAspecies_pairwise (with SES and P values)" (Ricotta et al. 2021, Appendix 3).

Usage

```
dbMANOVAspecies(comm, groups, nrep = 999, method = c("Euclidean", "Manhattan",
    "Canberra", "BrayCurtis"), global = TRUE, species = TRUE, padjust = "none",
    tol = 1e-8)

    dbMANOVAspecies_pairwise(dbobj, signif = TRUE, salpha = 0.05, nrep = NULL)

    ## S3 method for class 'dbMANOVAspecies_pairwise'
summary(object, DIGITS = 3, ...)
```

Arguments

comm	a matrix of N plots $\times$ S species containing the relative or absolute abundance of all species. Columns are species and plots are rows.
groups	a vector of characters or a factor with the names of the groups associated with plots. Names of groups must be listed in the same order as plots in object comm. For example, the first value of groups gives the name of the group for the first plot (first row in object comm).
nrep	a numeric that gives the number of permutations to be done. nrep can be set to NULL in function dbMANOVAspecies_pairwise, in which case the number of permutations used to built object dbobj (first argument of function dbMANOVAspecies_pairwise) is used by default.
method	a string, one of "Euclidean" (to perform the Euclidean distance), "Manhattan" (to select the Manhattan distance), "Canberra" (to perform the Canberra distance), "BrayCurtis" (to opt for the BrayCurtis distance).
global	a logical. If TRUE the global test (using the SES of QB) is performed. The global test can be performed in addition to one test per species.
species	a logical. If TRUE one test per species is performed. If species = FALSE, then argument global is ignored and the global test is performed.
padjust	a method of correction for multiple tests one of p.adjust.methods (see ?p.adjust.methods for possible choices); choose "none" if you don't need any correction for multiple tests. The argument padjust is ignored if species = FALSE.
tol	a numeric tolerance threshold. Any value between -tol and tol will be considered as equal to zero.
dbobj	an object of class "dbMANOVAspecies" obtained with function dbMANOVAspecies.
signif	a logical. If TRUE, the post-hoc tests are performed only for species that were associated with significant tests in object dbobj.

salpha   a numeric. The level of significance (nominal alpha error) for P values (must be between 0 and 1). Ignored if `signif = FALSE`.

object   an object of class `"dbMANOVAspecies_pairwise"` obtained with function `dbMANOVAspecies_pairwise`.

DIGITS   integer indicating the number of decimal places to retain when displaying the results. If NULL, all decimals are retained.

...   further arguments passed to or from other methods.

Value

`dbMANOVAspecies` returns a list with the following objects:

-observations: a vector or a data frame with the observed values of statistics (QT, QB and QW) (if global = TRUE) and/or the contribution each species has to these statistics (if species = TRUE);

`-test`: an object of class randtest or krandtest with the results of the test for the global differences, in terms of species composition, between groups of plots (if global = TRUE) and/or for the contribution each species has in these differences (if species = TRUE). `dbMANOVAspecies_pairwise` provides in a list the same objects as function `dbMANOVAspecies` but for each pair of groups (see examples below).

If only a global test was performed, `summary.dbMANOVAspecies_pairwise` provides a data frame with two rows named "SES" for the SES values (of the QB statistic) and "pvalue" for the P values and as many columns as there are pairs of groups of plots. Columns are named according to the two groups that are compared. For example, if there are two groups named Group1 and Group2, they are compared in column named "Group1:Group2". Else, it provides a data frame with species names as rows (in addition to a row named GLOBAL if a global test was also performed) and as columns the SES, P values, and if relevant adjusted P values, for each combination of two groups. Consider that Group1 and Group2 are the names of two groups. The column names that correspond to the comparison between Group1 and Group2 are written Group1:Group2.SES (for the SES values), Group1:Group2.pvalue (for the P values), and Group1:Group2.adj.pvalue (for the P values adjusted after correction for multiple tests, if a correction was done). NAs (= missing values) may be present in the data frame for species that are absent from two compared groups of plots. NAs may also be present in the data frame if signif = TRUE was used in function `dbMANOVAspecies_pairwise` for species that were associated to a non-significant test when performing function `dbMANOVAspecies`.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C., Pavoine, S., Cerabolini, B.E.L., Pillar, V.D. (2021) A new method for indicator species analysis in the framework of multivariate analysis of variance. *Journal of Vegetation Science*, **32**, e13013. doi:10.1111/jvs.13013

Examples

```
## Not run:
data(RutorGlacier)
Qspecies <- dbMANOVAspecies(RutorGlacier$Abund, RutorGlacier$Fac,
    nrep=9999, global=TRUE)
```

```
Qspecies_adj <- dbMANOVAspecies(RutorGlacier$Abund, RutorGlacier$Fac,
    nrep=9999, global=FALSE, padj = "BY")
# In Qspecies and Qspecies_adj, Column "Std.Obs" contains the SES values.
# Now for all species that showed significant compositional difference among
# the three groups of plots (with a nominal alpha error of 0.05), we can test for
# pairwise differences among all pairs of groups thanks to function
# dbMANOVAspecies_pairwise as shown below:

# without any correction for multiple tests
Qspeciespairwise <- dbMANOVAspecies_pairwise(Qspecies)
summary(Qspeciespairwise)
# NAs are present in the data frame above for species that were associated
# to non-significant test in object Qspecies (where tests were done over all groups);
# and also, for species that are absent from the two compared groups of plots
# (e.g. species Adenostyles leucophylla, in mid- and late-successional stages).

# with correction for multiple test
Qspeciespairwise_adj <- dbMANOVAspecies_pairwise(Qspecies_adj)
summary(Qspeciespairwise_adj)
# Here again, NAs are present in the data frame above for species that were
# associated to non-significant test in object Qspecies_adj
# (where tests were done over all groups); and also, for species that are absent
# from the two compared groups of plots (e.g. species Adenostyles leucophylla,
# in mid- and late-successional stages).

# See Ricotta et al. 2021 Appendix 3 for details.

## End(Not run)
```

---

decdiv	*Decomposition of trait-based diversity along the nodes of a phyloge-netic tree*

Description

The function decdiv calculates trait-based differences between the lineages that descend from a node of a phylogenetic tree in one or several communities (using presence/absence or abundance data).

The function plot.decdiv plots the result of function decdiv for one of the communities.

The function rtestdecdiv tests, for one community (with presence/absence or abundance data), if the representation of trait diversity on the phylogenetic tree highlights a nonrandom pattern.

Usage

```
decdiv(phyl, comm, dis = NULL, tol = 1e-08, option = 1:5,
    formula = c("QE", "EDI"))

## S3 method for class 'decdiv'
plot(x, ncom = 1, col = "black",
```

```
        csize = 1, legend = TRUE, ...)

  rtestdecdiv(phyl, vecab, dis = NULL, tol = 1e-08,
      option = 1:5, formula = c("QE", "EDI"),
      vranking = c("complexity", "droot"),
      ties.method = c("average", "first", "last", "random",
      "max", "min"), statistic = 1:3, optiontest = NULL, nrep = 99)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust with species as tips. To ease the interpretation of the results, I advise you to add labels to the nodes of the phylogeny. For example, If your phylogeny is named 'tree' and is of class phylo, you can name the nodes with the following command: tree$node.label <- paste("n", 1:tree$Nnode, sep=""). Use plot(tree, show.node.label=TRUE) to see the result. If it is of class phylo4, you can use: nodeLabels(tree) <- paste("n", 1:nNodes(tree), sep="").
comm	a vector with species presence/absence or an index of abundance as entries, or a data frame or a matrix typically with communities as rows, species as columns and presence/absence or an index of abundance as entries. Species names in comm must be similar as in phyl. Even if comm is a vector, each entry needs to have a species name (use function name to attribute names to a vector).
dis	either NULL or an object of class dist that contains the trait-based distances among species. If NULL, the Gini-Simpson index is used.
tol	a tolerance threshold (a value between -tol and tol is considered equal to zero)
option	a numeric (either 1, 2, 3, 4 or 5) indicating the option to use to calculate the contribution of each node of the phylogenetic tree to trait-based diversity. See details. If several values are given, the function retains only the first one.
formula	either "QE" (default) or "EDI". See details. If several values are given, the function retains only the first one.
x	an object of class decdiv.
ncom	if comm is a matrix, the number of its row corresponding to the focal community for which the results of decdiv will be plotted. Igored if comm is a vector.
col	the color of circles displayed at each node.
csize	a positive numeric giving the scale for plotting the circle at each node. 1 is the default size; if zero, no circle is drawn.
legend	a logical indicating whether the legend for the circle size needs to be displayed.
...	further arguments that can be specified to the internal use of function plot.phylo (argument y.lim cannot be modified).
vecab	a numeric vector giving the presence/absence(1/0) or abundance(non-negative value) of species in a community.
vranking	a string with 2 possible values: either "complexity" for a ranking according to the complexity of the subtree rooted on each interior node (see Pavoine et al. 2010 for an explanation), or "droot" for ordering interior nodes by the distance between them and the root node of the tree.

ties.method	a string to be passed to function rank of the base of R. It can be one of "average", "first", "last", "random", "max", "min".
statistic	a numeric value or a vector of numeric values. Possible values are 1, 2, or 3. They correspond to the three statistics S1, S2 and S3, respectively, developed by Pavoine et al. (eqs. 5 to 7 in Pavoine et al. 2010).
optiontest	a vector of strings specifying the alternative hypothesis of each test, which must be one of "greater", "less" or "two-sided". If null, then statistic=1 is associated with "greater" and statistic=2 and =3 with "two-sided". See function as.randtest of package ade4 for details on these alternatives. The length of the vector given to optiontest must be equal to that given to statistic.
nrep	numeric; the number of permutations to be done in each permutation test.

Details

The function decdiv relies on Rao's (1982) quadratic entropy (QE) to measure the trait-based diversity of a set of species. Two formulas for QE have been introduced in the literature one is the original formula by Rao (1982) (which corresponds to formula = "QE") and the other one introduced by Champely and Chessel (2002), named Euclidean Diversity Index (which corresponds to formula = "EDI"). See function QE for more details.

In function decdiv, each node has a weight proportional to the summed relative abundance of its descending species (or to the relative number of descending species if presence/absence data are used).

With option = 1, the function decdiv apportions trait-based diversity across the nodes of a phylogenetic tree using the algorithm defined in Pavoine et al. (2010). In that case the value at a given node is equal to the weight of a node times a measure of beta trait-based diversity between the lineages that descend from the node. The sum of all values attributed to the nodes of a phylogeny is equal to the total trait-based diversity of the species (tips of the phylogeny) as defined by Rao's quadratic entropy. In case of dichotomic trees, only two lineages descend from a given node. Here I consider a more general case where more than two lineages may descend from a node (polytomy). The beta trait-based diversity among the lineages that descend from a node is measured here as the average trait-based dissimilarity between any two of these descending lineages. With option = 1, the trait-based dissimilarity between two lineages is measured by Rao's DISC index (gamma diversity [average trait-based dissimilarity between any two species descending from the node] - alpha diversity [average trait-based dissimilarity between any two species descending from one of the lineages branched to the node]).

In the present version of function decdiv, I have added other options. Options 2 and 3 code different ways of measuring trait-based differences between lineages, standardized between 0 and 1: with option = 2, the formula is (gamma - alpha) / (1 - alpha) * M / (M - 1), where gamma and alpha are defined above (for option = 1) and M is the number of lineages descending from the node; with option = 3, the formula is (gamma - alpha) / (1 - gamma) / (M - 1).

Options 4 and 5 decompose the result given by option = 1. option = 4 returns gamma minus alpha (a measure of beta functional diversity between the lineages that descend from a node). option = 5 returns the weights of the nodes (the summed relative abundance of its descending species or the relative number of descending species if presence/absence data are used).

Values for option different from 1 needs that values in dis (the trait-based dissimilarities between species) are bounded between 0 and 1 if formula = QE or sqrt(2) if formula = EDI. If they are not bounded, the function decdiv will bound them using the maximum observed value in dis.

The argument `ties.method` in `rtestdecdiv` allows you to explicitly take into account potential ties when ranking nodes according to their complexity or their distance to root (see Pavoine et al. 2010 for more details on the permutation test implemented in rtestdecdiv).

Value

Function `decdiv` returns a matrix with nodes of the phylogenetic tree as rows and the decomposition of trait-based diversity in communities as columns. An attribute of this matrix is the phylogenetic tree (of class `phylo` with specified names for nodes). If the nodes of `phyl` had no names, the function `decdiv` automatically attributed names to them.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Champely, S. and Chessel, D. (2002) Measuring biological diversity using Euclidean metrics. *Environmental and Ecological Statistics*, **9**, 167–177.

Pavoine, S., Baguette, M., Bonsall, M.B. (2010) Decomposition of trait diversity among the nodes of a phylogenetic tree. *Ecological Monographs*, **80**, 485–507.

Rao, C.R. (1982) Diversity and dissimilarity coefficients: a unified approach. *Theoretical Population Biology*, **21**, 24–43.

Examples

```
## Not run:
if(require(ape) && require(adephylo)){
data(ungulates)
ung.phy <- read.tree(text=ungulates$tre)
ung.phy$node.label
plot(ung.phy, show.node.label=TRUE)
# Regaring traits, we log-tranformed the first three traits
# measuring volumes
# and we standardized all the traits (mean=0; variance=1).
tab <- cbind.data.frame(afbw = log(ungulates$tab$afbw),
mnw = log(ungulates$tab$mnw), fnw = log(ungulates$tab$fnw),
ls = ungulates$tab$ls)
ung.tab0 <- data.frame(scalewt(tab))
ung.tab0 <- data.frame(scalewt(log(ungulates$tab)))
ung.pres <- rep(1, nrow(ung.tab0))
names(ung.pres) <- rownames(ung.tab0)
ung.dec1 <- decdiv(ung.phy, ung.pres, dist(ung.tab0),
    option=1, formula = "EDI")
plot.decdiv(ung.dec1)

ung.dec2 <- decdiv(ung.phy, ung.pres, dist(ung.tab0),
    option=2, formula = "EDI")
plot.decdiv(ung.dec2)

ung.dec3 <- decdiv(ung.phy, ung.pres, dist(ung.tab0),
```

```
    option=3, formula = "EDI")
plot.decdiv(ung.dec3)

ung.dec4 <- decdiv(ung.phy, ung.pres, dist(ung.tab0),
    option=4, formula = "EDI")
plot.decdiv(ung.dec4)

ung.dec5 <- decdiv(ung.phy, ung.pres, dist(ung.tab0),
    option=5, formula = "EDI")
plot.decdiv(ung.dec5)
}

## End(Not run)
```

---

dislptransport	*Algorithmic index of plot-to-plot functional (or phylogenetic) dissimilarity*

Description

The function `dislptransport` calculates the DKG measure of dissimilarity (Kosman 1996; Gregorius et al. 2003) applied here to functional differences between plots (eq. 3 in Ricotta et al. 2021).

Usage

```
dislptransport(comm, dis, diag = FALSE, upper = FALSE, Nind = 10000)
```

Arguments

comm	a matrix containing the relative or absolute abundance of all species in plots. Columns are species and plots are rows. Column labels (species names) should be assigned as in `dis`.
dis	a matrix or an object of class dist providing the functional dissimilarities between species (dissimilarities are nonnegative, symmetric, and the dissimilarity between a species and itself is zero). **Species here must be in the same order as in the columns of comm**.
diag	logical value indicating whether the diagonal of the dissimilarity matrix should be printed by `print.dist`.
upper	logical value indicating whether the upper triangle of the distance matrix should be printed by `print.dist`.
Nind	an integer. The algorithmic index will be applied by assuming that each plot contains `Nind` individuals. The highest `Nind`, the most precise the index value will be (see Gregorius et al. 2003, for more details).

Value

The function `dislptransport` returns an object of class `dist` matrix with the values of the dissimilarity index DKG (Ricotta et al. 2021) for each pair of plots.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C., Kosman, E., Laroche, F., Pavoine, S. (2021) Beta redundancy for functional ecology. *Methods in Ecology and Evolution*, **12**, 1062–1069. doi:10.1111/2041210X.13587

Gregorius, H.-R., Gillet, E.M., Ziehe, M. (2003) Measuring differences of trait distributions between populations. *Biometrical Journal*, **8**, 959–973. doi:10.1002/bimj.200390063

Kosman, E. (1996) Difference and diversity of plant pathogen populations: a new approach for measuring. *Phytopathology*, **86**, 1152–1155.

See Also

`betaUniqueness` for plot-to-plot functional (or phylogenetic) uniqueness indices.

Examples

```
## Not run:
data(RutorGlacier)
fundis <- dist(scale(RutorGlacier$Traits2[1:6]))
fundis <- fundis/max(fundis)
DKG <- dislptransport(RutorGlacier$Abund, fundis)

## End(Not run)
```

---

dissABC   *Phylogenetic and Functional Similarity between Communities*

Description

Coefficients of similarity between communities that rely on the presence/absence of species are generally based on various combinations of the matching/mismatching components of the classical 2 x 2 contingency table. Three of these components are: a=the number of species shared by the two communities; b=the number of species in the first community that are not in the second; c=the number of species in the second community that are not in the first. These coefficients are extended in `dissABC` to include phylogenetic or functional information on species (Ricotta and Pavoine 2015).

Usage

```
dissABC(comm, dis, option = 1:4, method = c("J", "S", "O", "K", "SS","Si"))
```

Arguments

comm	a data frame or a matrix typically with communities as rows, species as columns and relative abundance or absolute abundance as entries. Column labels (species names) should be assigned as in the object dis.
dis	a matrix (or data frame) of (phylogenetic or functional) dissimilarities among species rescaled in the range [0, 1] or an object of class dist containing these dissimilarities [obtained by functions like dsimFun in this package adiv, vegdist in package vegan, gowdis in package FD, or dist.ktab in package ade4 for functional dissimilarities, or functions like dsimTree in this package adiv, cophenetic.phylo in package ape or distTips in package adephylo for phylogenetic dissimilarities]. See also function dsimTaxo for taxonomic data. Note that dissimilarities among species need first to be rescaled in the range [0, 1]. If the dissimilarities are outside the range 0-1, a warning message is displayed and each dissimilarity is divided by the maximum over all pair-wise dissimilarities.
option	a numeric, either 1, 2, 3, or 4 (if several values are given only the first one is considered). See details.
method	a character or string, either "J", "S", "O", "K", "SS", or "Si" (if several values are given only the first one is considered). See details.

Details

To obtain the dissimilarities among plots, one needs to choose the equations to be used for the (phylogenetic or functional) components A, B, and C thanks to argument option and the way the components will be combined, thanks to argument method.

Let $\mathbf{D} = (d_{ij})$ a matrix of (functional, morphological or phylogenetic) dissimilarities between pairs of species with $d_{ij} = d_{ji}$ and $d_{ii} = 0$. If the dissimilarity coefficient *d* is in the range [0, 1], it is possible to define a corresponding similarity coefficient *s* as the complement of *d*: *s* = 1 - *d*. Let x_{ik} the abundance of species *i* in community *k*. *S(kh)* is the number of species in the pooled communities *k* and *h* (i.e. the species for which $min\{x_{ik}, x_{ih}\} > 0$). The (absolute) abundance of species similar to *i* in plot *k* is

$$Z_{ik} = \sum_{j=1}^{S(kh)} x_{jk} s_{ij}$$

.

If option=1, equations 6-8 of the main text of Ricotta and Pavoine (2015) are used for calculating components A, B, C:

$$A = \sum_{i=1}^{S(kh)} min\{Z_{ik}, Z_{ih}\}$$

$$B = \sum_{i=1}^{S(kh)} (max\{Z_{ik}, Z_{ih}\} - Z_{ih})$$

$$C = \sum_{i=1}^{S(kh)} (max\{Z_{ik}, Z_{ih}\} - Z_{ik})$$

If option=2, equations A1-A3 from Appendix S1 of Ricotta and Pavoine (2015) are used.

If `option=3`, equations A5-A7 from Appendix S1 of Ricotta and Pavoine (2015) are used.

If `option=4`, equations A10-A12 from Appendix S1 of Ricotta and Pavoine (2015) are used.

If `method="J"`=the Jaccard index is used:

$$\frac{A}{A + B + C}$$

If `method="S"`=the Sorensen index is used:

$$\frac{2A}{2A + B + C}$$

If `method="O"`=the Ochiai index is used:

$$A/(\sqrt{A + B}\sqrt{A + C})$$

If `method="K"`, the Kulczynski index is used:

$$\frac{1}{2}\left(\frac{A}{A + B} + \frac{A}{A + C}\right)$$

If `method="SS"`, the Sokal-Sneath index is used:

$$\frac{A}{A + 2B + 2C}$$

If `method="Si"`, the Simpson index is used:

$$\frac{A}{A + min(B, C)}$$

Value

Function `dissABC` returns a matrix with the values of the proposed similarities among communities based on interspecies resemblances.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C. and Pavoine, S. (2015) Measuring similarity among plots including similarity among species: an extension of traditional approaches. *Journal of Vegetation Science*, **26**, 1061–1067.

See Also

[discomQE](), [evodiss]()

Examples

```
data(RP15JVS)
dissABC(RP15JVS$ab, RP15JVS$D1, method="J", option=1)
J <- as.matrix(dissABC(RP15JVS$ab, RP15JVS$D1, method="J", option=1))[, 1]
SS <- as.matrix(dissABC(RP15JVS$ab, RP15JVS$D1, method="SS", option=1))[, 1]
S <- as.matrix(dissABC(RP15JVS$ab, RP15JVS$D1, method="S", option=1))[, 1]
O <- as.matrix(dissABC(RP15JVS$ab, RP15JVS$D1, method="O", option=1))[, 1]
K <- as.matrix(dissABC(RP15JVS$ab, RP15JVS$D1, method="K", option=1))[, 1]
plot(1:9, J,
xlab="Number of the plots which plot 1 is compared to",
 ylab="Similarity", type="b", ylim=c(0,1), pch=18)
lines(1:9, SS, type="b", pch=15)
lines(1:9, S, type="b", pch=17)
lines(1:9, O, type="b", pch=12)
lines(1:9, K, type="b", pch=1)
legend("bottomleft",
c("Jaccard","Sokal-Sneath","Sorensen","Ochiai","Kulczynski"),
pch=c(18,15,17,12,1), lty=1)
```

---

dissRicotta   *Plot-to-Plot Functional or Phylogenetic Dissimilarity*

Description

The function calculates plot-to-plot functional or phylogenetic dissimilarity based on index D_{AB} in Ricotta et al. (2015).

Usage

```
dissRicotta(comm, dis)
```

Arguments

comm	a matrix of the relative or absolute abundance of species in communities. Columns are species and communities are rows. Column labels (species names) should be assigned as in dis.
dis	a matrix of (functional or phylogenetic) dissimilarities rescaled in the range [0, 1] or an object of class dist [obtained by functions like dsimFun in this package adiv, vegdist in package vegan, gowdis in package FD, or dist.ktab in package ade4 for functional dissimilarities, or functions like dsimTree in this package adiv, cophenetic.phylo in package ape or distTips in package adephylo for phylogenetic dissimilarities; See also function dsimTaxo for taxonomic data].

Value

The function returns a semi-matrix of class `dist` with the values of the proposed dissimilarities for each pair of plots. Note that dissimilarities among species need first to be rescaled in the range [0, 1]. If the dissimilarities are outside the range 0-1 (as it is usually the case in phylogenetic studies for instance), a warning message is displayed and all dissimilarities are divided by the maximum observed dissimilarity.

Author(s)

Giovanni Bacaro and Sandrine Pavoine `<sandrine.pavoine@mnhn.fr>`

References

Ricotta, C., Bacaro, G., Pavoine, S. (2015) A cautionary note on some phylogenetic dissimilarity measures. *Journal of Plant Ecology*, **8**, 12–16.

Examples

```
## Not run:
if(require(ape)){
# Phylogenetic tree
s<-"test(((v:20,w:20):10,(x:20,y:20):10):15,z:45):5;"
plot(test <- read.tree(text=s))

# Phylogenetic distances among species
tdist <- cophenetic(test)/100

# Matrix of abundances of the species in four communities;
# communities A and C are identical;
# communities B and D are identical;
comm <- t(data.frame(A = rep(0.2, 5),
B = c(0.1, 0.2, 0.2, 0, 0.5), C = rep(0.2, 5),
D = c(0.1, 0.2, 0.2, 0, 0.5), row.names = letters[22:26]))

# Index DAB
dissRicotta(comm, tdist)
}

## End(Not run)
```

---

distinctAb	*Abundance-based measures of species' rarity, functional or phylogenetic distinctiveness and functional or phylogenetic effective originality*

Description

The function calculates parametric indices of species' rarity and functional or phylogenetic distinctiveness and effective originality according to Pavoine and Ricotta (2021).

Usage

```
distinctAb(comm, disORtree, method = c("Q", "KY", "KstarI"), palpha = 2,
    option = c("asymmetric", "symmetric"), tol = 1e-10)
```

Arguments

comm   a data frame or a matrix typically with communities as rows, species as columns
                and abundance as entry. Species should be labelled as in object `disORtree`.

disORtree   an object inheriting the class `dist`, giving the FP-dissimilarities between species,
                or inheriting the class `phylo` (see package ape), `phylo4` (see package phy-
                lobase), or `hclust`, where species are tips.

method   a string either `"Q"` for the quadratic entropy, `"KY"` for index $^{\alpha}K$ (if `disORtree`
                is of class `dist`) or $^{\alpha}Y$ if `disORtree` is of class `phylo`, `phylo4` or `hclust`; or
                `"KstarI"` for index $^{\alpha}K*$ (if `disORtree` is of class `dist`) or $^{\alpha}I$ if `disORtree` is
                of class `phylo`, `phylo4` or `hclust`. If several values are given, only the first one
                is considered.

palpha   a numeric, nonnegative value for parameter α. `palpha` is ignored if `method` =
                `"Q"`.

option   a string either `"asymmetric"`, or`"symmetric"` The parameter `option` is only
                used if `disORtree` is of class `phylo`, `phylo4` or `hclust` If `option="symmetric"`,
                the distance between two tips on a tree is defined as half the sum of branch length
                on the smallest path that connects the two species; while if `option="asymmetric"`
                the distance between a tip *i* and a tip *j* on a tree is defined as the sum of branch
                lengths between tip *i* and its most recent ancestor with tip *j*. If the tree is ultra-
                metric, the two options are equivalent.

tol   numeric tolerance threshold: values between `-tol` and `tol` are considered equal
                to zero.

Value

If `palpha <= 1`, then, the function returns a list of four objects of class `data.frame` (with com-
munities as rows and species as columns): `TotContr` provides for each species in each community
the effective originality multiplied by the relative abundance; `EffOriPres` provides for each species
present in each community the effective originality; NA for absent species; `DistinctPres` provides
for each species present in each community the distinctiveness; NA for absent species; `Rarity` pro-
vides for each species in each community the rarity (maximum rarity for absent species).

Else, the function returns a list with the three objects of class `data.frame` presented above and
two additional objects also of class `data.frame`: `EffOriAll` provides for each species its effective
originality compared with the composition of each community; even absent species have a value
considering they have zero abundance so maximum rarity and considering their functional dis-
similarity or phylogenetic dissimilarity with all species present in each community; `DistinctAll`
provides for each species its effective originality compared with the composition of each commu-
nity; even absent species have a value considering they have zero abundance so maximum rarity
and considering their functional dissimilarity or phylogenetic dissimilarity with all species present
in each community.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S., Ricotta, C. (2021) On the relationships between rarity, uniqueness, distinctiveness, originality and functional/phylogenetic diversity. *BiorXiv*. doi:10.1101/2021.08.09.455640

See Also

distinctDis, distinctTopo, distinctTree, distinctUltra

Examples

```
## Not run:
if(require(ape) && require(adephylo) && require(phylobase)){
data(batcomm)
phy <- read.tree(text=batcomm$tre2)
disAb <- distinctAb(batcomm$ab2, phy, method="Q")
U.4d <- phylo4d(phy, t(disAb[[2]]))
dotp4d(U.4d, center = FALSE, scale = FALSE, data.xlim = range(t(disAb[[2]]), na.rm=TRUE),
    tree.ratio = 0.20, dot.cex=2)
title("Effective originality associated with quadratic entropy (diversity)")
}

## End(Not run)
```

---

distinctDis   *Dissimilarity-based Species' Originality*

Description

The function calculates five indices of species' originality.

Usage

```
distinctDis(dis, method = c("Rb", "AV", "FV", "NN", "Dstar", "full"),
    palpha = 0, standardized = FALSE)
```

Arguments

dis   an object of class dist containing pair-wise (functional or phylogenetic) dis-
             similarities between species.

method   a vector of strings. Possible values are "Rb", "AV", "FV", "NN", "Dstar" and
             "full". "Rb" is for Pavoine et al. (2017) index Rb; "AV" is for AV, the average
             dissimilarity between a species and all others in a set (Eiswerth and Haney 1992;
             Ricotta 2004); "FV" is for FV, the average dissimilarity between a species and
             any other (including the focal species itself) (Schmera et al. 2009); "NN" is for

	the minimum dissimilarity between a species and any other (the dissimilarity to its Nearest Neighbor) (Pavoine et al. 2017); Dstar is the parametric indices $^{\alpha}D*$ developed by Pavoine and Ricotta (2021, Appendix B); "full" returns all indices.
palpha	a numeric which provides the value of parameter α in index $^{\alpha}D*$ developed by Pavoine and Ricotta (2021, Appendix B).
standardized	a logical. If TRUE, the vector of originalities is divided by its sum (transforming absolute originalities into relative originalities).

Value

A data frame with species as rows and originality indices as columns.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Eiswerth, M.E. and Haney, J.C. (1992) Allocating conservation expenditures: accounting for interspecies genetic distinctiveness. *Ecological Economics*, **5**, 235–249.

Pavoine, S., Bonsall, M.B., Dupaix, A., Jacob, U., Ricotta, C. (2017) From phylogenetic to functional originality: guide through indices and new developments. *Ecological Indicators*, **82**, 196–205.

Pavoine, S., Ricotta, C. (2021) On the relationships between rarity, uniqueness, distinctiveness, originality and functional/phylogenetic diversity. *BiorXiv*. doi:10.1101/2021.08.09.455640

Ricotta, C. (2004) A parametric diversity measure combining the relative abundances and taxonomic distinctiveness of species. *Diversity and Distributions*, **10**, 143–146.

Schmera, D., Podani, J., Eros, T. (2009) Measuring the contribution of community members to functional diversity. *Oikos*, **118**, 961–971.

See Also

distinctTopo, distinctTree, distinctUltra

Examples

```
e <- rlnorm(10)
e <- sort(e)
names(e) <- paste("s", 1:10, sep="")
d <- dist(e)
barplot(e)

D <- distinctDis(d, standardized = TRUE)

par(mfrow=c(4,2))
plot(e, D[,1], xlab="trait", ylab="Rb")
plot(e, D[,2], xlab="trait", ylab="AV")
plot(e, D[,3], xlab="trait", ylab="FV")
```

```
plot(e, D[,4], xlab="trait", ylab="NN")

plot(D[,1], D[,2], xlab="Rb", ylab="AV")
plot(D[,1], D[,3], xlab="Rb", ylab="FV")
plot(D[,2], D[,3], xlab="AV", ylab="FV")
plot(D[,2], D[,4], xlab="AV", ylab="NN")
par(mfrow=c(1,1))
```

---

distinctTopo *Topology-based Species' Originality*

Description

The function calculates three indices of species' originality that rely on the topology of (phylogenetic) trees. Trees with polytomies are allowed.

Usage

```
distinctTopo(phyl, method = c("VW","M","A","full"), standardized = FALSE)
```

Arguments

phyl   an object inheriting the class `phylo` (see package ape), `phylo4` (see package phylobase), or `hclust`.

method   a character or string or a vector of characters/strings. Possible values are `"VW"`, `"M"`, `"A"` and `"full"`. `"VW"` is for Vane-Wright et al. index (Vane-Wright et al. 1991); `"M"` is for May index (May 1990); `"A"` is for Pavoine et al. index of originality derived from Abouheif (1999) (Pavoine et al. 2008); and "full" returns all indices.

standardized   a logical. If `FALSE`, the minimum score is scaled to 1 as in May (1990). If `TRUE`, the vector of originalities is divided by its sum (transforming absolute originalities into relative originalities).

Value

A data frame with species as rows and originality indices as columns.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr> with contributions of Stephane Dray

References

May, R.M. (1990) Taxonomy as destiny. *Nature*, **347**, 129–130.

Vane-Wright, R.I., Humphries, C.J., Williams, P.H. (1991) What to protect? Systematics and the agony of choice. *Biological Conservation*, **55**, 235–254.

Abouheif, E. (1999) A method for testing the assumption of phylogenetic independence in comparative data. *Evolutionary Ecology Research*, **1**, 895–909.

Pavoine, S., Ollier, S., Pontier, D., Chessel, D. (2008) Testing for phylogenetic signal in phenotypic traits: new matrices of phylogenetic proximities. *Theoretical Population Biology*, **73**, 79–91.

See Also

[distinctDis](), [distinctTree](), [distinctUltra]()

Examples

```
## Not run:
if(require(ape) && require(adephylo) && require(phylobase)){
data(carni70, package = "adephylo")
tre <- read.tree(text=carni70$tre)
U <- distinctTopo(tre, standardized = TRUE)
U.4d <- phylo4d(tre, as.matrix(U))
dotp4d(U.4d, center = FALSE, scale = FALSE)
barp4d(U.4d, center = FALSE, scale = FALSE)
gridp4d(U.4d, center = FALSE, scale = FALSE)

parmar <- par()$mar
par(mar=rep(.1,4))
table.phylo4d(U.4d, show.node=FALSE, symbol="squares", center=FALSE, scale=FALSE, cex.symbol=2)
par(mar=parmar)
}

## End(Not run)
```

---

distinctTree   *Tree-based Species' Originality*

Description

The function calculates indices of species' originality that rely on the structure and branch lengths of (phylogenetic) trees. Trees with polytomies are allowed.

Usage

```
distinctTree(phyl, method = c("ED", "ES", "Delta*"), palpha = 0, standardized = FALSE)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust.
method	a string or a vector of strings. Possible values are "ED", "ES", and "Delta*". "ED" is for the evolutionary distinctiveness, also named fair-proportion, index (Redding 2003; Isaac et al. 2007); "ES" is for the Equal-Splits index (Redding and Mooers 2006); "Delta*" is for the parametric $^\alpha\Delta$ index of Pavoine and Ricotta (2021).

| palpha | a numeric value or a numeric vector of values for parameter α of $^{\alpha}\Delta$ index by Pavoine and Ricotta (2021). |
| standardized | a logical. If TRUE, the vector of originalities is divided by its sum (transforming absolute originalities into relative originalities). |

Value

A data frame with species as rows and originality indices as columns.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Isaac, N.J., Turvey, S.T., Collen, B., Waterman, C., Baillie, J.E. (2007) Mammals on the EDGE: conservation priorities based on threat and phylogeny. *PloS ONE*, **2**, e296.

Pavoine, S., Ricotta, C. (2021) On the relationships between rarity, uniqueness, distinctiveness, originality and functional/phylogenetic diversity. *BiorXiv*. doi:10.1101/2021.08.09.455640

Redding, D.W. (2003) *Incorporating genetic distinctness and reserve occupancy into a conservation priorisation approach*. Master thesis: University of East Anglia, Norwich.

Redding, D.W., Mooers, A.O. (2006) Incorporating evolutionary measures into conservation prioritization. *Conservation Biology*, **20**, 1670–1678.

See Also

distinctDis, distinctTopo, distinctUltra

Examples

```
## Not run:
if(require(ape) && require(adephylo) && require(phylobase)){
data(carni70, package = "adephylo")
tre <- read.tree(text=carni70$tre)
U <- distinctTree(tre, standardize = TRUE)
U.4d <- phylo4d(tre, as.matrix(U))
dotp4d(U.4d, center = FALSE, scale = FALSE)
barp4d(U.4d, center = FALSE, scale = FALSE)
gridp4d(U.4d, center = FALSE, scale = FALSE)

parmar <- par()$mar
par(mar=rep(.1,4))
table.phylo4d(U.4d, show.node=FALSE, symbol="squares", center=FALSE, scale=FALSE, cex.symbol=2)
par(mar=parmar)
}

## End(Not run)
```

---

distinctUltra | *Ultrametric Tree-based Species' Originality*

Description

The function calculates two indices of species' originality that rely on the structure and branch lengths of ultrametric (phylogenetic) trees. Trees with polytomies are allowed.

Usage

```
distinctUltra(phyl, method = c("Qb","2Hb"))
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase), or hclust, with ultrametric properties.
method	a string or a vector of strings. Possible values are Qb and 2Hb. Qb is for the Pavoine et al. QE-based (also named Qb) index (Pavoine et al. 2005); 2Hb is for Pavoine et al. 2H-based index (which could also be named more shortly as 2Hb) (vector that maximizes index ^2H, Pavoine and Izsak 2014).

Value

A data frame with species as rows and originality indices as columns.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr> with contributions of Stephane Dray

References

Pavoine, S., Ollier, S., Dufour, A.B. (2005) Is the originality of a species measurable? *Ecology Letters*, **8**, 579–586.

Pavoine, S. and Izsak, J. (2014) New biodiversity measure that includes consistent interspecific and intraspecific components. *Methods in Ecology and Evolution*, **5**, 165–172.

See Also

[distinctDis](), [distinctTopo](), [distinctTree]()

Examples

```
## Not run:
if(require(ape) && require(adephylo) && require(phylobase)){
data(carni70, package = "adephylo")
tre <- read.tree(text=carni70$tre)
U <- distinctUltra(tre)
U.4d <- phylo4d(tre, as.matrix(U))
```

```
dotp4d(U.4d, center = FALSE, scale = FALSE)
barp4d(U.4d, center = FALSE, scale = FALSE)
gridp4d(U.4d, center = FALSE, scale = FALSE)

parmar <- par()$mar
par(mar=rep(.1,4))
table.phylo4d(U.4d, show.node=FALSE, symbol="squares", center=FALSE, scale=FALSE, cex.symbol=2)
par(mar=parmar)
}

## End(Not run)
```

---

distMS   *Marczewski-Steinhaus Coefficient*

Description

The function calculates the Marczewski-Steinhaus coefficient of dissimilarity between pairs of entities (e.g. communities)

Usage

```
distMS(comm, diag = FALSE, upper = FALSE)
```

Arguments

comm	a data frame or a matrix of nonnegative values (e.g. abundance of species (columns) within communities (rows) to obtain dissimilarities between communities).
diag	a logical value indicating whether the diagonal of the distance matrix should be printed by function `print.dist`.
upper	a logical value indicating whether the upper triangle of the distance matrix should be printed by function `print.dist`.

Value

an object of class `dist`

Note

This function is a modification of function `dist.quant` from library ade4 where other dissimilarity coefficients can be found.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Orloci, L. (1978) *Multivariate Analysis in Vegetation Research*. The Hague: Junk.

Legendre, P. and Legendre, L. (1998) Numerical Ecology. Amsterdam: Elsevier.

Ricotta, C., de Bello, F., Moretti, M., Caccianiga, M., Cerabolini, B.E., Pavoine, S. (2016). Measuring the functional redundancy of biological communities: a quantitative guide. *Methods in Ecology and Evolution*, **7**, 1386–1395.

Examples

```
data(birdData)
distMS(birdData$fau)
```

---

divparam   *Parametric Indices of Species Diversity*

Description

The function `divparam` calculates parametric diversity indices. The parameter controls the relative importance given to rare versus abundant species in a community. The function `plot.divparam` plots the results of function `divparam`.

Usage

```
divparam(comm, method = c("hill", "tsallis", "renyi"), q = 2, tol = 1e-08)

## S3 method for class 'divparam'
plot(x, legend = TRUE,
legendposi = "topright", axisLABEL = "Diversity", type = "b",
col = if (is.numeric(x)) NULL
else sample(colors(distinct = TRUE), nrow(x$div)),
lty = if (is.numeric(x)) NULL else rep(1, nrow(x$div)),
pch = if (is.numeric(x)) NULL else rep(19, nrow(x$div)),
...)
```

Arguments

comm	a data frame or a matrix typically with communities as rows, species as columns and abundance as entry.
method	a string: either "hill" for the Hill numbers (Hill 1973), "tsallis" for the Tsallis or HCDT entropy (Harvda and Charvat 1967; Daroczy 1970; Tsallis 1988), or "renyi" for Renyi's entropy (Renyi 1960).
q	a positive numeric or a vector of positive numerics that gives values for the q parameter.
tol	numeric tolerance threshold: values between -tol and tol are considered equal to zero.

x	an object of class `divparam` obtained with function `divparam`.
legend	a logical. If TRUE a legend is given with the colour, the type of line (etc.) used to define the diversity curve of each community.
legendposi	a string or a numeric that gives the position of the legend to be passed to function `legend` of the base of R.
axisLABEL	a string to display on the main axis of the plot to designate what we are measuring. The default is "Diversity".
type	a string to be passed to the graphic argument `type` of functions `plot` and `lines` used to draw the diversity curve of each community.
col	colours to be passed to the graphic argument `col` of functions `plot` and `lines` to define the colour of the diversity curve of each community.
lty	type of line (plain, broken etc.) to be passed to the graphic argument `lty` of functions `plot` and `lines` used to draw the diversity curve of each community.
pch	type of point (open circle, close circle, square etc.) to be passed to the graphic argument `pch` of functions `plot` and `lines` used to draw the diversity curve of each community.
...	other arguments can be added and passed to the functions `plot` and `lines` used to draw the graphic.

Value

If only a single value of `q` is given, function `divparam` returns a vector with the diversities of the communities. If more than one value of `q` is given, a list of two objects is returned:

q	the vector of values for `q`
div	a data frame with the diversities of the communities calculated for all values of `q`

The function `plot.divparam` returns a graphic.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Daroczy, Z. (1970) Generalized information functions. *Information and Control*, **16**, 36–51.

Havrda, M., Charvat, F. (1967) Quantification method of classification processes: concept of structural alpha-entropy. *Kybernatica*, **3**, 30–35.

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

Renyi, A. (1960) On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, **1**, 547–561.

Tsallis, C. (1988) Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, **52**, 480–487.

Examples

```
data(batcomm)
ab <- batcomm$ab
plot(divparam(ab))
plot(divparam(ab, q=0:4))
```

DP   *Plot-to-plot functional or phylogenetic dissimilarity*

Description

The function DP calculates Ricotta et al. (2020) plot-to-plot functional or phylogenetic beta unique-
ness (index named D_F for functional data and D_P for phylogenetic data, calculated with equation
2 in Ricotta et al. 2020).

Usage

```
DP(mtree, comm, height = NULL, diag = FALSE, upper = FALSE, tol = 0.001)
```

Arguments

mtree	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase), or hclust. The tree must be ultrametric: equal distance from any tip to the root.
comm	a matrix containing the relative or absolute abundance of all species in plots. Columns are species and plots are rows. Column labels (species names) should be assigned as in mtree.
height	either NULL or a numeric. See details.
diag	a logical value indicating whether the diagonal of the distance matrix should be printed by function `print.dist`.
upper	a logical value indicating whether the upper triangle of the distance matrix should be printed by function `print.dist`.
tol	a tolerance threshold. A value between -tol and tol is considered as zero. See details.

Details

Object mtree defines a tree with species as tips. If argument height is NULL, then the root of the
tree will be placed at the most recent common ancestor of all species occurring in the set of plots
(given in object comm). An alternative position for the root can be given by specifying the height of
the tree (argument height). In that case, height must be higher than the distance between tips and
the most recent common ancestor of all species.

The tolerance threshold tol is particularly important if your tree is not exactly ultrametric due
to approximation problems. In that case, the distance from tip to root varies according to the tip
considered, although it should not (variations are due to approximation problems). A difference
smaller than tol in the distance to root for two species will thus be considered as null.

Value

The function returns a (semi-)matrix of class dist with the values of the proposed dissimilarities for each pair of plots.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C., Laroche, F., Szeidl, L., Pavoine, S. (2020) From alpha to beta functional and phylogenetic redundancy. *Methods in Ecology and Evolution*. In press.

See Also

[betaTreeUniqueness](#) for beta uniqueness, [treeUniqueness](#) for alpha uniqueness

Examples

```
## Not run:
if(require(ape) && require(ade4)){
data(RutorGlacier)
phy <- read.tree(text=RutorGlacier$TreeNW)
plot(phy)

ab <- RutorGlacier$Abund[, phy$tip.label]

# Phylogenetic dissimilarities between plots
# (Ricotta et al. 2020)
Dp <- DP(phy, ab, tol=0.00001)
# Principal Coordinate Analysis of the dissimilarities
pcoDp <- dudi.pco(sqrt(Dp), full=TRUE)
s.class(pcoDp$li, as.factor(RutorGlacier$Fac))
}

## End(Not run)
```

---

dsimcom   *Functional or Phylogenetic Similarity between Species and Communities*

Description

The R function dsimcom calculates Pavoine and Ricotta (2014) coefficients SSokal-Sneath, SJaccard, SSorensen, SOchiai, and Sbeta of similarities among communities or their complements (1-S, dissimilarities).

Function SQ calculates Pavoine and Ricotta (2014) index S_Q (similarities between communities) and its complements $1 - S_Q$ (dissimilarities between communities).

The functions `dsimTax` and `dsimTree` calculate pair-wise taxonomic and phylogenetic (dis)similarities between species, respectively. `dsimTree` can also be used to calculate pair-wise functional (dis)similarities between species if a functional dendrogram is used to describe species.

Function `dsimFun` calculates pair-wise functional (dis)similaties between species.

Usage

```
dsimcom(comm, Sigma = NULL, method = 1:5,
option = c("relative", "absolute"),
type = c("similarity", "dissimilarity"))

SQ(comm, Sigma = NULL, type = c("similarity", "dissimilarity"))

dsimTaxo(tax, method = c(1, 2, 3, 4, 5),
type = c("similarity", "dissimilarity"))

dsimTree(phyl, method = c(1, 2, 3, 4, 5), rootedge = NULL,
type = c("similarity", "dissimilarity"))

dsimFun(df, vartype = c("Q", "N", "M", "P"), method = 1:5,
type = c("similarity", "dissimilarity"))
```

Arguments

comm	a data frame or matrix with communities as rows, species as columns and non-negative values as entries.
Sigma	matrix of similarities among species (species as rows and columns in the same order as in `comm`; values in Sigma are bounded between 0 and 1). The matrix must be nonnegative definite, i.e. all its eigenvalues are nonnegative. Functions `dsimTaxo`, `dsimTree` and `dsimFun` can be used to obtain Sigma as these functions lead to nonnegative definite matrices.
method	an integer (1, 2, 3, 4, 5) indicating which basic coefficient should be used: Sokal-Sneath, Jaccard, Sorensen, Ochiai, beta, respectively.
option	a string. If `option = "relative"`, the rows of `comm` are standardized into proportions that sum to 1. If `option = "absolute"`, raw values are retained in `comm`.
tax	an object of class `taxo` (of package ade4).
type	a string. If `type = "similarity"`, the functions `dsimcom` and `SQ` calculates similarities (S) between communities and the functions `dsimTaxo`, `dsimTree` and `dsimFun` calculates similarities between species. If `type = "dissimilarity"`, the functions `dsimcom` and `SQ` calculates dissimilarities (1-S) between communities and the functions `dsimTaxo`, `dsimTree` and `dsimFun` calculates dissimilarities between species.
phyl	an object inheriting the class `phylo` (see package ape), `phylo4` (see package phylobase), or `hclust`.

rootedge	a numeric equal to the length of the branch at the nearest common ancestor of all species (here referred to as the root node). This branch is thus anterior to the root. It is ignored if `rootedge` is NULL.
df	either a data frame, a matrix or an object of class `ktab` of package ade4. If class `ktab` is used, each data frame must contain a single type of traits (see argument `vartype`).
vartype	a vector of characters indicating the type of traits used in each data frame of `df`: "Q" for quantitative; "N" for nominal; "M" for multichoice; "P" for traits that can be expressed as proportions. Values in type must be in the same order as data frames in `df` (one value per table).

Details

Formulas for the indices are given in Pavoine and Ricotta (2014) main text and appendixes.

Value

If `type = "similarities"`, a matrix of similarities between communities. If `type = "dissimilarities"`, an object of class `dist` with the dissimilarities between communities.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S., Ricotta, C. (2014) Functional and phylogenetic similarity among communities. *Methods in Ecology and Evolution*, **5**, 666–675.

Examples

```
## Not run:
if(require(ade4)){
data(macroloire, package="ade4")

Ssokalsneath <- dsimcom(t(macroloire$fau), method=1, option=c("relative"))
Sjaccard <- dsimcom(t(macroloire$fau), method=2, option=c("relative"))
Ssorensen <- dsimcom(t(macroloire$fau), method=3, option=c("relative"))
Sochiai <- dsimcom(t(macroloire$fau), method=4, option=c("relative"))
Sbeta <- dsimcom(t(macroloire$fau), method=5, option=c("relative"))


SQUNIF <- SQ(t(macroloire$fau))

# The taxonomy is contained in macroloire$taxo

s_species_sokalsneath_taxo <- dsimTaxo(macroloire$taxo, method=1) # Using formula a/(a+2b+2c)
# (see notations below)
s_species_jaccard_taxo <- dsimTaxo(macroloire$taxo, method=2) # Using formula a/(a+b+c)
s_species_sorensen_taxo <- dsimTaxo(macroloire$taxo, method=3) # Using formula 2a/(2a+b+c)
s_species_ochiai_taxo <- dsimTaxo(macroloire$taxo, method=4) # Using a/sqrt((a+b)(a+c))
s_species_beta_taxo <- dsimTaxo(macroloire$taxo, method=5) # Using 4a/(4a+b+c)
```

```
# To check that these matrices of taxonomic similarities
# among species are positive semidefinite (p.s.d.)
# we have to verify that their eigenvalues are all nonnegative:
all(eigen(s_species_sokalsneath_taxo)$val>-(1e-8))
all(eigen(s_species_jaccard_taxo)$val>-(1e-8))
all(eigen(s_species_sorensen_taxo)$val>-(1e-8))
all(eigen(s_species_ochiai_taxo)$val>-(1e-8))
all(eigen(s_species_beta_taxo)$val>-(1e-8))

# Compositions of the communities
m <- t(macroloire$fau[rownames(s_species_sokalsneath_taxo), ])
# Taxomonic similarities among species
s <- s_species_sokalsneath_taxo

Ssokalsneath_taxo <- dsimcom(m, s, method = 1)
SQwith_species_sokalsneath_taxo <- SQ(m, s)

# Compositions of the communities
m <- t(macroloire$fau[rownames(s_species_jaccard_taxo), ])
# Taxonomic similarities among species
s <- s_species_jaccard_taxo
Sjaccard_taxo <- dsimcom(m, s, method = 2)
SQwith_species_jaccard_taxo <- SQ(m, s)

# Compositions of the communities
m <- t(macroloire$fau[rownames(s_species_sorensen_taxo), ])
# Taxonomic similarities among species
s <- s_species_sorensen_taxo

Ssorensen_taxo <- dsimcom(m, s, method = 3)
SQwith_species_sorensen_taxo <- SQ(m, s)

# Compositions of the communities
m <- t(macroloire$fau[rownames(s_species_ochiai_taxo), ])
# Taxonomic similarities among species
s <- s_species_ochiai_taxo

Sochiai_taxo <- dsimcom(m, s, method = 4)
SQwith_species_ochiai_taxo <- SQ(m, s)


# Compositions of the communities
m <- t(macroloire$fau[rownames(s_species_beta_taxo), ])
# Taxonomic similarities among species
s <- s_species_beta_taxo

Sbeta_taxo <- dsimcom(m, s, method = 5)
SQwith_species_beta_taxo <- SQ(m, s)

# The matrix named feed below contains feeding attributes as rows,
# species as columns, and affinities (proportions) as entries.
```

```
feed <- macroloire$traits[ ,-(1:4)]

# Feeding habits comprise seven categories: engulfers, shredders, scrapers,
# deposit-feeders, active filter-feeders, passive filter-feeders and piercers, in this order.

# Functional similarities among species are computed as indicated in the main text
s_species_sokalsneath_feed <- dsimFun(feed, vartype = "P", method=1)
s_species_jaccard_feed <- dsimFun(feed, vartype = "P", method=2)
s_species_sorensen_feed <- dsimFun(feed, vartype = "P", method=3)
s_species_ochiai_feed <- dsimFun(feed, vartype = "P", method=4)
s_species_beta_feed <- dsimFun(feed, vartype = "P", method=5)

all(eigen(s_species_sokalsneath_feed)$val>-(1e-8))
all(eigen(s_species_jaccard_feed)$val>-(1e-8))
all(eigen(s_species_sorensen_feed)$val>-(1e-8))
all(eigen(s_species_ochiai_feed)$val>-(1e-8))
all(eigen(s_species_beta_feed)$val>-(1e-8))

Ssokalsneath_feed <- dsimcom(t(macroloire$fau), s_species_sokalsneath_feed, method=1)
SQwith_species_sokalsneath_feed <- SQ(t(macroloire$fau), s_species_sokalsneath_feed)

Sjaccard_feed <- dsimcom(t(macroloire$fau), s_species_jaccard_feed, method=2)
SQwith_species_jaccard_feed <- SQ(t(macroloire$fau), s_species_jaccard_feed)

Ssorensen_feed <- dsimcom(t(macroloire$fau), s_species_sorensen_feed, method=3)
SQwith_species_sorensen_feed <- SQ(t(macroloire$fau), s_species_sorensen_feed)

Sochiai_feed <- dsimcom(t(macroloire$fau), s_species_ochiai_feed, method=4)
SQwith_species_ochiai_feed <- SQ(t(macroloire$fau), s_species_ochiai_feed)

Sbeta_feed <- dsimcom(t(macroloire$fau), s_species_sorensen_feed, method=5)


all(eigen(Ssokalsneath_feed)$val>-(1e-8))
all(eigen(Sjaccard_feed)$val>-(1e-8))
all(eigen(Ssorensen_feed)$val>-(1e-8))
all(eigen(Sochiai_feed)$val>-(1e-8))
all(eigen(Sbeta_feed)$val>-(1e-8))


par(mfrow=c(3, 5))
plot(SQUNIF, Ssokalsneath, xlim=c(0,1), ylim=c(0,1), asp=1)
segments(0, 0, 1,1)
plot(SQUNIF, Sjaccard, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)
plot(SQUNIF, Ssorensen, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)
plot(SQUNIF, Sochiai, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)
plot(SQUNIF, Sbeta, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)

plot(SQwith_species_sokalsneath_taxo, Ssokalsneath_taxo, xlim=c(0,1), ylim=c(0,1), asp=1)
```

```
segments(0, 0, 1,1)
plot(SQwith_species_jaccard_taxo, Sjaccard_taxo, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)
plot(SQwith_species_sorensen_taxo, Ssorensen_taxo, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)
plot(SQwith_species_ochiai_taxo, Sochiai_taxo, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)
plot(SQwith_species_beta_taxo, Sbeta_taxo, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)

plot(SQwith_species_sokalsneath_feed, Ssokalsneath_feed, xlim=c(0,1), ylim=c(0,1), asp=1)
segments(0, 0, 1,1)
plot(SQwith_species_jaccard_feed, Sjaccard_feed, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)
plot(SQwith_species_sorensen_feed, Ssorensen_feed, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)
plot(SQwith_species_ochiai_feed, Sochiai_feed, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)
plot(SQwith_species_sorensen_feed, Sbeta_feed, xlim=c(0,1), ylim=c(0,1) , asp=1)
segments(0, 0, 1,1)

par(mfrow=c(1,1))
}

## End(Not run)
```

---

dspca   *Double similarity principal component analysis*

Description

The function dspca performs the double similarity principal component analysis (DSPCA) (Pavoine 2019): an ordination approach to analyse functional or phylogenetic similarities between species communities. The function plot.dspca displays factorial maps of dspca.

Usage

```
dspca(comm, S=NULL, tol=1e-8)

## S3 method for class 'dspca'
plot(x, xaxis = 1, yaxis = 2, labels = TRUE,
    arrows = TRUE, points = FALSE, autolab = TRUE, title = NULL,
    colors = NULL, type = c("X&Y","X","Y"), zoom = TRUE, ...)
```

Arguments

comm   a data frame or a matrix typically with communities (or sites, plots, etc.) as
                  rows, species as columns and presence/absence (1/0) or an index of abundance
                  as entries.

S	a matrix of functional or phylogenetic similarities between species (species as rows and columns in the same order as in comm).
tol	a numeric tolerance threshold: a value between -tol and tol is considered as zero.
x	an object of class `dspca`.
xaxis	the number of the DSPCA axis chosen for the x-axis of the 3d plot.
yaxis	the number of the DSPCA axis chosen for the y-axis of the 3d plot.
labels	a logical specifying whether the names of the species and those of the communities must be displayed in the factorial maps.
arrows	a logical specifying whether arrows must be used to indicate the positions of the species and those of the communities.
points	a logical specifying whether points must be used to indicate the positions of the species and those of the communities.
autolab	a logical indicating if a function autoLab extracted from package `FactoMineR` version 2.6 must be used to display the labels. If there are more than 50 species or more than 50 communities, this option may be time-consuming. If `autolab = FALSE`, the names of species and those of the communities will be placed quickly but may overlap.
title	a string (if `type="X"` or `type="Y"`) or a vector of strings (if `type="X&Y"`) giving the title of each graph.
colors	a vector (if `type="X"` or `type="Y"`) providing the color or the colors to be used for species names, points and arrows (if `type="X"`), or community names, points and arrows (if `type="Y"`), OR, if `type="X&Y"`, a list of two vectors, the first one with the colors to be used for species names, points and arrows and the other one for community names, points and arrows.
type	a string that can be `"X&Y"`,`"X"`,`"Y"`. If `type="X"`, species scores are displayed. If `type="Y"`, community scores are displayed. If `"X&Y"`, both species and community scores are displayed.
zoom	a logical indicating if the graphs must zoom on the positions of the species and those of the communities (defaults to `TRUE`). If `zoom=FALSE` then species and communities are positioned in the full circle of unit radius.
...	further arguments passed to or from other methods. These must be common to functions plot, arrows, points, and autoLab or text.

Details

Coordinates can be visualized with graphic tools available in R. Examples are provided below (see section named "examples").

Value

dspca returns a list of the following objects:

eig	Final eigenvalues: positive eigenvalues of the matrix of similarities among communities.

X	final coordinates of the species ($\mathbf{X}_{final}$): matrix with the coordinates of the species on the principal components associated with the matrix of similarities among communities. The names of the matrix start with "CPC" indicating "communities' principal component".
Y	final coordinates of the communities ($\mathbf{Y}_{final}$): matrix with the coordinates of the communities on the principal components associated with the matrix of similarities among communities. The names of the matrix start with "CPC" indicating "communities' principal component".
Scom	matrix of similarities among communities (obtained with coefficient SOchiai (Pavoine and Ricotta 2014))

.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S. (2019) An ordination approach to explore similarities among communities. *Journal of Theoretical Biology*, **462**, 85–96.

Pavoine, S. and Ricotta, C. (2014) Functional and phylogenetic similarity among communities. *Methods in Ecology and Evolution*, **5**, 666–675.

See Also

[evoNSCA](), [evopcachord](), [evopcahellinger](), [dsimcom]()

Examples

```
## Not run:
if(require(ade4)){

data(macroloire, package="ade4")
# Analysis of the feeding habits of macroinvertebrates in the Loire river, France

# The matrix named feed below contains feeding attributes as rows,
# species as columns, and affinities (proportions) as entries.
feed <- macroloire$traits[, -(1:4)]

# Feeding habits comprise seven categories: engulfers, shredders, scrapers,
# deposit-feeders, active filter-feeders, passive filter-feeders and piercers, in this order.

# Functional similarities among species are computed as indicated in Pavoine and Ricotta (2014)
s_species_ochiai_feed <- dsimFun(feed, vartype = "P", method=4)

# To check that this matrix of trait-based similarities
# among species is positive semidefinite (p.s.d.)
# we have to verify that its eigenvalues are all nonnegative:
all(eigen(s_species_ochiai_feed)$val>-(1e-8))
```

```
dspca_feed <- dspca(t(macroloire$fau), s_species_ochiai_feed)

# eigenvalue distribution:
barplot(dspca_feed$eig)
# The eigenvalues show strong similarities in the composition of the sites
# regarding the feeding habits of the macroinvertebrates.

# In this data set, communities (sites) are named from S1 to S38 from upstream to downstream
rownames(t(macroloire$fau))

# species are coded from E1 to E40
colnames(t(macroloire$fau))

plot(dspca_feed,  autolab=TRUE,
    colors= list(ifelse(feed[,6]>0.5,"red","black"),
    hcl.colors(38, "cyan")))

# The graphical display shows that most sites are characterized by the dominance of a few species
# (which are mostly passive filter-feeders, in red).
# In a few sites mostly upstream (in blue), the diversity in feeding habits is higher.
}

## End(Not run)
```

EH   *Faith Phylogenetic Diversity*

Description

Function EH computes the sum of branch lengths on a phylogenetic tree.

Usage

```
EH(phyl, select = NULL)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust.
select	a vector containing the numbers of the leaves (species) which must be considered in the computation of Phylogenetic Diversity (PD) (or merely sum of branch lengths on the tree). This argument allows the calculation of PD for a subset of species (including the branch between the subtree and the most ancient node of the full tree).

Value

Function EH returns a real value

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr> with contributions of Stephane Dray

References

Faith, D.P. (1992). Conservation evaluation and phylogenetic diversity. *Biological Conservation*, **61**, 1–10.

Nee, S. and May, R.M. (1997) Extinction and the loss of evolutionary history. *Science*, **278**, 692–694.

See Also

[optimEH](#), [randEH](#)

Examples

```
## Not run:
if(require(ape) && require(adephylo)){
data(carni70, package = "adephylo")
tre <- read.tree(text = carni70$tre)
adiv:::EH(tre)
adiv:::EH(tre, select=c("Mustela.nigripes", "Mustela.frenata", "Puma.concolor"))
adiv:::EH(tre, select=c(1,68,70))
}

## End(Not run)
```

EqRao   *Apportionment of Diversity*

Description

Eight functions are available.

EqRS performs the first decomposition of diversity developed in Pavoine et al. (2016) and rtestEqRS can be used for the associated permutation tests.

EqRSintra performs the second decomposition of diversity developed in Pavoine et al. (2016) and rtestEqRSintra can be used for the associated permutation tests.

EqRao performs the third decomposition of diversity introduced in Pavoine et al. (2016) and rtestEqRao can be used for the associated permutation tests.

wapqe performs the apportionment of quadratic entropy (Rao 1986) and rtestwapqe associated permutation tests (Pavoine et al. 2016).

Usage

```
EqRS(comm, dis = NULL, structures = NULL,
option = c("eq", "normed1", "normed2"),
formula = c("QE", "EDI"), tol = 1e-08)

EqRSintra(comm, dis = NULL, structures = NULL,
option = c("eq", "normed1", "normed2"),
formula = c("QE", "EDI"), tol = 1e-08,
metmean = c("harmonic", "arithmetic"))

EqRao(comm, dis = NULL, structures = NULL,
option = c("eq", "normed1", "normed2"),
formula = c("QE", "EDI"), wopt = c("even", "speciesab"),
tol = 1e-08, metmean = c("harmonic", "arithmetic"))

wapqe(comm, dis = NULL, structures = NULL,
formula = c("QE", "EDI"), wopt = c("even", "speciesab"),
tol = 1e-08)

rtestEqRS(comm, dis = NULL, structures = NULL,
formula = c("QE", "EDI"),
option = c("normed1", "normed2", "eq"),
popt = c("aggregated", "independent"), level = 1, nrep = 99,
alter = c("greater", "less", "two-sided"), tol = 1e-08)

rtestEqRSintra(comm, dis = NULL, structures = NULL,
formula = c("QE", "EDI"),
option = c("normed1", "normed2", "eq"),
popt = c("aggregated", "independent"), level = 1,
nrep = 99, alter = c("greater", "less", "two-sided"),
tol = 1e-08, metmean = c("harmonic", "arithmetic"))

rtestEqRao(comm, dis = NULL, structures = NULL,
formula = c("QE", "EDI"),
option = c("normed1", "normed2", "eq"),
wopt = c("even", "speciesab"),
popt = c("aggregated", "independent"), level = 1, nrep = 99,
alter = c("greater", "less", "two-sided"), tol = 1e-08,
metmean = c("harmonic", "arithmetic"))

rtestwapqe(comm, dis = NULL, structures = NULL,
formula = c("QE", "EDI"), wopt = c("even", "speciesab"),
popt = c("aggregated", "independent"),
level = 1, nrep = 99, alter = c("greater", "less", "two-sided"),
tol = 1e-08)
```

Arguments

comm	a data frame or a matrix with communities as rows and species as columns. Entries are abundances of species within sites.
dis	either NULL or an object of class dist that contains the (functional or phylogenetic) distances among species. If NULL, the Gini-Simpson index is used.
structures	either NULL or a data frame that contains, in the *j*th row and the *k*th column, the name of the group of level *k* to which the *j*th community belongs. Communities in structures should be in the same order as in comm. See details.
option	"eq", "normed1" or "normed2" (if a vector with several codes is given, only the first one is considered). "eq" means that the diversity components are given in terms of equivalent number (E) of species, sites, regions etc. "normed1" means that the normed components of diversity will be returned with formula (1 - 1 / E) / (1 - 1 / Emax); "normed2" means that the normed components of diversity will be returned with formula (E - 1) / (Emax - 1). Emax is the maximum possible value for E. For Ealpha (local diversity) and Egamma (global diversity), Emax=S (the number of species in the data set).
formula	either "QE" (default) or "EDI". See details.
wopt	either "even" (default), "speciesab", or a numeric vector. If w="speciesab", then the communities will be weighted by their sum of species' abundances. If w="even", the communities will be evenly weighted within the factors defined by the argument structures.
popt	a sring: either "aggregated" or "independent". This choice will define the permutation test within communities (level 1 of the permutation scheme). If "aggregated" is chosen, sampled individuals will be considered to be aggregated in space so that. (For example, consider that fish are sampled by net, and fish are aggregated in space, then when using the net, one might have zero abundance once because the net was not placed in a shoal of fish and a very high abundance another time because the net was placed in a shoal of fish; the permutation scheme was developed for this scenario). In that case, the abundance values of each species are permuted between communities. Permutations leading to an empty community (with all species abundances equal to zero) are excluded. If "independent" is chosen, sampled individuals will be considered independent from each other and now abundances are randomly defined with the only constraints that the sum of species abundance within a community is kept constant and the sum of abundance of a species across all communities (if argument structures is NULL or across the communities of a group (of level 2, see argument level) if structures is not NULL. Decimals are not allowed for abundance data with popt = "independent".
tol	a tolerance threshold (a value between -tol and tol is considered equal to zero).
metmean	a string: either "arithmetic" or "harmonic" (default).
level	a number. The number is discarded if the argument structures is set to NULL. If structures is different from NULL then 1 means test for differences among communities, within the levels of the first factor given in argument structures (column 1), 2 means test for differences among levels of the first factor given in argument structures (column 1) but within levels of the second factor given in

argument structures (column 2) (if available), etc. level should thus be between 1 and 1+the number of columns in structures. See details for concrete examples.

nrep   the number of permutations.

alter   a string specifying the alternative hypothesis, must be one of "greater" (default), "less" or "two-sided".

Details

If formula = "QE", the definition of the quadratic entropy is:

$$QE(\mathbf{p}_i, \mathbf{D}) = \sum_{k=1}^{S} \sum_{l=1}^{S} p_{k|i} p_{k|j} d_{kl}$$

where $\mathbf{p}_i = (p_{1|i}, ..., p_{k|i}, ..., p_{S|i})$ is the vector of relative species abundance within community i; S is the number of species; $\mathbf{D} = (d_{kl})$ is the matrix of (phylogenetic or functional) dissimilarities among species, and d_{kl} is the (phylogenetic or functional) dissimilarity between species k and l.

If formula = "EDI", the definition of the quadratic entropy is:

$$EDI(\mathbf{p}_i, \mathbf{D}) = \sum_{k=1}^{S} \sum_{l=1}^{S} p_{k|i} p_{k|j} \frac{d_{kl}^2}{2}$$

EDI stands for the Euclidean Diversity Index of Champely and Chessel (2002). For example, applying EqRao with the distances dis=mydis and formula=QE corresponds to applying it with dis=sqrt(2*mydis) and formula=EDI.

The argument level permits to choose the component of beta diversity that will be tested for significance.

Examples are given below:

Scenario 1: Imagine that structures is NULL, then we only have a set of communities, species within them and a matrix of (functional or phylogenetic) dissimilarities among species. Then there is only one level of beta diversity which represents the average dissimilarities among communities. The functions rtestEqRS, rtestEqRSintra, rtestEqRao, and rtestwapqe will always, in that case, test for the significance of the dissimilarities among communities (whatever the value given to argument level).

Scenario 2: Imagine that structures is a data frame with sites as rows and only one column representing how sites are distributed among regions. Then, there are two levels of beta diversity: beta1 diversity represents the (functional or phylogenetic) dissimilarities among sites within regions; and beta2 diversity represents the (functional or phylogenetic) dissimilarities among regions. If level = 1 then functions rtestEqRS, rtestEqRSintra, rtestEqRao, and rtestwapqe will test for the significance of the dissimilarities among sites within regions (beta1 diversity); in contrast, if level = 2 functions rtestEqRS, rtestEqRSintra, rtestEqRao, and rtestwapqe will test for the significance of the dissimilarities among regions (beta2 diversity). As there is only one column in argument structures and thus only two levels of diversity, level cannot be higher than 2.

Scenario 3: Imagine now that argument comm contains quadrats as rows and species as columns and that structures is a data frame with quadrats as rows and three columns representing how quadrats are organized in three nested factors: hamlets, towns, and counties. Then there would be

four levels of beta diversity: beta1 diversity represents the (functional or phylogenetic) dissimilarities among quadrats within hamlets, towns, and counties; beta2 diversity represents the dissimilarities among hamlets within towns and counties; beta3 diversity represents the dissimilarities among towns within counties; and beta4 diversity represents the dissimilarities among counties. If `level` = 1 then functions `rtestEqRS`, `rtestEqRSintra`, `rtestEqRao`, and `rtestwapqe` will test for the significance of the dissimilarities among quadrats within hamlets, towns, and counties (beta1 diversity); if `level` = 2 functions `rtestEqRS`, `rtestEqRSintra`, `rtestEqRao`, and `rtestwapqe` will test for the significance of the dissimilarities among hamlets within towns and regions (beta2 diversity); if `level` = 3 functions `rtestEqRS`, `rtestEqRSintra`, `rtestEqRao`, and `rtestwapqe` will test for the significance of the dissimilarities among towns within regions (beta3 diversity); if `level` = 4 functions `rtestEqRS`, `rtestEqRSintra`, `rtestEqRao`, and `rtestwapqe` will test for the significance of the dissimilarities among regions (beta4 diversity). As there are only three columns in argument `structures` and thus only four levels of diversity, `level` cannot be higher than 4. Test for `level=1` is performed by permuting the abundances of each species across quadrats but within hamlets, towns, and counties. Test for `level=2` is performed by permuting the quadrats among hamlets within towns and counties. Test for `level=3` is performed by permuting the hamlets among towns within counties. Test for `level=4` is performed by permuting the towns among counties.

Other permutation schemes will be added in the future.

Value

The functions `EqRS`, `EqRSintra`, `EqRao`, `wapqe` return a data frame with each component of the selected diversity decomposition. The functions `rtestEqRS`, `rtestEqRSintra`, `rtestEqRao`, and `rtestwapqe` return a list of class `randtest` or `krandtest` (classes of package ade4).

Author(s)

Sandrine Pavoine `<sandrine.pavoine@mnhn.fr>`

References

Pavoine, S., Marcon, E., Ricotta, C. (2016) "Equivalent numbers" for species, phylogenetic, or functional diversity in a nested hierarchy of multiple scales. *Methods in Ecology and Evolution*, **7**, 1152–1163.

Champely, S. and Chessel, D. (2002) Measuring biological diversity using Euclideanmetrics. *Environmental and Ecological Statistics*, **9**, 167–177.

Rao, C.R. (1986) Rao's axiomatization of diversity measures. In: *Encyclopedia of Statistical Sciences*. Vol. 7 (eds S. Kotz and N.L. Johnson), pp. 614–617. New York: Wiley and Sons.

Examples

```
## Not run:
if(require(ade4)){

data(macroloire, package="ade4")

# Taxonomic dissimilarities among species:
dTaxo <- dist.taxo(macroloire$taxo)^2/2
dTaxo <- dTaxo/max(dTaxo)
```

```
# Size-based dissimilarities among species
dSize <- dist.prop(macroloire$traits[ ,1:4], method = 2)
# Dissimilarities among species in terms of feeding categories
dFeed <- dist.prop(macroloire$traits[ ,5:11], method = 2)
# Dissimilarities among species in terms of
# both size and feeding categories
dSF <- (dSize+dFeed)/2

# Table with sites as rows (stations),
# species as columns and abundances as entries
ab <- as.data.frame(t(macroloire$fau))
# Table with sites as rows and one column only.
# Entries indicate the geological region associated
# with each site
stru <- macroloire$envir["Morphoregion"]

EqRao(ab, , stru, option="eq")
EqRao(ab, dTaxo, stru, formula = "QE", option="eq")
EqRao(ab, dSize, stru, formula = "QE", option="eq")
EqRao(ab, dFeed, stru, formula = "QE", option="eq")
EqRao(ab, dSF, stru, formula = "QE", option="eq")

EqRao(ab, , stru, option="normed2")
EqRao(ab, dTaxo, stru, formula = "QE", option="normed2")
EqRao(ab, dSize, stru, formula = "QE", option="normed2")
EqRao(ab, dFeed, stru, formula = "QE", option="normed2")
EqRao(ab, dSF, stru, formula = "QE", option="normed2")

# Tests for dissimilarities among sites within regions:
### TIME CONSUMING

rb1_GS <- rtestEqRao(ab, , stru, level=1, nrep=999, option="normed2")
rb1_GS
plot(rb1_GS)
rb1_Taxo <- rtestEqRao(ab, dTaxo, stru, formula = "QE",
level=1, nrep=999, option="normed2")
rb1_Taxo
plot(rb1_Taxo)
rb1_Size <- rtestEqRao(ab, dSize, stru, formula = "QE",
level=1, nrep=999, option="normed2")
rb1_Size
plot(rb1_Size)
rb1_Feed <- rtestEqRao(ab, dFeed, stru, formula = "QE",
level=1, nrep=999, option="normed2")
rb1_Feed
plot(rb1_Feed)
rb1_SF <- rtestEqRao(ab, dSF, stru, formula = "QE",
level=1, nrep=999, option="normed2")
rb1_SF
plot(rb1_SF)

# Tests for dissimilarities among regions:
### TIME CONSUMING
```

```
r2_GS <- rtestEqRao(ab, , stru, level=2, nrep=999, option="normed2")
r2_GS
plot(r2_GS)
r2_Taxo <- rtestEqRao(ab, dTaxo, stru, formula = "QE",
level=2, nrep=999, option="normed2")
r2_Taxo
plot(r2_Taxo)
r2_Size <- rtestEqRao(ab, dSize, stru, formula = "QE",
level=2, nrep=999, option="normed2", w="even")
r2_Size
plot(r2_Size)
r2_Feed <- rtestEqRao(ab, dFeed, stru, formula = "QE",
level=2, nrep=999, option="normed2", w="even")
r2_Feed
plot(r2_Feed)
r2_SF <- rtestEqRao(ab, dSF, stru, formula = "QE",
level=2, nrep=999, option="normed2", w="even")
r2_SF
plot(r2_SF)
}

## End(Not run)
```

---

eveparam   *Parametric Indices of Species Evenness*

Description

The function eveparam calculates parametric evenness indices. The parameter controls the relative importance given to rare versus abundant species in a community. The function plot.eveparam plots the results of function eveparam.

Usage

```
eveparam(comm, method = c("hill", "tsallis", "renyi"), q = 2, tol = 1e-08)

## S3 method for class 'eveparam'
plot(x, legend = TRUE,
legendposi = "topright", axisLABEL = "Evenness", type = "b",
col = if (is.numeric(x)) NULL
else sample(colors(distinct = TRUE), nrow(x$eve)),
lty = if (is.numeric(x)) NULL else rep(1, nrow(x$eve)),
pch = if (is.numeric(x)) NULL else rep(19, nrow(x$eve)), ...)
```

Arguments

comm   a data frame or a matrix typically with communities as rows, species as columns
                and abundance as entry.

method	a string: either "hill" for the Hill numbers (Hill 1973), "tsallis" for the Tsallis or HCDT entropy (Harvda and Charvat 1967; Daroczy 1970; Tsallis 1988), or "renyi" for Renyi's entropy (Renyi 1960). These indices are divided by the value they would have if all species had even abundances.
q	a positive numeric or a vector of positive numerics that give values for the q parameter.
tol	numeric tolerance threshold: values between -tol and tol are considered equal to zero.
x	an object of class eveparam obtained with function eveparam.
legend	a logical. If TRUE a legend is given with the colour, the type of line (etc.) used to define the evenness curve of each community.
legendposi	a string or a numeric that gives the position of the legend to be passed to function legend of the base of R.
axisLABEL	a string to display on the main axis of the plot to designate what we are measuring. The default is "Evenness".
type	a string to be passed to the graphic argument type of functions plot and lines used to draw the evenness curve of each community.
col	colours to be passed to the graphic argument col of functions plot and lines to define the colour of the evenness curve of each community.
lty	type of line (plain, broken etc.) to be passed to the graphic argument lty of functions plot and lines used to draw the evenness curve of each community.
pch	type of point (open circle, close circle, square etc.) to be passed to the graphic argument pch of functions plot and lines used to draw the evenness curve of each community.
...	other arguments can be added and passed to the functions plot and lines used to draw the graphic.

Value

Function eveparam, if only one value of q is given, returns a vector with the evenness in the communities. If more than one value of q is given, a list of two objects is returned:

q	the vector of values for q;
div	a data frame with the evenness in the communities calculated for all values of q.

The function plot.eveparam returns a graphic.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Daroczy, Z. (1970) Generalized information functions. *Information and Control*, **16**, 36–51.

Havrda, M., Charvat, F. (1967) Quantification method of classification processes: concept of structural alpha-entropy. *Kybernatica*, **3**, 30–35.

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

Renyi, A. (1960) On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, **1**, 547–561.

Tsallis, C. (1988) Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, **52**, 480–487.

Examples

```
data(batcomm)
ab <- batcomm$ab
plot(eveparam(ab))
plot(eveparam(ab, q=0:4))
```

---

evoCA   *Evolutionary Correspondence Analysis*

Description

The function evoCA performs the evolutionary correspondence analysis (evoCA) (Pavoine 2016): an adaptation of the correspondence analysis (CA) to analyse the distributions of lineages among sites and, simultaneously, to analyse the phylogenetic composition of sites. The function plot.evoCA displays the phylogeny on the factorial maps of evoCA.

Usage

```
evoCA(phyl, comm, scannf = TRUE, nf = 2, abundance = TRUE)

## S3 method for class 'evoCA'
plot(x, xaxis = 1, yaxis = 2, graph = FALSE, ...)
```

Arguments

phyl   an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust. To ease the interpretation of factorial maps, it is advised to use a rooted phylogenetic tree and to add labels to the nodes of the phylogeny. For example, If you phylogeny is named 'tree' and is of class phylo, you can root the phylogeny using tree$root.edge <- 0 and you can name the nodes with the following command: tree$node.label <- paste("n", 1:tree$Nnode, sep=""). Use plot(tree, show.node.label=TRUE) to see the result. If it is rooted and of class phylo4, you can use: nodeLabels(tree) <- paste("n", 1:nNodes(tree), sep="").

comm   a data frame or a matrix typically with communities (or sites, plots, etc.) as rows, species as columns and presence/absence (1/0) or an index of abundance as entries. Species should be labeled as in the phylogenetic tree where they are the tips.

scannf	a logical value indicating whether the screeplot (eigenvalues) should be displayed for choosing the number of axes to be kept.
nf	if scannf is FALSE, an integer indicating the number of axes to be kept.
abundance	a logical value, if FALSE, only the presences/absences of the **evolutionary units** in sites are considered.
x	an object of class evoCA
xaxis	the number of the evoCA axis chosen for the x-axis of the 3d plot.
yaxis	the number of the evoCA axis chosen for the y-axis of the 3d plot.
graph	a logical value. plot.evoCA depends on package rgl. This package will only work if you have graphics support on your computer. If you do have graphics support, choose graph = TRUE and you will be able to visualize the 3d plot with coordinates for species and phylogenetic nodes.
...	other arguments can be added and passed to the function plot3d of package rgl.

Value

evoCA returns an object of class evoCA and of class dudi (see package ade4, ?dudi). Graphical tools are associated with class dudi in packages ade4 and adegraphics (see example section below and ?scatter.dudi).

The returned object contains the following components:

tab	a data frame with n rows and p columns, with communities as rows and nodes of the phylogeny as columns; the entries of the data frame evaluate the degree of dependence (values that depart from zero)/independence(close-to-zero values) between the occurrence in a community and the position in the phylogeny;
cw	weights attributed to the nodes of the phylogeny, a vector with p components;
lw	weights attributed to the communities, a vector with n components;
eig	vector of eigenvalues;
rank	integer, number of axes;
nf	integer, number of kept axes;
c1	normed scores for the nodes of the phylogeny, data frame with p rows and nf columns;
l1	normed scores for the communities, data frame with n rows and nf columns;
co	scores for the nodes of the phylogeny, data frame with p rows and nf columns;
li	scores for the communities, data frame with n rows and nf columns;
call	the original call.

If X is an object of class evoCA, then attributes(X)$phy contains the phylogenetic tree (of class phylo) with names for internal nodes.

plot.evoCA returns a dynamics 3-dimensional plot

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

See Also

evoNSCA, evopcachord, evopcahellinger, evodiss

Examples

```
## Not run:
if(require(ade4) && require(ape) && require(adegraphics)){
O <- adegpar()$plabels$optim
adegpar("plabels.optim" = TRUE)

data(batcomm)
ab <- batcomm$ab
phy <- read.tree(text=batcomm$tre)
plot(phy, show.node=TRUE)
evoCAbat <- evoCA(phy, ab, scan=FALSE, nf=2)
evoCAbat$eig/sum(evoCAbat$eig)
s.label(evoCAbat$li)
s.label(evoCAbat$co)
s.arrow(evoCAbat$co)
inertia.dudi(evoCAbat, row=TRUE)$row.abs
inertia.dudi(evoCAbat, col=TRUE)$col.abs

evoCAbat <- evoCA(phy, ab, scan=FALSE, nf=3) ## All axes are now retained
# The Euclidean (canonical) distances among habitat points on the evoCA space is
dist(evoCAbat$li)
# which is equal to evoDchi2:
evodiss(phy, ab, "chi2")

# Run the following commands only
# if you have graphics support to use
# the rgl package on your computer:
plot(evoCAbat, graph = TRUE)
# With argument "graph = TRUE"
# you will be able to vizualize the 3d plot.

# See also,
plot(evoCAbat, xaxis=1, yaxis=3, graph = TRUE)
adegpar("plabels.optim" = O)

}

## End(Not run)
```

---

evodiss	*Pair-wise Phylogenetic Dissimilarities between Communities*

Description

The function calculates PD-dissimilarity indices described and/or discussed in Pavoine (2016). Part of them are parametric indices.

Usage

```
evodiss(phyl, comm, method = NULL, q = NULL,
w = c("evoab", "even", "speciesab"), diag = FALSE,
upper = FALSE, tol = 1e-08)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust.
comm	a data frame typically with communities as rows, species as columns and an index of abundance as entries. Species should be labeled as in the phylogenetic tree where they are the tips.
method	one of the following string codes: "Minkowski", "Euclidean", "Manhattan", "Chord", "ScaledCanberra", "Divergence", "BC", "MH", "LG", "Hellinger", "chi2", "Hill", "Renyi", "C", "U", "S". See Details.
q	a vector with nonnegative value(s) for parameter q only used with method="Minkowski", "Hill", "Renyi" (parametric indices). q controls the importance given to rare compared to abundant evolutionary units. See Details.
w	either a numeric vector giving weights for communities (same order as in comm), or a string among "even", "evoab", and "speciesab". If several strings are given, only the first one is used. w is used only if method="Hill", "Renyi", "C", "U", or "S" is chosen. See details.
diag	logical argument passed to function as.dist (R base).
upper	logical argument passed to function as.dist (R base).
tol	numeric tolerance threshold: values between -tol and tol are considered equal to zero.

Details

The indices available are (formulas can be found in Supplementary material Appendix 1 of Pavoine 2016):

"Minkowski": $^{q}evoD_{Minkowski}$

"Euclidean": $evoD_{Euclidean}$

"Manhattan": $evoD_{Manhattan}$

Chord: $evoD_{Chord}$

ScaledCanberra: $evoD_{ScaledCanberra}$

Divergence: $evoD_{Divergence}$

BC: $evoD_{Bray-Curtis}$

MH: $evoD_{Morisita-Horn}$

LG: $evoD_{Profile}$

Hellinger: $evoD_{Hellinger}$

chi2: $evoD_{Chi2}$

Hill: $1 - \bar{V}_{q2}$

Renyi: $^{q}evoD_{Renyi}$

C: $1 - \bar{C}_{q2}$

U: $1 - \bar{U}_{q2}$

S: $1 - \bar{S}_{q2}$

The weights of the communities (argument w) can be ″even″ (even weights, i.e. relative abundances are considered for evolutionary units), ″evoab″ (proportional to the summed abundances of all evolutionary units, i.e. absolute abundances are considered for evolutionary units), or ″speciesab″ (proportional to the summed abundances of all species). Note that if the phylogenetic tree is ultrametric (the distance from any species to the root is constant), then options ″evoab″ and ″speciesab″ are equivalent.

Value

An object of class dist containing the PD-dissimilarities (phylogenetic dissimilarities) between communities.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

The methodologies are presented in

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

They gather, in a common framework, indices introduced in

Chiu, C.-H., Jost, L., Chao, A. (2014) Phylogenetic beta diversity, similarity and differentiation measures based on Hill numbers. *Ecological Monographs*, **84**, 21–44.

and earlier work extended here in a phylogenetic context and reviewed in

Legendre, P. and De Caceres, M. (2013) Beta diversity as the variance of community data: dissimilarity coefficients and partitioning. *Ecology Letters*, **16**, 951–963.

See Also

evodiss_family

Examples

```
## Not run:
if(require(ape)){

data(batcomm)
phy <- read.tree(text=batcomm$tre)
ab <- batcomm$ab[,phy$tip.label]

evodiss(phy, ab, "LG")
evodiss(phy, ab, "Hellinger")
evodiss(phy, ab, "Chord")

evodiss(phy, ab, "Hill", q=2)
evodiss(phy, ab, "Hill", q=2, w="even")

}

## End(Not run)
```

---

evodiss_family	*A Family of Indices Dedicated to Pair-wise Phylogenetic Dissimilarities between Communities*

Description

The function evodiss_family was written thanks to function dist.binary of package ade4. Function dist.binary calculates specific compositional distances. The new function here replaces species with evolutionary units. It calculates Nipperess et al. (2010) parameters a, b, c, d (with incidence data), or A, B, C, D (with abundance data) and then use these parameters to compute pair-wise phylogenetic dissimilarities between communities.

The graphical function evodiss_ternaryplot displays Nipperess et al. (2010) parameters a, b, c (with incidence data), or A, B, C (with abundance data) on a ternary plot (see Koleff et al. 2003).

Usage

```
evodiss_family(phyl, comm, method = NULL, abundance = TRUE,
squareroot = TRUE, diag = FALSE, upper = FALSE, tol = 1e-08)

evodiss_ternaryplot(phyl, comm, abundance = TRUE,
tol = 1e-08, ...)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust.
comm	a data frame typically with communities as rows, species as columns and presence/absence or an index of abundance as entries. Species should be labeled as in the phylogenetic tree where they are the tips.

method	either NULL or a number between 1 and 14. If NULL, the choice is made with a console message. See details.
abundance	a logical indicating whether abundance data (if TRUE) or incidence data (presence/absence if FALSE) must be used.
squareroot	a logical. First a similarity index (S) is calculated (see details). Then if squareroot=TRUE, the dissimilarity is calculated as $\sqrt{1-S}$. If squareroot=FALSE, it is calculated as 1-S. The square root is useful to obtain Euclidean properties for the PD-dissimilarity indices (with methods 1 to 10).
diag	logical argument passed to function as.dist (R base).
upper	logical argument passed to function as.dist (R base).
tol	numeric tolerance threshold: values between -tol and tol are considered equal to zero.
...	other arguments can be added and passed to the function triangle.plot of the ade4 package (labels cannot be changed and are defined thanks to the row names of comm).

Details

The function was written thanks to function dist.binary of package ade4. Function dist.binary calculates specific compositional distances. The new function here replaces species with evolutionary units and adds several indices. It calculates Nipperess et al. (2010) parameters a, b, c, d (with incidence data), or A, B, C, D (with abundance data). Then, the parameters are combined thanks to one out of 14 methods as defined below:

method = 1: Jaccard index (1901); S3 coefficient of Gower and Legendre (1986) = a / (a+b+c).

method = 2: Simple matching coefficient of Sokal and Michener (1958); S4 coefficient of Gower and Legendre (1986) = (a+d) / (a+b+c+d).

method = 3: Sokal and Sneath(1963); S5 coefficient of Gower and Legendre (1986) = a / (a + 2(b + c)).

method = 4: Rogers and Tanimoto (1960); S6 coefficient of Gower and Legendre (1986) = (a + d) / (a + 2(b + c) +d).

method = 5: Dice (1945) or Sorensen (1948); S7 coefficient of Gower and Legendre (1986) = 2a / (2a + b + c).

method = 6: Hamann coefficient; S9 index of Gower and Legendre (1986) = (a - (b + c) + d) / (a + b + c + d).

method = 7: Ochiai (1957); S12 coefficient of Gower and Legendre (1986) = a / sqrt((a + b)(a + c)).

method = 8: Sokal and Sneath (1963); S13 coefficient of Gower and Legendre (1986) = ad / sqrt((a + b)(a + c)(d + b)(d + c)).

method = 9: Phi of Pearson; S14 coefficient of Gower and Legendre (1986) = (ad - bc) / sqrt((a + b)(a + c)(d + b)(d + c)).

method = 10: S2 coefficient of Gower and Legendre (1986) = a / (a + b + c + d) (imposed unit self-similarity).

method = 11: Kulczynski index; S10 coefficient of Gower and Legendre (1986) = 0.5 * (a/(a+b) + a/(a+c))

method = 12: S11 coefficient of Gower and Legendre (1986) = 0.25 * (a/(a+b) + a/(a+c) + d/(b+d) + d/(c+d))

method = 13: S8 coefficient of Gower and Legendre (1986) = (a+d)/(a+0.5*(b+c)+d)

method = 14: Simpson coefficient = a/(a+min(b,c))

Value

Function `evodiss_family` returns an object of class `dist` containing the PD-dissimilarities (phylogenetic dissimilarities) between communities.

Function `evodiss_ternaryplot` returns a graph.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

The methodologies are presented in Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

They gather in a common framework and extend earlier work introduced in Koleff, P., Gaston, K.J., Lennon, J.J. (2003) Measuring beta diversity for presence-absence data. *Journal of Animal Ecology*, **72**, 367–382.

Nipperess, D.A., Faith, D.P., Barton, K.(2010) Resemblance in phylogenetic diversity among ecological assemblages. *Journal of Vegetation Science*, **21**, 809–820.

Dice, L.R. (1945) Measures of the amount of ecologic association between species. *Ecology*, **26**, 297–302.

Gower, J.C., Legendre, P. (1986) Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification*, **3**, 5–48.

Jaccard, P. (1901) Etude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, **37**, 547–579.

Ochiai, A. (1957) Zoogeographic studies on the soleoid fishes found in Japan and its neighbouring regions. *Bulletin of the Japanese Society of Scientific Fisheries*, **22**, 526–530.

Rogers, J.S. and Tanimoto, T.T. (1960) A computer program for classifying plants. *Science*, **132**, 1115–1118.

Sokal, R.R. and Michener, C.D. (1958) A Statistical Method for Evaluating Systematic Relationships. *The University of Kansas Science Bulletin*, **38**, 1409–1438.

Sokal, R.R. and Sneath, P.H.A. (1963) *Principles of numerical taxonomy*. San Francisco: W. H. Freeman.

Sorensen, T. (1948) A method of establishing groups of equal amplitude in plant sociology based on similarity of species content. *Kongelige Danske Videnskabernes Selskabs Biologiske Skrifter*, **5**, 1–34.

See Also

evodiss

Examples

```
## Not run:
if(require(ape)){

data(batcomm)
phy <- read.tree(text=batcomm$tre)
ab <- batcomm$ab[,phy$tip.label]

# PD-dissimilarity indices that use Nipperess et al. (2010)
# parameters can be obtained thanks to function evodiss_family.
# For example, with incidence data,
# indices evoDJaccard, evoDSorensen, and evoDOchiai
# (supplementary Appendix 1 in Pavoine 2016)
# can be obtained as follows:

evodiss_family(phy, ab, method=1, abundance=FALSE) # Jaccard
evodiss_family(phy, ab, method=5, abundance=FALSE) # Sorensen
evodiss_family(phy, ab, method=7, abundance=FALSE) # Ochiai

# With abundance data, indices evoDTJ, evoDTS, evoDTO
# (supplementary Appendix 1 in Pavoine 2016)
# can be obtained as follows:

evodiss_family(phy, ab, method=1) # evoDTJ
evodiss_family(phy, ab, method=5) # evoDTS
evodiss_family(phy, ab, method=7) # evoDTO

# Ternary plots can be obtained for Nipperess et al. (2010)
# parameters a, b, c (incidence data)
# (see Supplementary material Appendix 4 in Pavoine 2016):

evodiss_ternaryplot(phy, ab, abundance = FALSE)

# and for Nipperess et al. (2010) parameters A, B, C
# (abundance data):

evodiss_ternaryplot(phy, ab, abundance = TRUE)

# The ternary plots can be adjusted thanks
# to the arguments of function triangle.plot (package ade4).
# For example, full triangles can be obtained as follows
# (previous graphs were zoomed on the smallest principal
# equilateral triangle that contained the points,
# as indicated by the embedded close grey triangle
# at the left-hand corner of ternary plot given above):

evodiss_ternaryplot(phy, ab, abundance = FALSE, adjust=FALSE, showposition=FALSE)
# Incidence data

evodiss_ternaryplot(phy, ab, abundance = TRUE, adjust=FALSE, showposition=FALSE)
# abundance data
```

```
    }

    ## End(Not run)
```

---

evodiv   *Indices of Phylogenetic Diversity*

Description

The function evodiv calculates diversity indices that rely on the relative or absolute abundance of features on a phylogenetic tree, with the assumption that the number of features on a given branch of a phylogenetic tree is equal to the length of this branch (see Pavoine 2016).

Usage

```
evodiv(phyl, comm, method = "full", tol = 1e-8)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase), or hclust.
comm	a data frame or a matrix typically with communities as rows, species as columns and an index of abundance as entries. Species should be labeled as in the phylogenetic tree where they are the tips. Note that with presence/absence (0/1) data, only feature richness will be calculated correctly.
method	a string or a vector of strings: one or several of "richness", "GiniSimpson", "Simpson", "Shannon", "Margalef", "Menhinick", "McIntosh", "full". See details.
tol	a tolerance threshold (a value between -tol and tol is considered equal to zero)

Details

Let L_b the length of branch b in the phylogenetic tree. Let $a_b i$ the absolute abundance of branch b in community i (sum of abundance of all species descending from it in the phylogenetic tree). $p_{bi} = a_{bi} / \sum_b L_b a_{bi}$ is the relative abundance of branch b in community i. If method="richness", the diversity index is the number of features (branch units): $\sum_b L_b$. It corresponds to Faith (1992) Phylogenetic Diversity index. If method="GiniSimpson", the diversity index is that of Gini (1912) and Simpson (1949): $1 - \sum_b L_b p_{bi}^2$. If method="Simpson", the diversity index is that of Simpson (1949): $1/\sum_b L_b p_{bi}^2$. If method="Shannon", the diversity index is that of Shannon (1948) with neperian logarithm: $-\sum_b L_b p_{bi} log(p_{bi})$. If method="Margalef", the diversity index is that of Margalef (1972): $(\sum_b L_b - 1)/log(\sum_b L_b a_{bi})$. If method="Menhinick", the diversity index is that of Menhinick (1964): $\sum_b L_b/\sqrt{sum_b L_b a_b i}$. If method="McIntosh", the diversity index is that of McIntosh (1967): $(\sum_b L_b a_{bi} - \sqrt{\sum_b L_b a_{bi}^2})/(\sum_b L_b a_{bi} - \sqrt{\sum_b L_b a_{bi}})$. If one of the strings is "full", then all indices are calculated.

For the indices relying on relative abundances to be valid, each species must support at least one feature. If this is not the case and one of these indices has to be calculated, the phylogenetic

tree is re-scaled so that the shortest distance from tip to root is equal to 1. This means that the scale of the branch lengths will be changed, which will impact the calculation of feature richness (method="richness") and the indices of Margalef, Menhinick and McIntosh.

In the rare cases where this scaling will be necessary, a better option is that the scaling be done by the user itself (prior to the use of the evodiv function). Indeed, this will unable them to choose an appropriate scaling. For example, if branch lengths are expressed as billion years of evolution and the sum of branch lengths from tip to root is lower than one billion. Then expressing branch lengths in million years of evolution may solve the problem, leading to sum of branch lengths from tip to root higher than 1 million years.

If the phylogenetic tree has no branch lengths, all branches are set to a length of 1.

Value

Function evodiv returns a matrix with communities as rows and the diversity indices as columns.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Gini, C. (1912) Variabilita e mutabilita. Studi economicoaguridici delle facoltta di giurizprudenza dell, Universite di Cagliari III, Parte II.

Magurran, A.E. (2004) Measuring biological diversity. Blackwell Publishing, Oxford, U.K.

Margalef, R. (1972) Homage to Evelyn Hutchinson, or why is there an upper limit to diversity? *Transactions of the Connecticut Academy of Arts and Sciences*, **44**, 211–235.

McIntosh, R.P. (1967) An index of diversity and the relation of certain conepts to diversity. *Ecology*, **48**, 392–404.

Menhinick, E.F. (1964) A Comparison of Some Species-Individuals Diversity Indices Applied to Samples ofField Insects. *Ecology*, **45**, 859–861.

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

Pavoine, S., Ricotta, C. (2019) A simple translation from indices of species diversity to indices of phylogenetic diversity. *Ecological Indicators*, **101**, 552–561.

Shannon, C.E. (1948) A mathematical theory of communication. *Bell System technical journal*, **27**, 379–423, 623–656.

Simpson, E.H. (1949) Measurement of diversity. *Nature*, **163**, 688.

See Also

[evodivparam](), [speciesdiv]()

Examples

```
## Not run:
if(require(ape)){
data(batcomm)
phy <- read.tree(text=batcomm$tre)
ab <- batcomm$ab[,phy$tip.label]
evodiv(phy, ab)
}

## End(Not run)
```

---

evodivparam   *Parametric Indices of Phylogenetic Diversity*

Description

Function `evodivparam` calculates phylogenetic diversity in communities using parametric indices derived from Tsallis and Hill compositional indices. It can also be applied to functional trees rather than phylogenies, to calculate a functional diversity. The function `plot.evodivparam` plots the results of function `evodivparam`.

Usage

```
evodivparam(phyl, comm,
    method = c("hill", "tsallis", "renyi"),
    q = 2, tol = 1e-08)

## S3 method for class 'evodivparam'
plot(x, legend = TRUE,
    legendposi = "topright", axisLABEL = "Tree-based diversity",
    type="b", col = if(is.numeric(x)) NULL
    else sample(colors(distinct = TRUE), nrow(x$div)),
    lty = if(is.numeric(x)) NULL else rep(1, nrow(x$div)),
    pch = if(is.numeric(x)) NULL else rep(19, nrow(x$div)),
    ...)
```

Arguments

phyl	an object inheriting the class `phylo` (see package ape), `phylo4` (see package phylobase), or `hclust`.
comm	a data frame or a matrix typically with communities as rows, species as columns and abundance as entry. Species should be labeled as in the phylogenetic tree where they are the tips.
method	a string: either "hill" for the Hill numbers (Hill 1973), "tsallis" for the Tsallis or HCDT entropy (Harvda and Charvat 1967; Daroczy 1970; Tsallis 1988), or "renyi" for Renyi's entropy (Renyi 1960). If several values are given, only the first one is considered. See details.

q   a vector with nonnegative value(s) for parameter q. See details.

tol   numeric tolerance threshold: values between -tol and tol are considered equal
                                    to zero.

x   an object of class evodivparam obtained with function evodivparam.

legend   a logical. If TRUE a legend is given with the colour, the type of line (etc.) used
                                    to define the diversity curve of each community.

legendposi   a string that gives the position of the legend to be passed to function legend of
                                    the base of R.

axisLABEL   a string to display on the main axis of the plot to designate what we are measur-
                                    ing. The default is "Tree-based diversity".

type   a string to be passed to the graphic argument type of functions plot and lines
                                    used to draw the diversity curve of each community.

col   vector of colours to be passed to the graphic argument col of functions plot
                                    and lines to define the colour of the diversity curve of each community.

lty   vector of type of line (plain, broken etc.) to be passed to the graphic argument
                                    lty of functions plot and lines used to draw the diversity curve of each com-
                                    munity.

pch   type of point (open circle, close circle, square etc.) to be passed to the graphic
                                    argument pch of functions plot and lines used to draw the diversity level of
                                    each community.

...   other arguments can be added and passed to the functions plot and lines used
                                    to draw the graphic.

Details

Consider a phylogenetic tree T, b_T the set of branches in T, k a branch, L_k the length of branch
k, j a community (j=1,...,m), a_{jk} the abundance associated with branch k in community j (sum of
abundance of all species descending from the branch). q is the parameter that increases with the
importance given to abundant species compared to rare species in diversity.

The methods available to calculate the phylogenetic diversity in community j are: tsallis:

$$^q evoTsallis_j = \left[1 - \sum_{k \in b_T} L_k \left(\frac{a_{jk}}{\sum_{k \in b_T} L_k a_{jk}} \right)^q \right] / (q-1)$$

hill:

$$^q evoHill_j = \left[\sum_{k \in b_T} L_k \left(\frac{a_{jk}}{\sum_{k \in b_T} L_k a_{jk}} \right)^q \right]^{1/(1-q)}$$

renyi:

$$^q evoRenyi_j = log(^q evoHill_j)$$

Value

If only one value of q is given, a vector with the phylogenetic diversity of each community is returned. If more than one value of q is given, a list of two objects is returned:

q   the vector of values for q;

div   a data frame with the phylogenetic diversity of each community calculated for all values of q.

The function `plot.evodivparam` returns a graphic.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

The methodologies and scripts were developed by

Pavoine, S., Ricotta, C. (2019) A simple translation from indices of species diversity to indices of phylogenetic diversity. *Ecological Indicators*, **101**, 552–561.

using earlier work by:

Chao, A., Chiu, C.-H., Jost, L. (2010) Phylogenetic diversity measures based on Hill numbers. *Philosophical Transactions of the Royal Society London Series B*, **365**, 3599–3609.

Daroczy, Z. (1970) Generalized information functions. *Information and Control*, **16**, 36–51.

Havrda, M., Charvat F. (1967) Quantification method of classification processes: concept of structural alpha- entropy. *Kybernetik*, **3**, 30–35.

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

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

Renyi, A. (1960) On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, **1**, 547–561.

Tsallis, C. (1988) Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, **52**, 480–487.

See Also

abgevodivparam, divparam

Examples

```
## Not run:
if(require(ape)){

data(batcomm)
phy <- read.tree(text=batcomm$tre)
ab <- batcomm$ab[, phy$tip.label]
plot(evodivparam(phy, ab))
```

```
plot(evodivparam(phy, ab, q=seq(0, 10, length=20)))

}

## End(Not run)
```

---

evoeveparam   *Parametric Indices of Phylogenetic evenness*

Description

Function evoeveparam calculates phylogenetic evenness (evenness in features, which are branch units of a phylogenetic tree) in communities. It uses parametric indices derived from Tsallis and Hill compositional indices, and named qfeveHCDT, qfeveHill, and qfeveRenyi in Pavoine and Ricotta (2019). evoeveparam can also be applied to functional trees rather than phylogenies, to calculate a functional evenness. The function plot.evoeveparam plots the results of function evoeveparam.

Usage

```
evoeveparam(phyl, comm, method = c("hill", "tsallis", "renyi"),
    q = 2, option = 1:3, H = NULL, tol = 1e-8)

## S3 method for class 'evoeveparam'
plot(x, legend = TRUE,
    legendposi = "topright", axisLABEL = "Tree-based evenness",
    type="b", col = if(is.numeric(x)) NULL
    else sample(colors(distinct = TRUE), nrow(x$eve)),
    lty = if(is.numeric(x)) NULL else rep(1, nrow(x$eve)),
    pch = if(is.numeric(x)) NULL else rep(19, nrow(x$eve)),
    ...)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase), or hclust.
comm	a data frame or a matrix typically with communities as rows, species as columns and abundance as entry. Species should be labeled as in the phylogenetic tree (object phyl) where they are the tips.
method	a string: either "hill" for qfeveHill using the Hill numbers (Hill 1973), "tsallis" for qfeveHCDT using the Tsallis or HCDT entropy (Harvda and Charvat 1967; Daroczy 1970; Tsallis 1988), or "renyi" for qfeveRenyi using Renyi's entropy (Renyi 1960). If several values are given, only the first one is considered. See details.
q	a vector with nonnegative value(s) for parameter q of functions qfeveHCDT, qfeveHill, and qfeveRenyi.

option	an integer (either 1, 2 or 3). If 1, the (Hill, Tsallis or Renyi) diversity index is divided by the value it would have if species had same abundance and were independent, if 2, the diversity is divided by the value it would have if species had same abundance, were independent, and at the maximum observed distance from tip to root, if 3, the diversity is divided by the value it would have if species had same abundance, were independent, and at a distance equal to H from the root of the tree. Options 1 and 2 are equivalent in case of an ultrametric tree.
H	a numeric; H must be higher than the largest observed distance from tip to root.
tol	numeric tolerance threshold: values between -tol and tol are considered equal to zero.
x	an object of class evoeveparam obtained with function evoeveparam.
legend	a logical. If TRUE a legend is given with the colour, the type of line (etc.) used to define the evenness curve of each community.
legendposi	a string that gives the position of the legend to be passed to function legend of the base of R.
axisLABEL	a string to display on the main axis of the plot to designate what we are measuring. The default is "Tree-based evenness".
type	a string to be passed to the graphic argument type of functions plot and lines used to draw the evenness curve of each community.
col	vector of colours to be passed to the graphic argument col of functions plot and lines to define the colour of the evenness curve of each community.
lty	vector of type of line (plain, broken etc.) to be passed to the graphic argument lty of functions plot and lines used to draw the evenness curve of each community.
pch	type of point (open circle, close circle, square etc.) to be passed to the graphic argument pch of functions plot and lines used to draw the evenness level of each community.
...	other arguments can be added and passed to the functions plot and lines used to draw the graphic.

Details

Function evoeveparam calculates feature evenness (features = branch units on a phylogenetic [or functional] tree) using parametric indices qfeveHCDT (with method=tsallis), qfeveHill (with method=hill), qfeveRenyi (with method=renyi) developed in Pavoine and Ricotta (2019). Note that Pavoine and Ricotta (2019) recommend the use of index qfeveHill (with method=hill).

Value

If only one value of q is given, the function evoeveparam returns a vector with the evenness values for the communities. If more than one value of q is given, a list of two objects is returned:

q	the vector of values for q;
eve	a data frame with the phylogenetic evenness in each community calculated for all values of q.

The function plot.evoeveparam returns a graphic.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

The methodologies and scripts were developed by

Pavoine, S., Ricotta, C. (2019) A simple translation from indices of species diversity to indices of phylogenetic diversity. *Ecological Indicators*, **101**, 552–561.

using earlier work by:

Chao, A., Chiu, C.-H., Jost, L. (2010) Phylogenetic diversity measures based on Hill numbers. *Philosophical Transactions of the Royal Society London Series B*, **365**, 3599–3609.

Daroczy, Z. (1970) Generalized information functions. *Information and Control*, **16**, 36–51.

Havrda, M., Charvat F. (1967) Quantification method of classification processes: concept of structural alpha- entropy. *Kybernetik*, **3**, 30–35.

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

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

Renyi, A. (1960) On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, **1**, 547–561.

Tsallis, C. (1988) Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, **52**, 480–487.

See Also

evodivparam, evouniparam

Examples

```
## Not run:
if(require(ape)){

data(batcomm)
phy <- read.tree(text=batcomm$tre)
ab <- batcomm$ab[, phy$tip.label]
plot(evoeveparam(phy, ab))
plot(evoeveparam(phy, ab, q=seq(0, 10, length=20)))

}

## End(Not run)
```

evoNSCA	*Evolutionary Non-Symmetric Correspondence Analysis*

Description

The function performs evoNSCA (Pavoine 2016): an adaptation of the non-symmetric correspondence analysis (NSCA) (see e.g. Kroonenberg and Lombardo 1999) to analyse the distributions of lineages among sites and, simultaneously, to analyse the phylogenetic composition of sites.

Usage

```
evoNSCA(phyl, comm, scannf = TRUE, nf = 2, abundance = TRUE)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust. To ease the interpretation of factorial maps, it is advised to use a rooted phylogenetic tree and to add labels to the nodes of the phylogeny. For example, If your phylogeny is named 'tree' and is of class phylo, you can root the phylogeny using tree$root.edge <- 0 and you can name the nodes with the following command: tree$node.label <- paste("n", 1:tree$Nnode, sep=""). Use plot(tree, show.node.label=TRUE) to see the result. If it is rooted and of class phylo4, you can use: nodeLabels(tree) <- paste("n", 1:nNodes(tree), sep="").
comm	a data frame or a matrix typically with communities as rows, species as columns and presence/absence (1/0) or an index of abundance as entries. Species should be labeled as in the phylogenetic tree where they are the tips.
scannf	a logical value indicating whether the screeplot (eigenvalues) should be displayed for choosing the number of axes to be kept.
nf	if scannf is FALSE, an integer indicating the number of axes to be kept.
abundance	a logical value, if FALSE, only the presences/absences of the **evolutionary units** in sites are considered.

Value

evoNSCA returns an object of class evoNSCA and of class dudi (see package ade4). Graphical tools are associated with class dudi in packages ade4 and adegraphics (see section "Examples" below).

The returned object contains the following components:

tab	a data frame with n rows and p columns, with communities as rows and nodes of the phylogeny as columns. Internal data frame used by the algorithm;
cw	weights attributed to the nodes of the phylogeny, a vector with p components;
lw	weights attributed to the communities, a vector with n components;
eig	vector of eigenvalues;
rank	integer, number of axes;

nf	integer, number of kept axes;
c1	normed scores for the nodes of the phylogeny, data frame with p rows and nf columns;
l1	normed scores for the communities, data frame with n rows and nf columns;
co	scores for the nodes of the phylogeny, data frame with p rows and nf columns;
li	scores for the communities, data frame with n rows and nf columns;
call	the original call.

If X is an object of class evoNSCA, then attributes(X)$phy contains the phylogenetic tree (of class phylo) with names for internal nodes.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

Kroonenberg PM, Lombardo R (1999) Nonsymmetric correspondence analysis: a tool for analysing contingency tables with a dependence structure. *Multivariate Behavioral Research*, **34**, 367–396.

See Also

evoCA, evopcachord, evopcahellinger, evodiss

Examples

```
## Not run:
if(require(ade4) && require(ape) && require(adegraphics)){
O <- adegpar()$plabels$optim
adegpar("plabels.optim" = TRUE)

data(batcomm)
ab <- batcomm$ab
phy <- read.tree(text=batcomm$tre)
plot(phy, show.node=TRUE)
evoNSCAbat <- evoNSCA(phy, ab, scan=FALSE, nf=2)
evoNSCAbat$eig/sum(evoNSCAbat$eig)
s.label(evoNSCAbat$li)
s.label(evoNSCAbat$co)
s.arrow(evoNSCAbat$co)
inertia.dudi(evoNSCAbat, row=TRUE)$row.abs
inertia.dudi(evoNSCAbat, col=TRUE)$col.abs

evoNSCAbat <- evoNSCA(phy, ab, scan=FALSE, nf=3) ## All axes are now retained
# The Euclidean (canonical) distances among habitat points on the evoNSCA space is
dist(evoNSCAbat$li)
# which is equal to evoDprofile (see function evodiss):
```

```
evodiss(phy, ab, "LG")

adegpar("plabels.optim" = 0)
}

## End(Not run)
```

---

evopca   *Evolutionary Principal Component analyses*

Description

The functions `evopcachord` and `evopcahellinger` perform two adaptations of Principal Component Analysis (PCA) for the analysis of phylogenetic diversity patterns across species communities: the evolutionary PCA based on Chord distance (evoPCAChord) and the evolutionary PCA based on Hellinger distance (evoPCAHellinger) (Pavoine 2016).

Usage

```
evopcachord(phyl, comm, option = c("centred", "decentred"),
w = c("evoab", "even", "speciesab"), scannf = TRUE,
nf = 2, abundance = TRUE)

evopcahellinger(phyl, comm, option = c("centred", "decentred"),
w = c("evoab", "even", "speciesab"), scannf = TRUE,
nf = 2, abundance = TRUE)
```

Arguments

phyl	an object inheriting the class `phylo` (see package ape), `phylo4` (see package phylobase) or `hclust`. To ease the interpretation of factorial maps, it is advised to use a rooted phylogenetic tree and to add labels to the nodes of the phylogeny. For example, If your phylogeny is named 'tree' and is of class `phylo`, you can root the phylogeny using `tree$root.edge <- 0` and you can name the nodes with the following command: `tree$node.label <- paste("n", 1:tree$Nnode, sep="")`. Use `plot(tree, show.node.label=TRUE)` to see the result. If it is rooted and of class `phylo4`, you can use: `nodeLabels(tree) <- paste("n", 1:nNodes(tree), sep="")`.
comm	a data frame or a matrix typically with communities as rows, species as columns and presence/absence (1/0) or an index of abundance as entries. Species should be labeled as in the phylogenetic tree where they are the tips.
option	a string: either `"centered"` or `"decentered"`. If several strings are given, only the first one is used. See details.
w	a string: either `"evoab"`, `"even"`, or `"speciesab"`. If several strings are given, only the first one is used. See details.
scannf	a logical value indicating whether the screeplot (eigenvalues) should be displayed for choosing the number of axes to be kept.

nf	if scannf is FALSE, an integer indicating the number of axes to be kept.
abundance	a logical value. If FALSE, only the presences/absences of the **evolutionary units** in communities are considered.

Details

Let L_k be the length of branch k in the phylogenetic tree (out of K branches); a_{jk} the sum of abundances, in community j, for all species descending from branch k; w_j a positive weight attributed to community j (the definition for w_j is flexible with the only requirement that $\sum_{j=1}^m w_j = 1$); $a_{j+} = \sum_{k=1}^K L_k a_{jk}$; $a_{+k} = \sum_{j=1}^m a_{jk}$; $a_{++} = \sum_{j=1}^m \sum_{k=1}^K L_k a_{jk}$.

The weights of the communities (argument w) can be "even" (even weights, i.e. relative abundances are considered for evolutionary units), "evoab" (proportional to the summed abundances of all evolutionary units, i.e. absolute abundances are considered for evolutionary units), or "speciesab" (proportional to the summed abundances of all species). Note that if the phylogenetic tree is ultrametric (the distance from any species to the root is constant), then options "evoab" and "speciesab" are equivalent.

In evopcahellinger, as recommended by Rao (1995), the vector used to centre matrix $(\sqrt{a_{jk}/a_{j+}})_{j,k}$ in PCA can be defined as

$$\left(\sum_{j=1}^m w_j \sqrt{a_{jk}/a_{j+}} \right)_k$$

(ordinary weighted mean, option "centered") or as

$$\left(\sqrt{(\sum_{j=1}^m w_j a_{jk})/(\sum_{k=1}^K L_k \sum_{j=1}^m w_j a_{jk})} \right)_k$$

(option "decentered"); see Pavoine (2016) for an introduction of all ordination approaches.

Similarly, in evopcachord, the vector used to centre matrix $(a_{jk}/\sqrt{\sum_{k=1}^K a_{jk}^2})_{j,k}$ can be defined as

$$\left(\sum_{j=1}^m w_j a_{jk} / \sqrt{\sum_{k=1}^K a_{jk}^2} \right)_k$$

(ordinary weighted mean, option "centered") or as

$$\left((\sum_{j=1}^m w_j a_{jk}) / \sqrt{\sum_{k=1}^K L_k (\sum_{j=1}^m w_j a_{jk})^2} \right)_k$$

(option "decentered").

Value

evopcachord and evopcahellinger both return an object of class evopca and of class dudi (see package ade4). Graphical tools are associated with class dudi in packages ade4 and adegraphics (see section "Examples" below).

The returned object contains the following components:

tab	a data frame with n rows and p columns, with communities as rows and nodes of the phylogeny as columns. Internal data frame used by the algorithm;
cw	weights attributed to the nodes of the phylogeny, a vector with p components;
lw	weights attributed to the communities, a vector with n components;
eig	vector of eigenvalues;
rank	integer, number of axes;
nf	integer, number of kept axes;
c1	normed scores for the nodes of the phylogeny, data frame with p rows and nf columns;
l1	normed scores for the communities, data frame with n rows and nf columns;
co	scores for the nodes of the phylogeny, data frame with p rows and nf columns;
li	scores for the communities, data frame with n rows and nf columns;
call	the original call.

If X is an object of class evopca, then attributes(X)$phy contains the phylogenetic tree (of class phylo) with names for internal nodes.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

Rao, C.R. (1995) A review of canonical coordinates and an alternative to correspondence analysis using Hellinger distance. *Questiio*, **19**, 23–63.

See Also

evoCA, evoNSCA, evodiss

Examples

```
## Not run:
if(require(ade4) && require(ape) && require(adegraphics)){
O <- adegpar()$plabels$optim
adegpar("plabels.optim" = TRUE)

data(batcomm)
ab <- batcomm$ab
phy <- read.tree(text=batcomm$tre)
plot(phy, show.node=TRUE)

evopcaHbat <- evopcahellinger(phy, ab, scan=FALSE, nf=3)
dist(evopcaHbat$li)
evodiss(phy, ab, "Hellinger")
```

```
evopcaHbat$eig/sum(evopcaHbat$eig)
s.label(evopcaHbat$li)
s.label(evopcaHbat$co)
s.arrow(evopcaHbat$co)
inertia.dudi(evopcaHbat, row=TRUE)$row.abs
inertia.dudi(evopcaHbat, col=TRUE)$col.abs

evopcaCbat <- evopcachord(phy, ab, scan=FALSE, nf=3)
dist(evopcaCbat$li)
evodiss(phy, ab, "Chord")

evopcaCbat$eig/sum(evopcaCbat$eig)
s.label(evopcaCbat$li)
s.label(evopcaCbat$co)
s.arrow(evopcaCbat$co)
inertia.dudi(evopcaCbat, row=TRUE)$row.abs
inertia.dudi(evopcaCbat, col=TRUE)$col.abs

adegpar("plabels.optim" = O)
}

## End(Not run)
```

---

evouniparam   *Parametric Indices of Phylogenetic Uniqueness*

Description

Function `evouniparam` calculates phylogenetic uniqueness in communities using parametric indices derived from Tsallis and Hill compositional indices. `evouniparam` can also be applied to functional trees rather than phylogenies, to calculate a functional uniqueness. The function `plot.evouniparam` plots the results of function `evouniparam`.

Usage

```
evouniparam(phyl, comm,
    method = c("hill", "tsallis", "renyi"),
    q = 2, tol = 1e-08)

## S3 method for class 'evouniparam'
plot(x, legend = TRUE,
    legendposi = "topright",
    axisLABEL = "Tree-based uniqueness",
    type="b", col = if(is.numeric(x)) NULL
    else sample(colors(distinct = TRUE), nrow(x$uni)),
    lty = if(is.numeric(x)) NULL else rep(1, nrow(x$uni)),
    pch = if(is.numeric(x)) NULL else rep(19, nrow(x$uni)),
    ...)
```

Arguments

phyl	an object inheriting the class `phylo` (see package ape), `phylo4` (see package phylobase), or `hclust`.
comm	a data frame or a matrix typically with communities as rows, species as columns and abundance as entry. Species should be labeled as in the phylogenetic tree where they are the tips.
method	a string: either "hill" for the Hill numbers (Hill 1973), "tsallis" for the Tsallis or HCDT entropy (Harvda and Charvat 1967; Daroczy 1970; Tsallis 1988), or "renyi" for Renyi's entropy (Renyi 1960). If several values are given, only the first one is considered. See details.
q	a vector with nonnegative value(s) for parameter q. See details.
tol	numeric tolerance threshold: values between -tol and tol are considered equal to zero.
x	an object of class `evouniparam` obtained with function `evouniparam`.
legend	a logical. If TRUE a legend is given with the colour, the type of line (etc.) used to define the uniqueness curve of each community.
legendposi	a string that gives the position of the legend to be passed to function `legend` of the base of R.
axisLABEL	a string to display on the main axis of the plot to designate what we are measuring. The default is "Tree-based uniqueness".
type	a string to be passed to the graphic argument type of functions `plot` and `lines` used to draw the uniqueness curve of each community.
col	vector of colours to be passed to the graphic argument col of functions `plot` and `lines` to define the colour of the uniqueness curve of each community.
lty	vector of type of line (plain, broken etc.) to be passed to the graphic argument `lty` of functions `plot` and `lines` used to draw the uniqueness curve of each community.
pch	type of point (open circle, close circle, square etc.) to be passed to the graphic argument pch of functions `plot` and `lines` used to draw the uniqueness level of each community.
...	other arguments can be added and passed to the functions `plot` and `lines` used to draw the graphic.

Details

Function `evouniparam` calculates feature uniqueness (features = branch units on a phylogenetic [or functional] tree) using parametric indices qfuniHCDT (with `method=tsallis`), qfuniHill (with `method=hill`), qfuniRenyi (with `method=renyi`) developed in Pavoine and Ricotta (2019). Note that Pavoine and Ricotta (2019) recommend the use of index qfuniHill (with `method=hill`).

Value

If only one value of q is given, the function `evouniparam` returns a vector with the phylogenetic uniqueness of each community. If more than one value of q is given, a list of two objects is returned:

q	the vector of values for q;
uni	a data frame with the phylogenetic uniqueness in each community calculated for all values of q.

The function `plot.evouniparam` returns a graphic.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

The methodologies and scripts were developed by

Pavoine, S., Ricotta, C. (2019) A simple translation from indices of species diversity to indices of phylogenetic diversity. *Ecological Indicators*, **101**, 552–561.

using earlier work by:

Chao, A., Chiu, C.-H., Jost, L. (2010) Phylogenetic diversity measures based on Hill numbers. *Philosophical Transactions of the Royal Society London Series B*, **365**, 3599–3609.

Daroczy, Z. (1970) Generalized information functions. *Information and Control*, **16**, 36–51.

Havrda, M., Charvat F. (1967) Quantification method of classification processes: concept of structural alpha- entropy. *Kybernetik*, **3**, 30–35.

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

Pavoine, S. (2016) A guide through a family of phylogenetic dissimilarity measures among sites. *Oikos*, **125**, 1719–1732.

Renyi, A. (1960) On measures of entropy and information. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, **1**, 547–561.

Tsallis, C. (1988) Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, **52**, 480–487.

See Also

evodivparam, evoeveparam

Examples

```
## Not run:
if(require(ape)){

data(batcomm)
phy <- read.tree(text=batcomm$tre)
ab <- batcomm$ab[, phy$tip.label]
plot(evouniparam(phy, ab))
plot(evouniparam(phy, ab, q=seq(0, 10, length=20)))

}

## End(Not run)
```

FPdivparam   *Parametric Indices of Functional and Phylogenetic Diversity*

Description

Function `FPivparam` calculates functional or phylogenetic (FP-)diversity in communities using parametric indices $^{\alpha}K$, $^{\alpha}K*$, $^{\alpha}Y$ and $^{\alpha}I$ discussed and developed in Pavoine and Ricotta (2021). The function `plot.FPdivparam` plots the results of function `FPdivparam`.

Usage

```
FPdivparam(comm, disORtree, method = c("KY", "KstarI"), palpha = 2,
    equivalent = FALSE, option = c("asymmetric", "symmetric"),
    dmax = NULL, tol = 1e-8)

## S3 method for class 'FPdivparam'
plot(x, legend = TRUE,
    legendposi = "topright", axisLABEL = "FP-diversity",
    type="b", col = if(is.numeric(x)) NULL
    else sample(colors(distinct = TRUE), nrow(x$div)),
    lty = if(is.numeric(x)) NULL else rep(1, nrow(x$div)),
    pch = if(is.numeric(x)) NULL else rep(19, nrow(x$div)),
    ...)
```

Arguments

comm	a data frame or a matrix typically with communities as rows, species as columns and abundance as entry. Species should be labelled as in object disORtree.
disORtree	an object inheriting the class `dist`, giving the FP-dissimilarities between species, or inheriting the class `phylo` (see package ape), `phylo4` (see package phylobase), or `hclust`, where species are tips.
method	a string either `"KY"` for index $^{\alpha}K$ (if `disORtree` is of class `dist`) or $^{\alpha}Y$ if `disORtree` is of class `phylo`, `phylo4` or `hclust`; or `"KstarI"` for index $^{\alpha}K*$ (if `disORtree` is of class `dist`) or $^{\alpha}I$ if `disORtree` is of class `phylo`, `phylo4` or `hclust`. If several values are given, only the first one is considered.
palpha	a vector with nonnegative value(s) for parameter α.
equivalent	a logical. If TRUE, the diversity values are calculated in terms of equivalent number of species (see Pavoine and Ricotta, 2021).
option	a string either `"asymmetric"` or `"symmetric"`. The parameter option is only used if `disORtree` is of class `phylo`, `phylo4` or `hclust` If `option="symmetric"`, the distance between two tips on a tree is defined as half the sum of branch length on the smallest path that connects the two species; while if `option="asymmetric"` the distance between a tip *i* and a tip *j* on a tree is defined as the sum of branch lengths between tip *i* and its most recent ancestor with tip *j*. If the tree is ultrametric, the two options are equivalent.

dmax	a nonnegative numeric indicating the maximum possible dissimilarity between two species. dmax must be higher than or equal to the maximum observed dissimilarity between two species.
tol	a numeric tolerance threshold: values between -tol and tol are considered equal to zero.
x	an object of class FPdivparam obtained with function FPdivparam.
legend	a logical. If TRUE a legend is given with the colour, the type of line (etc.) used to define the diversity curve of each community.
legendposi	a string that gives the position of the legend to be passed to function legend of the base of R.
axisLABEL	a string to display on the main axis of the plot to designate what we are measuring. The default is "FP-diversity".
type	a string to be passed to the graphic argument type of functions plot and lines used to draw the diversity curve of each community.
col	vector of colours to be passed to the graphic argument col of functions plot and lines to define the colour of the diversity curve of each community.
lty	vector of type of line (plain, broken etc.) to be passed to the graphic argument lty of functions plot and lines used to draw the diversity curve of each community.
pch	type of point (open circle, close circle, square etc.) to be passed to the graphic argument pch of functions plot and lines used to draw the diversity level of each community.
...	other arguments can be added and passed to the functions plot and lines used to draw the graphic.

Value

If only one value of palpha is given, function FPdivparam returns a vector with the phylogenetic diversity of each community. If more than one value of palpha is given, a list of two objects is returned:

palpha	the vector of values for palpha;
div	a data frame with the phylogenetic diversity of each community calculated for all values of palpha.

The function plot.FPdivparam returns a graphic.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

The methodologies and scripts were developed by

Pavoine, S., Ricotta, C. (2021) On the relationships between rarity, uniqueness, distinctiveness, originality and functional/phylogenetic diversity. *BiorXiv*. doi:10.1101/2021.08.09.455640

unifying and extending earlier work by:

Pavoine, S., Love, M., Bonsall, M. (2009) Hierarchical partitioning of evolutionary and ecological patterns in the organization of phylogenetically-structured species assemblages: application to rockfish (genus: Sebastes) in the Southern California Bight. *Ecology Letters*, **12**, 898–908.

and

Ricotta, C., Szeidl, L. (2006) Towards a unifying approach to diversity measures: Bridging the gap between the Shannon entropy and Rao's quadratic index. *Theoretical Population Biology*, **70**, 237–243.

See Also

`pIa`, `evodivparam`

Examples

```
## Not run:
if(require(ape)){

data(batcomm)
phy2 <- read.tree(text=batcomm$tre2)
ab2 <- batcomm$ab2[, phy2$tip.label]
plot(FPdivparam(ab2, phy2))
plot(FPdivparam(ab2, phy2, palpha=seq(0, 10, length=20)))


}

## End(Not run)
```

FunImbalance   *Indices of functional imbalance*

Description

The function `FunImbalance` calculates functional imbalance within species communities, as an indicator of the strength of interaction between species abundances and their functional dissimilarities.

Usage

```
FunImbalance(comm, dis, method = c("CorB", "SESB", "QB"), nrep = 10000, tol = 1e-16)
```

Arguments

comm	matrix or data frame with communities as rows, species as columns and abundance data as entries.
dis	an object of class dist that provides the functional dissimilarities between species.

method	a character string: one or a vector of "CorB", "SESB", and "QB". The strings indicate the name of the index of functional imbalance that must be used.
nrep	an integer. The number of repetitions (permutations) to use in the calculation of SESB and QB.
tol	a tolerance number: a value in [-tol, tol] is considered as zero.

Value

The function `betaUniqueness` returns a data frame with communities as rows and the selected indices of functional imbalance as columns.

Author(s)

Sandrine Pavoine `<sandrine.pavoine@mnhn.fr>`

References

Ricotta, C., Bacaro, G., Maccherini, S., Pavoine, S. (2022) Functional imbalance not functional evenness is the third component of community structure, *Contact author for information*

Examples

```
data(RutorGlacier)
fundis <- dist(scale(RutorGlacier$Traits2[1:6]))
fundis <- fundis/max(fundis)
funImb <- FunImbalance(RutorGlacier$Abund, fundis, method = "CorB")
funImb
```

---

generalized_Tradidiss	*Plot-to-plot dissimilarity taking account of functional dissimilarities between species*

Description

Given a matrix of *S* species' relative or absolute abundance values in *N* plots, together with an *S* x *S* (functional) dissimilarity matrix, the function `generalized_Tradidiss` calculates a semimatrix with the values of a plot-to-plot dissimilarity index, as proposed in Pavoine and Ricotta (2019).

Usage

```
generalized_Tradidiss(comm, dis, method = c("GC", "MS", "PE"),
    abundance = c("relative", "absolute", "none"),
    weights = c("uneven", "even"), tol = 1e-8)
```

Arguments

`comm`	a data frame typically with communities as rows, species as columns and an index of abundance as entries. Species must be labeled as in the object `dis`.
`dis`	an object of class `dist` or a matrix containing pair-wise (functional or phylogenetic) dissimilarities between species [obtained by functions like vegdist in package vegan (Oksanen et al. 2013), gowdis in package FD (Laliberte and Shipley 2011), or dist.ktab in package ade4 for functional dissimilarities (Dray et al. 2007), or functions like cophenetic.phylo in package ape (Paradis et al. 2004) or distTips in package adephylo (Jombart and Dray 2010) for phylogenetic dissimilarities]. If the dissimilarities are outside the range 0-1, a warning message is displayed and each dissimilarity is divided by the maximum over all pairwise dissimilarities.
`method`	one of the following strings: `"GC"`, `"MS"`, `"PE"`. See Details.
`abundance`	a string with three possible values: "relative" for the use of relative species abundance, "absolute" for the use of absolute species abundance, and "none" for the use of presence/absence data (1/0).
`weights`	a string. Two types of weights are available in the function: `"uneven"` (Eq. 5 in Pavoine and Ricotta (2009)) or `"even"` (1/S, where S is the number of species in the two compared plots).
`tol`	numeric tolerance threshold: values between `-tol` and `tol` are considered equal to zero.

Details

The plot-to-plot dissimilarity coefficients used in this function are as follows: `"GC"`: Equation 6 in Pavoine and Ricotta (2019) `"MS"`: Equation 8 in Pavoine and Ricotta (2019) `"PE"`: Equations 9 and 10 in Pavoine and Ricotta (2019)

Value

The function returns an object of class `"dist"` with the values of the proposed dissimilarities for each pair of plots.

Author(s)

Sandrine Pavoine `<sandrine.pavoine@mnhn.fr>`

References

Pavoine, S. and Ricotta, C. (2019) Measuring functional dissimilarity among plots: adapting old methods to new questions. *Ecological Indicators*, **97**, 67–72.

Examples

```
## Not run:
if(require(ade4) && require(adephylo) && require(ape)){

data(birdData)
```

```
phy <- read.tree(text=birdData$tre)
phydis <- sqrt(distTips(phy, method="nNodes")+1)
fau <- birdData$fau[1:6, phy$tip.label]
disGC <- generalized_Tradidiss(fau, phydis, method="GC")
disGC

### The second example is a bit TIME CONSUMING
data(mafragh)
namspe <- rownames(mafragh$traits[[1]])
M <- mafragh$flo
colnames(M) <- namspe
Bin <- prep.binary(mafragh$traits$tabBinary, c(3, 4))
distraits <- dist.ktab(ktab.list.df(list(mafragh$traits$tabOrdinal[,2:3], Bin)),
    c("O","B"), scan=FALSE)
disGC <- generalized_Tradidiss(M, distraits, method="GC")
pcoGC <- dudi.pco(as.dist(cailliez(disGC)), full=TRUE)
s.value(mafragh$xy, pcoGC$li[,1])
}

## End(Not run)
```

K   *Indices K, Kstar and Kw of Phylogenetic Signal*

Description

The functions K, Kstar and Kw calculate Blomberg et al. (2003) statistics *K*, and *K** and Pavoine and Ricotta (2013) statistic K_w, respectively. Then they perform a permutation test where species identities are maintained in the phylogeny while the trait values of species are randomly shuffled (permuted) (Pavoine and Ricotta 2013).

Usage

```
K(phyl, trait, nrep = 999, alter = c("greater", "less", "two-sided"))

Kstar(phyl, trait, nrep = 999, alter = c("greater", "less", "two-sided"))

Kw(phyl, trait, nrep = 999, alter = c("greater", "less", "two-sided"))
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust.
trait	a vector with the trait value for each species (tip) in the phylogenetic tree. Trait values for species must be in the same order as species in the phylogenetic tree.
nrep	a numeric: the number of randomizations.
alter	a string specifying the alternative hypothesis; it must be one of "greater" (phylogenetic signal; the default), "less" or "two-sided". If several values are given, only the first one is retained.

Details

Blomberg et al. (2003) introduced two statistics of phylogenetic signal:

$$K = MSE0/MSE$$

$$K^* = MSE^*/MSE$$

where *MSE* is the mean squared error of the trait values calculated using the variance-covariance matrix derived from the phylogenetic tree, *MSE0* is the mean squared error of the tip trait values, measured from a phylogenetically correct mean of tip trait values and *MSE** is the mean squared error of the tip trait values, measured from the estimate of the mean of the raw tip trait values. In both statistics *K* and *K**, the value of *MSE* will be relatively small if the phylogenetic tree accurately describes the variance-covariance pattern observed in the data, leading to high values for *K* and *K** (meaning high phylogenetic signal). In functions K and Kstar, *K* and *K**, respectively, are divided (normalized) by their expected value if the trait evolved under a Brownian motion along the branches of the phylogenetic tree (this expected value is invariant under permutation of trait values among the tips of the phylogeny).

To test for phylogenetic signal, Blomberg et al. (2003) actually considered neither *K* nor *K** but *MSE* as the core statistic associated with random permutations of trait values among tips of the phylogenetic tree. Although the literature on phylogenetic signal has currently mostly ignored *K** focusing on statistic *K*, *K** could thus actually have been considered as the core statistic of Blomberg et al. (2003) test for phylogenetic signal. Indeed, as *MSE** is independent of permutations of trait values among the tips of the phylogeny while *MSE0* is, Blomberg et al. (2003) approach corresponds to considering *K** and not *K* as the statistic of the test of phylogenetic signal in traits. This test is also equivalent to an alternative implemented via phylogenetically independent contrasts also proposed by Blomberg et al. (2003).

Function KW implements index K_w, a modified version of *K** that grants a higher importance in the calculation of phylogenetic signal to the tips that have many closely related tips (Pavoine and Ricotta 2013).

In functions, K, Kstar and Kw, I considered the same permutation scheme as in Blomberg et al. (2003) but used *K*, *K** and K_w, as the core statistic, respectively. The test developed by Blomberg et al. (2003) thus corresponds to function Kstar.

Value

Each function returns an object of class randtest with the results of the permutation tests. (see function randtest in package ade4)

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Blomberg, S.P., Garland, T., Ives, A.R. (2003) Testing for phylogenetic signal in comparative data: behavioral traits are more labile. *Evolution*, **57**, 717–745.

Pavoine, S., Ricotta, C. (2013) Testing for phylogenetic signal in biological traits: the ubiquity of cross-product statistics. *Evolution*, **67**, 828–840.

See Also

[Kstar](#), [Kw](#)

Examples

```
## Not run:
if(require(ape) && require(ade4)){

data(rockfish)
phy <- read.tree(text=rockfish$tre)

theK <- K(phy, rockfish$traits[phy$tip.label, 1])
theK
plot(theK)

theKstar <- Kstar(phy, rockfish$traits[phy$tip.label, 1])
theKstar
plot(theKstar)

theKw <- Kw(phy, rockfish$traits[phy$tip.label, 1])
theKw
plot(theKw)

}

## End(Not run)
```

---

optimEH   *Nee and May's Optimizing Process*

Description

The function optimEH performs Nee and May's optimizing scheme. When branch lengths in an ultrametric phylogenetic tree are expressed as divergence times, the total sum of branch lengths in that tree expresses the amount of evolutionary history. Nee and May's algorithm optimizes the amount of evolutionary history preserved if only *k* species out of *n* were to be saved. The *k*-1 closest-to-root nodes are selected, which defines *k* clades; one species from each clade is picked. At this last step, we decide to select the most original species of each of the *k* clades.

Usage

```
optimEH(phyl, nbofsp, tol = 1e-08, give.list = TRUE)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust.
nbofsp	an integer indicating the number of species saved (*k*).

tol	a tolerance threshold for null values (a value less than tol in absolute terms is considered as zero).
give.list	a logical indicating whether a list of optimizing species should be provided. If give.list = TRUE, optimEH provides the list of the k species that optimize the amount of evolutionary history preserved and are the most original species in their clades. If give.list = FALSE, optimEH returns directly the real value giving the amount of evolutionary history preserved.

Value

If give.list=FALSE, function optimEH returns a numeric.

If give.list=TRUE, function optimEH returns a list containing:

value	a real value providing the amount of evolutionary history preserved;
selected.sp	a data frame containing the list of the k species which optimize the amount of evolutionary history preserved and are the most original species in their clades.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr> with contributions of Stephane Dray

References

Pavoine, S., Ollier, S. and Dufour, A.-B. (2005) Is the originality of a species measurable? *Ecology Letters*, **8**, 579–586.

See Also

randEH

Examples

```
## Not run:
if(require(ape) && require(adephylo)){

data(carni70, package="adephylo")
tre <- read.tree(text=carni70$tre)
adiv:::EH(tre)
adiv:::optimEH(tre, 10)

}

## End(Not run)
```

---

orisaved   *Maximal or Minimal Amount of Originality Saved under Optimal Conditions*

Description

Function `orisaved` computes the maximal or minimal amount of originality saved over all combinations of species optimizing the amount of evolutionary history preserved. The originality of a species is measured with the QE-based index (Pavoine et al. 2005).

Usage

```
orisaved(phyl, rate = 0.1, method = 1:2)
```

Arguments

phyl   an object inheriting the class `phylo` (see package ape), `phylo4` (see package phylobase) or `hclust`.

rate   a real value (between 0 and 1) indicating how many species will be saved for each calculation. For example, if the total number of species is 70 and 'rate = 0.1' then the calculations will be done at a rate of 10% i.e. for 0 (= 0 %), 7 (= 10 %), 14 (= 20 %), 21 (= 30 %), ..., 63 (= 90 %) and 70(= 100 %) species saved. If 'rate = 0.5' then the calculations will be done for only 0 (= 0 %), 35 (= 50 %) and 70(= 100 %) species saved.

method   an integer either 1 or 2 (see details).

Details

1 = maximum amount of originality saved 2 = minimum amount of originality saved

Value

Returns a numeric vector.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr> with contributions of Stephane Dray

References

Pavoine, S., Ollier, S. and Dufour, A.-B. (2005) Is the originality of a species measurable? *Ecology Letters*, **8**, 579–586.

See Also

[optimEH](#), [randEH](#)

Examples

```
## Not run:
if(require(ape) && require(adephylo)){

data(carni70, package="adephylo")
tre <- read.tree(text=carni70$tre)
tmax <- adiv:::orisaved(tre, rate = 1 / 70, method = 1)
tmin <- adiv:::orisaved(tre, rate = 1 / 70, method = 2)
plot(c(0, 1:70), tmax, xlab = "nb of species saved", ylab = "Originality saved", type = "l")
lines(c(0, 1:70), tmin, lty = 2)


}

## End(Not run)
```

PADDis   *Functional Dissimilarity Measures for Presence-Absence Data*

Description

Functions `PADDis`, `DJac` and `Jac` calculate the dissimilarity coefficients introduced in Ricotta et al. (2016). These dissimilarity coefficients use traditional mismatching components a, b and c of the 2 x 2 contingency table expressed as to include functional or phylogenetic differences among species and noted A, B, C. Components B and C represent the functional or phylogenetic uniqueness of community X compared with community Y and vice versa. Component A represents the functional or phylogenetic similarities between communities X and Y.

Usage

```
PADDis(comm, dis, method = NULL, diag = FALSE, upper = FALSE)

DJac(comm, dis, diag = FALSE, upper = FALSE)

Jac(comm, diag = FALSE, upper = FALSE)
```

Arguments

comm	a matrix or a data frame with communities (or plots, assemblages, etc.) as rows and species as columns containing the incidence (0/1) of all species in the communities.
dis	an object of class `dist` containing the (functional) dissimilarities among species.
method	an integer between 0 and 6. If `NULL` the choice is made with a console message. See details.
diag	a logical value indicating whether the diagonal of the distance matrix should be printed by function `print.dist`.
upper	a logical value indicating whether the upper triangle of the distance matrix should be printed by function `print.dist`.

Details

In PADDIS, dissimilarities among communities are calculated with the following formulas:

Generalized Jaccard dissimilarity, with method = 1

$$\frac{B + C}{a + b + c}$$

Generalized Sorensen dissimilarity, with method = 2

$$\frac{B + C}{2a + b + c}$$

Generalized Sokal and Sneath dissimilarity, with method = 3

$$\frac{2(B + C)}{a + 2(b + c)}$$

Generalized Ochiai dissimilarity, with method = 4

$$\frac{\sqrt{A + B}\sqrt{A + C} - A}{\sqrt{a + b}\sqrt{a + c}}$$

Generalized Simpson dissimilarity, with method = 5

$$\frac{min\{B + C\}}{a + min\{b + c\}}$$

Generalized Kulczynski dissimilarity, with method = 6

$$0.5 * \left(\frac{B}{a + b} + \frac{C}{a + c}\right)$$

DJac and Jac use the additive decomposition of the Jaccard index into turnover and richness difference. DJac takes into account the (functional or phylogenetic) dissimilarities among species while Jac does not.

Value

In function PADDis, if method=0, then the function PADDis returns 6 matrices corresponding to the a, b, c, A, B, and C values per pair of communities. Otherwise, it returns an object of class dist corresponding to the dissimilarities among communities.

Functions DJac and Jac return a list of three objects of class dist:

J	for the full dissimilarities between communities;
JRepl	for the turnover component of the dissimilarities;
JRich	for the component of difference in richness.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C., Podani, J., Pavoine, S. (2016) A family of functional dissimilarity measures for presence and absence data. *Ecology and Evolution*, **6**, 5383–5389

Examples

```
data(RPP16EE)
Com <- RPP16EE$Com
Dis <- as.dist(RPP16EE$Dis)
J <- Jac(Com)
DJ <- DJac(Com, Dis)

plot(c(as.matrix(DJ$J)[1,]), ylab="Dissimilarity",
xlab="Plot-to-plot comparison", pch=15, type="b",
ylim=c(0,1), main="Jaccard")

lines(c(as.matrix(J$J)[1,]), type="b", pch=18)

legend("bottomright", legend=c("P/A scores", "functional data"),
pch=c(15,18), lty=1)

plot(c(as.matrix(DJ$JRepl)[1,]), ylab="Dissimilarity",
xlab="Plot-to-plot comparison", pch=15, type="b",
ylim=c(0,1), main="Species replacement")

lines(c(as.matrix(J$JRepl)[1,]), type="b", pch=18)

legend("bottomright", legend=c("P/A scores", "functional data"),
pch=c(15,18), lty=1)


#Use the following instruction to obtain all components:

PADDis(Com, Dis, method=0)
```

---

plot.phylo4d   *Plots for phylogenetic tree and data*

Description

Functions `barp4d`, `dotp4d` and `gridp4d` provide plots for phylo4d objects (i.e. phylogenetic tree and data). Function `plot.phylo4d` provides a general interface for all other functions. Function `barp4d` uses barplots of trait values along the phylogenetic tree. Function `dotp4d` uses dotplots of trait values along the phylogenetic tree, and function `gridp4d` gridplots.

Usage

```
## S3 method for class 'phylo4d'
plot(x, trait = names(tdata(p4d)), center = TRUE,
```

```
    scale = TRUE, plot.type = "barplot", tree.ladderize = FALSE,
    tree.type = "phylogram", tree.ratio = NULL,
    tree.xlim = NULL, tree.open.angle = 0,
    tree.open.crown = TRUE, show.tip = TRUE, tip.labels = NULL,
    tip.col = "black", tip.cex = 1, tip.font = 3, tip.adj = 0,
    data.xlim = NULL, bar.lwd = 10, bar.col = "grey35",
    show.data.axis = TRUE, dot.col = "black", dot.pch = 20,
    dot.cex = 2, cell.col = topo.colors(100),
    show.color.scale = TRUE, show.trait = TRUE,
    trait.labels = NULL, trait.col = "black",
    trait.cex = 1, trait.font = 1, trait.bg.col = "grey90",
    error.bar.sup = NULL, error.bar.inf = NULL,
    error.bar.col = 1, show.box = FALSE, grid.vertical = TRUE,
    grid.horizontal = FALSE, grid.col = "grey25",
    grid.lty = "dashed", ...)

barp4d(height, trait = names(tdata(height)),
    center = TRUE, scale = TRUE, tree.ladderize = FALSE,
    tree.type = "phylogram", tree.ratio = NULL,
    tree.xlim = NULL, tree.open.angle = 0,
    tree.open.crown = TRUE, show.tip = TRUE, tip.labels = NULL,
    tip.col = "black", tip.cex = 1, tip.font = 3, tip.adj = 0,
    data.xlim = NULL, bar.lwd = 10, bar.col = "grey35",
    show.data.axis = TRUE, show.trait = TRUE,
    trait.labels = NULL, trait.col = "black", trait.cex = 1,
    trait.font = 1, trait.bg.col = "grey90",
    error.bar.sup = NULL, error.bar.inf = NULL,
    error.bar.col = 1, show.box = FALSE, grid.vertical = TRUE,
    grid.horizontal = FALSE, grid.col = "grey25",
    grid.lty = "dashed", ...)

dotp4d(p4d, trait = names(tdata(p4d)), center = TRUE,
    scale = TRUE, tree.ladderize = FALSE,
    tree.type = "phylogram", tree.ratio = NULL,
    tree.xlim = NULL, tree.open.angle = 0,
    tree.open.crown = TRUE, show.tip = TRUE, tip.labels = NULL,
    tip.col = "black", tip.cex = 1, tip.font = 3,
    tip.adj = 0, data.xlim = NULL, show.data.axis = TRUE,
    dot.col = "black", dot.pch = 20, dot.cex = 2,
    show.trait = TRUE, trait.labels = NULL, trait.col = "black",
    trait.cex = 1, trait.font = 1, trait.bg.col = "grey90",
    error.bar.sup = NULL, error.bar.inf = NULL,
    error.bar.col = 1, show.box = FALSE, grid.vertical = FALSE,
    grid.horizontal = TRUE, grid.col = "grey25",
    grid.lty = "dashed", ...)

gridp4d(p4d, trait = names(tdata(p4d)), center = TRUE,
    scale = TRUE, tree.ladderize = FALSE,
```

```
tree.type = "phylogram", tree.ratio = NULL,
tree.xlim = NULL, tree.open.angle = 0,
tree.open.crown = TRUE, show.tip = TRUE, tip.labels = NULL,
tip.col = "black", tip.cex = 1, tip.font = 3, tip.adj = 0,
cell.col = topo.colors(100), show.color.scale = TRUE,
show.trait = TRUE, trait.labels = NULL, trait.col = "black",
trait.cex = 0.7, trait.font = 1, trait.bg.col = "grey90",
show.box = FALSE, grid.vertical = FALSE,
grid.horizontal = FALSE, grid.col = "grey25",
grid.lty = "dashed", ...)
```

Arguments

x, p4d, height	a phylo4d object.
trait	the traits in the phylo4d object to include in the plot. Can be a character vector giving the name of the traits or numbers giving the column index in the table of the data slot of the p4d or height object. Can be used to reorder the traits in the plot.
center	a logical indicating whether traits values should be centered.
scale	a logical indicating whether traits values should be scaled.
plot.type	a character string specifying the type of plot for traits data. Can be "barplot", "dotplot" or "gridplot".
tree.ladderize	a logical indicating whether the tree should be (right) ladderized.
tree.type	a character string specifying the type of phylogeny to be drawn. Can be "phylogram", "cladogram" or "fan".
tree.ratio	a numeric value in [0, 1] giving the proportion of width of the figure for the tree.
tree.xlim	a numeric vector of length 2 giving the limits of the x-axis for the tree. If NULL, it is determined automatically.
tree.open.angle	
	a numeric value giving the angle in degrees left blank if tree.type = "fan".
tree.open.crown	
	a logical indicating whether the crowns should be drawn following the value of tree.open.angle (default TRUE).
show.tip	logical indicating whether tips labels should be drawn.
tip.labels	character vector to label the tips. If NULL the tips labels of the phylo4d object are used.
tip.col	a vector of R colors to use for the tips labels. Recycled if necessary.
tip.cex	a numeric vector to control character size of the tips labels. Recycled if necessary.
tip.font	an integer vector specifying the type of font for the tips labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.
tip.adj	a vector of numeric in [0, 1] to control tips labels justification: 0 (left-justification), 0.5 (centering), or 1 (right-justification). Recycled if necessary.

`data.xlim`	numeric vector of length 2 or matrix giving the x coordinates range for the barplots/dotplots.
`bar.lwd`	a vector of numeric giving bar widths of the barplot(s). Recycled along the tips, reapeated for each trait.
`bar.col`	a vector of R colors to use for the bars. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning.
`show.data.axis`	logical indicating whether barplots/dotplots axes should be drawn.
`dot.col`	a vector of R colors to use for the points. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning.
`dot.pch`	a numerical vector of symbol to use for the points. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning.
`dot.cex`	a numerical vector. Character (or symbol) expansion for the points. Recycled along the tips, reapeated for each trait. The user can also provide a matrix for a finer tuning.
`cell.col`	a vector of colors for `gridplot` cells. Easily generated by [heat.colors](), [topo.colors](), [terrain.colors]() or other functions created with `link{colorRampPalette}`.
`show.color.scale`	
	logical indicating whether color scale should be drawn.
`show.trait`	logical indicating whether traits labels should be drawn.
`trait.labels`	character vector to label the traits. If `NULL` the traits labels of the phylo4d object are used.
`trait.col`	a vector of R colors to use for the traits labels. Recycled if necessary.
`trait.cex`	a numeric vector to control character size of the trait labels. Recycled if necessary.
`trait.font`	an integer vector specifying the type of font for the traits labels: 1 (plain text), 2 (bold), 3 (italic), or 4 (bold italic). Recycled if necessary.
`trait.bg.col`	a vector of R colors to use for the background of the barplots. Recycled if necessary.
`error.bar.sup`	a matrix giving the superior limit for error bars. Columns and rows names must match with traits and tips labels, respectively.
`error.bar.inf`	a matrix giving the inferior limit for error bars. Columns and rows names must match with traits and tips labels, respectively.
`error.bar.col`	a vector of R colors to draw error bars.
`show.box`	a logical indicating whether a box should be drawn around the plots.
`grid.vertical`	a logical incating whether vertical lines of the grid should be drawn.
`grid.horizontal`	
	a logical incating whether horizontal lines of the grid should be drawn.
`grid.col`	a vector of R colors to use for the lines of the grid.
`grid.lty`	the lines type of the grid. Possibly a vector.
`...`	further arguments to be passed to `plot.phylo`.

Author(s)

The four functions were written by Francois Keck in the package named phylosignal. Functions were there named as follows: multiplot.phylo4d, barplot.phylo4d, dotplot.phylo4d, and gridplot.phylo4d. At the end of 2019, the package was orphaned and the functions were integrated in package adiv. The versions of the functions have been slightly modified compared to those developed by Francois Keck.

Examples

```
if(require(ape) && require(phylobase)){
data(batcomm)
# Phylogenetic tree for bat species
phy <- read.tree(text = batcomm$tre)
# Abondance data plotted in front of the phylogeny
# F = rainforest, P = cacao plantation
# O = oldfields, C = cornfields
ab.4d <- phylo4d(phy, t(batcomm$ab))
barp4d(ab.4d, center = FALSE, scale = FALSE,
    data.xlim = c(0, max(batcomm$ab)))
dotp4d(ab.4d, center = FALSE, scale = FALSE,
    data.xlim = c(0, max(batcomm$ab)))
gridp4d(ab.4d, center = FALSE, scale = FALSE)
}
```

---

qDT *Mean Diversity along a Phylogenetic Tree*

Description

Function qDT calculates the $^qD(T)$ index developed by Chao et al. (2010) as the mean diversity of order q over T years in a phylogenetic tree. In function qDT, the index is computed over the whole tree from root to tips. It uses the formula of the $^qD(T)$ index extended to non-ultrametric trees (where the distance from tip to root varies) (Chao et al. 2010, equation 4.5) .

Usage

```
qDT(phyl, comm, q = 2, tol = 1e-08)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase), or hclust.
comm	a data frame or a matrix typically with communities as rows, species as columns and abundance as entry. Species should be labeled as in the phylogenetic tree where they are the tips.
q	a vector with nonnegative value(s) for parameter q. See details.
tol	numeric tolerance threshold: values between -tol and tol are considered equal to zero.

Value

If only one value of q is given, a vector with the phylogenetic diversity of each community is returned. If more than one value of q is given, a list of two objects is returned:

q   the vector of values for q;

div   a data frame with the phylogenetic diversity of each community calculated for all values of q.

The results of function plot.qDT are of class "evodivparam". As such, function `plot.evodivparam` allows to plot the results of function qDT.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Chao, A., Chiu, C.-H., Jost, L. (2010) Phylogenetic diversity measures based on Hill numbers. *Philosophical Transactions of the Royal Society London Series B*, **365**, 3599–3609.

See Also

`evodivparam`

Examples

```
## Not run:
if(require(ape)){

data(batcomm)
phy <- read.tree(text=batcomm$tre)
ab <- batcomm$ab[, phy$tip.label]
plot(qDT(phy, ab))
plot(qDT(phy, ab, q=seq(0, 10, length=20)))

}

## End(Not run)
```

QE   *Quadratic Entropy*

Description

Function QE calculates Rao's quadratic entropy within communities

Function discomQE calculates Rao's dissimilarities between communities

Usage

```
QE(comm, dis = NULL, formula = c("QE", "EDI"), scale = FALSE)

discomQE(comm, dis = NULL, structures = NULL, formula = c("QE", "EDI"))
```

Arguments

comm
: a data frame or a matrix with communities as rows and species as columns. Entries are abundances of species within communities. If presences/absences (1/0) are used a given species in a community of S species will be considered to have a relative abundance of 1/S.

dis
: either NULL or an object of class dist that contains the (functional or phylogenetic) distances among species. If NULL the Gini-Simpson index is used. See details.

formula
: either "QE" (default) or "EDI". See details.

scale
: a logical value indicating whether or not the diversity coefficient should be scaled by its maximal value over all species abundance distributions.

structures
: either NULL or a data frame that contains, in the *j*th row and the *k*th column, the name of the group of level *k* to which the *j*th community belongs. Communities in structures should be in the same order as in comm.

Details

If formula = "QE", the definition of the quadratic entropy is:

$$QE(\mathbf{p}_i, \mathbf{D}) = \sum_{k=1}^{S} \sum_{l=1}^{S} p_{k|i} p_{k|j} d_{kl}$$

where $\mathbf{p}_i = (p_{1|i}, ..., p_{k|i}, ..., p_{S|i})$ is the vector of relative species abundance within community *i*; S is the number of species; $\mathbf{D} = (d_{kl})$ is the matrix of (phylogenetic or functional) dissimilarities among species, and d_{kl} is the (phylogenetic or functional) dissimilarity between species *k* and *l*. For the calculations of dissimilarities between communities see the description of the apportionment of quadratic entropy in Pavoine et al. (2016) and references therein.

If formula = "EDI", the definition of the quadratic entropy is:

$$EDI(\mathbf{p}_i, \mathbf{D}) = \sum_{k=1}^{S} \sum_{l=1}^{S} p_{k|i} p_{k|j} \frac{d_{kl}^2}{2}$$

EDI stands for the Euclidean Diversity Index of Champely and Chessel (2002) (equation 3 in Pavoine et al. 2004). If EDI is used, the dissimilarities between communities calculated by discomQE are obtained as in equation 4 in Pavoine et al. (2004).

In both cases, if dis = NULL, the quadratic entropy is equal to Gini-Simpson entropy:

$$H_{GS}(\mathbf{p}_i) = 1 - \sum_{k=1}^{S} p_{k|i}^2$$

.

For using function `discomQE`, the Euclidean properties are expected for object `dis`. See function `is.euclid` of package ade4. These properties are not necessary for using function `QE`. Note that `discomQE` can be used if `dis = NULL`. In that case species are considered to be equidifferent (i.e. the dissimilarity between any two species is a constant; such dissimilarities have Euclidean properties).

Value

Function `QE` returns a data frame with communities as rows and the diversity within communities as columns.

If `structures` is `NULL`, function `discomQE` returns an object of class `dist`. Otherwise it returns a list of objects of class `dist`.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Gini, C. (1912) *Variabilita e mutabilita*. Universite di Cagliari III, Parte II.

Simpson, E.H. (1949) Measurement of diversity. *Nature*, **163**, 688.

Rao, C.R. (1982) Diversity and dissimilarity coefficients: a unified approach. *Theoretical Population Biology*, **21**, 24–43.

Champely, S. and Chessel, D. (2002) Measuring biological diversity using Euclidean metrics. *Environmental and Ecological Statistics*, **9**, 167–177.

Pavoine, S., Dufour, A.B., Chessel, D. (2004) From dissimilarities among species to dissimilarities among communities: a double principal coordinate analysis. *Journal of Theoretical Biology*, **228:**, 523–537.

Pavoine, S., Marcon, E., Ricotta, C. (2016) "Equivalent numbers" for species, phylogenetic, or functional diversity in a nested hierarchy of multiple scales. *Methods in Ecology and Evolution*, **7**, 1152–1163.

Examples

```
## Not run:
if(require(ade4)){

# First case study (community level, bird diversity):
data(ecomor, package="ade4")
# taxonomic dissimilarities between species
dtaxo <- dist.taxo(ecomor$taxo)
# quadratic entropy
QE(t(ecomor$habitat), dtaxo, formula="EDI")
QE(t(ecomor$habitat), dtaxo^2/2, formula="QE")
table.value(as.matrix(discomQE(t(ecomor$habitat), dtaxo, formula="EDI")))
EDIcom <- discomQE(t(ecomor$habitat), dtaxo, formula="EDI")
QEcom <- discomQE(t(ecomor$habitat), dtaxo^2/2, formula="QE")
QEcom
EDIcom^2/2
```

```
# display of the results
bird.QE <- QE(t(ecomor$habitat), dtaxo, formula="EDI")
dotchart(bird.QE$diversity, labels = rownames(bird.QE),
    xlab = "Taxonomic diversity", ylab="Habitats")

# Second case study (population level, human genetic diversity):
data(humDNAm, package="ade4")
# quadratic entropy
QE(t(humDNAm$samples), humDNAm$distances/2, formula="QE")
QE(t(humDNAm$samples), sqrt(humDNAm$distances), formula="EDI")
QEhumDNA.dist <- discomQE(t(humDNAm$samples),
humDNAm$distances/2, humDNAm$structures)
is.euclid(QEhumDNA.dist$communities)
is.euclid(QEhumDNA.dist$regions)

EDIhumDNA.dist <- discomQE(t(humDNAm$samples),
sqrt(humDNAm$distances), humDNAm$structures, formula="EDI")
is.euclid(EDIhumDNA.dist$communities)
is.euclid(EDIhumDNA.dist$regions)

QEhumDNA.dist$communities
EDIhumDNA.dist$communities^2/2

# display of the results
hum.QE <- QE(t(humDNAm$samples), humDNAm$distances/2, formula="QE")
dotchart(hum.QE$diversity, labels = rownames(hum.QE),
    xlab = "Genetic diversity", ylab="Populations")

}

## End(Not run)
```

---

qHdiv	*Biodiversity Measure that Includes Consistent Interspecific and Intraspecific Components*

Description

Function qHdiv calculates the parametric diversity index developed by Pavoine and Izsak (2014)

Usage

```
qHdiv(comm, C, q = 2)
```

Arguments

comm   a data frame or a matrix typically with communities as rows, species as columns and an index of abundance as entries.

C	a matrix that contains measures of the chosen intraspecific components (as defined in Pavoine and Izsak 2014) on the diagonal and measures of interspecific components off diagonal. These interspecific components reflect functional or phylogenetic similarities among species.
q	a positive numeric for the value of the parameter q.

Value

a vector with the diversity in each of the communities (same order as in `comm`).

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S., Izsak, J. (2014) New biodiversity measure that includes consistent interspecific and intraspecific components. *Methods in Ecology and Evolution*, **5**, 165–172.

See Also

[twoHmax](), [CFprop](), [CFbinary]()

Examples

```
## Not run:
if(require(ape)){

# First case study on phylogenetic diversity:
# Below is a phylogenetic tree in newick format
tre <-"((((((((sA:4,sB:1):1,sC:3):2,((sD:2,sE:1):1,sF:1):2):1,sG:7):1,sH:1):3,(sI:2,sJ:1):2):0;"
#The number of tips is kept in parameter n:
n<-10
# Next we need to obtain matrix CP (see Pavoine and Izsak 2014).
phyape <- read.tree(text = tre)
plot(phyape)
CP <- vcv(phyape)
WP <- diag(diag(CP))
# With this particular illustration, a maximizing vector,
# for 2H applied to CP, that does not contain any zero can be found.
# This maximizing vector can thus be obtained directly,
# instead of being estimated.
# Two equivalent equations have been given to obtain
# the maximizing vector in Appendix S1 of
# Pavoine and Izsak (2014).
# We use the first one below:

Pmax<-(solve(CP^2)%*%diag(CP))/sum(solve(CP^2)%*%diag(CP))
Pmax


# The second equation equivalently provides:
```

```
Z <- ((diag(1/sqrt(diag(CP))))%*%CP%*%(diag(1/sqrt(diag(CP)))))^2
Pmax<-(solve(WP)%*%solve(Z)%*%rep(1,n))/sum(solve(WP)%*%solve(Z))
Pmax

# Applied to our case study, the function twoHmax
# provides a good approximation of the maximizing vector:

twoHmax(CP)

# Second case study on the redundancy among variables:
data(rhone, package="ade4")
V <- rhone$tab
# First consider the covariances among the variables:
C <- cov(V)
# A vector that maximizes 2H applied to C is estimated
# as follows:
pmax_covariances <- twoHmax(C)$vector
dotchart(as.matrix(pmax_covariances))

# If we apply 2H only to the diagonal matrix with the variances
# of the variables, the vector that maximizes 2H is:
W <- diag(diag(C))
rownames(W)<-colnames(W)<-rownames(C)
pmax_variances <- twoHmax(W)$vector
dotchart(as.matrix(pmax_variances))

# If C contains the correlations among variables,
# a vector that maximizes 2H applied to C is estimated
# as follows:
C <- cor(V)
pmax_correlations <- twoHmax(C)$vector
dotchart(as.matrix(pmax_correlations))

# By attributing equal weights to the variables,
# 2H applied to the correlation matrix measures
# the number of effective variables:
# from 0 if all variables are completely correlated
# with each other to n if they are not correlated.
# Similarly, by attributing equal weights to the variables,
# 2H applied to the covariance matrix measures
# the effective amount of variation:
# from 0 if all variables are completely correlated
# with each other to n if they are not correlated
# and have similar variances.

#Even if the data set contains 15 variables,
# the effective number of variables is lower:
C <- cor(V)
equalproportions <- cbind.data.frame(rep(1/ncol(C), ncol(C)))
names(equalproportions) <- "equalprop"
equalproportions <- t(equalproportions)
qHdiv(equalproportions, C)
```

```
# When considering the covariances, instead of the correlations,
# the effective number of variables is even lower,
# indicating also an imbalance in the variances
# of the variables.
C <- cov(V)
qHdiv(equalproportions, C)


}

## End(Not run)
```

---

randEH	*Amount of Evolutionary History Preserved when Random Species are Saved*

Description

When branch lengths in an ultrametric phylogenetic tree are expressed as divergence times, the total sum of branch lengths in that tree expresses the amount of evolutionary history. The function randEH calculates the amount of evolutionary history preserved when *k* random species out of *n* original species are saved.

Usage

```
randEH(phyl, nbofsp, nrep = 10)
```

Arguments

phyl	an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust.
nbofsp	an integer indicating the number of species saved (*k*).
nrep	an integer indicating the number of random sampling.

Value

Function randEH returns a numeric vector with the amount of evolutionary history preserved by each random drawing of the *k* species to be saved.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr> with contributions of Stephane Dray

References

Nee, S. and May, R.M. (1997) Extinction and the loss of evolutionary history. *Science*, **278**, 692–694.

Pavoine, S., Ollier, S. and Dufour, A.-B. (2005) Is the originality of a species measurable? *Ecology Letters*, **8**, 579–586.

See Also

optimEH

Examples

```
## Not run:
if(require(ape) && require(adephylo)){

data(carni70, package = "adephylo")
tre <- read.tree(text=carni70$tre)
adiv:::EH(tre)
R <- adiv:::randEH(tre, 10, nrep=1000)
hist(R)

}

## End(Not run)
```

---

rare_Rao   *Functional Rarefaction for Species Abundance Data*

Description

The function `Rare_Rao` performs distance-based rarefaction curves using species abundance data. It finds the expected functional diversity (if functional distances between species are used) as a function of the sampling effort. Two approaches are available: an analytical solution, a resampling approach.

Usage

```
rare_Rao(comm, dis, sim = TRUE, resampling = 999, formula = c("QE", "EDI"))
```

Arguments

comm	a data frame or a matrix with samples as rows, species as columns, and abundance or frequency as entries. If presences/absences (1/0) are given, the relative abundance of a given species in a community of S species will be considered equal to 1/S.
dis	an object of class `dist` containing pairwise distances among species. The distance matrix should be squared Euclidean or simply Euclidean (see definition and details in Appendix S1 of Ricotta et al. (2012)). Alternatively, if `dis` is NULL, the Gini-Simpson index is used.
sim	a logical; if `TRUE`, the resampling approach is used; if `FALSE`, the analytical solution is given.
resampling	a numeric; number of times data are resampled to calculate the mean functional rarefaction curve (used if `sim=TRUE`).
formula	either `"QE"` (default) or `"EDI"`. See details.

Details

If formula = "QE", the definition of the quadratic entropy is:

$$QE(\mathbf{p}_i, \mathbf{D}) = \sum_{k=1}^{S} \sum_{l=1}^{S} p_{k|i} p_{k|j} d_{kl}$$

where $\mathbf{p}_i = (p_{1|i}, ..., p_{k|i}, ..., p_{S|i})$ is the vector of relative species abundance within sample i; S is the number of species; $\mathbf{D} = (d_{kl})$ is the matrix of (phylogenetic or functional) dissimilarities among species, and d_{kl} is the (phylogenetic or functional) dissimilarity between species k and l.

If formula = "EDI", the definition of the quadratic entropy is:

$$EDI(\mathbf{p}_i, \mathbf{D}) = \sum_{k=1}^{S} \sum_{l=1}^{S} p_{k|i} p_{k|j} \frac{d_{kl}^2}{2}$$

EDI stands for the Euclidean Diversity Index of Champely and Chessel (2002) (equation 3 in Pavoine et al. 2004).

In both cases, if dis = NULL, the quadratic entropy is equal to Gini-Simpson entropy:

$$H_{GS}(\mathbf{p}_i) = 1 - \sum_{k=1}^{S} p_{k|i}^2$$

Value

If sim = TRUE, the function returns a data frame containing the Expected Rao Quadratic Entropy (column 'ExpRao'), the limits of the 95% Confidence Interval (columns 'LeftIC' and 'RightIC') for each subsample dimension (M) out of the total set of samples (N). If sim = FALSE, the function returns a data frame containing the analytical solution for the Expected Rao Quadratic Entropy (column 'ExpRao') for each subsample dimension (M) out of the total set of samples (N).

Author(s)

Giovanni Bacaro and Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C., Pavoine, S., Bacaro, G., Acosta, A. (2012) Functional rarefaction for species abundance data. *Methods in Ecology and Evolution*, **3**, 519–525.

Champely, S. and Chessel, D. (2002) Measuring biological diversity using Euclideanmetrics. *Environmental and Ecological Statistics*, **9**, 167–177.

Pavoine, S., Dufour, A.B., Chessel, D. (2004) From dissimilarities among species to dissimilarities among communities: a double principal coordinate analysis. *Journal of Theoretical Biology*, **228**, 523–537.

See Also

QE

Examples

```
## Not run:
if(require(ade4)){

data(aviurba, package="ade4")

# Trait-based distances between bird species:
distances<-dist.ktab(ktab.list.df(list(aviurba$traits)), type = "N")
# The distances should be squared Euclidean;
# note that Euclidean distances can be used
# as they also are squared Euclidean.

# Species abundances in sites
abundances<- aviurba$fau

# Rarefaction of functional diversity
rare_Rao(abundances, distances, sim = TRUE, resampling = 100)
rare_Rao(abundances, distances, sim = FALSE)

}

## End(Not run)
```

RDMCCP16   *Theoretical Data Set used in Ricotta et al. (2016)*

Description

Hypothetical communities C1-C9 composed of nine species S1-S9 with varying abundances divided into three groups of three species (S1-S3, S4-S6 and S7-S9) (say, legumes, herbs and forbs). For simplicity, all species within the same group are functionally identical to each other, while two species belonging to different groups are always maximally dissimilar (Ricotta et al. 2016).

Usage

```
data("RDMCCP16")
```

Format

A list of two objects:

ab, a matrix with communities as rows, species as columns and abundance values as entries.

dis, an object of class dist that contains theoretical dissimilarities between species.

Source

Table 1 in Ricotta et al. (2016)

References

Ricotta, C., de Bello, F., Moretti, M., Caccianiga, M., Cerabolini, B.E., Pavoine, S. (2016). Measuring the functional redundancy of biological communities: A quantitative guide. *Methods in Ecology and Evolution*, **7**, 1386–1395.

Examples

```
data(RDMCCP16)
uniqueness(RDMCCP16$ab, as.dist(RDMCCP16$dis))
```

Rentropy   *Pavoine et al. (2017) functional or phylogenetic entropy*

Description

Function `Rentropy` calculates Pavoine et al. (2017) functional or phylogenetic R entropy within communities; this index is closely related to Rao's quadratic entropy

Usage

```
Rentropy(comm, dis = NULL, scale = FALSE)
```

Arguments

comm	a data frame or a matrix with communities as rows and species as columns. Entries are abundances of species within sites.
dis	either NULL or an object of class dist that contains the (functional or phylogenetic) distances among species. If NULL species are said equidistant with a distance of 1 between any two species and a distance of 0 between a species and itself.
scale	a logical value indicating whether or not the diversity coefficient should be scaled by its maximal value over all species abundance distributions.

Details

The definition of the R entropy is:

$$R(\mathbf{p}_i, \mathbf{D}) = \sum_{k=1}^{S} \sum_{l=1}^{S} \sqrt{p_{k|i}} \sqrt{p_{k|j}} d_{kl}$$

where $\mathbf{p}_i = (p_{1|i}, ..., p_{k|i}, ..., p_{S|i})$ is the vector of relative species abundance within community i; S is the number of species; $\mathbf{D} = (d_{kl})$ is the matrix of (phylogenetic or functional) dissimilarities among species, and d_{kl} is the (phylogenetic or functional) dissimilarity between species k and l.

Value

Function `Rentropy` returns a data frame with communities as rows and the R entropy within communities as columns.

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S., Bonsall, M.B., Dupaix, A., Jacob, U., Ricotta, C. (2017) From phylogenetic to functional originality: guide through indices and new developments. *Ecological Indicators*, **82**, 196–205.

Examples

```
## Not run:
if(require(ade4)){

data(ecomor, package="ade4")
dtaxo <- dist.taxo(ecomor$taxo)
bird.R <- Rentropy(t(ecomor$habitat), dtaxo^2/2)
dotchart(bird.R$diversity, labels = rownames(bird.R))


}

## End(Not run)

## Not run:
if(require(ape)){

data(batcomm)
phy <- read.tree(text=batcomm$tre)
dphy <- as.dist(cophenetic(phy))/2
ab <- batcomm$ab[, phy$tip.label]
bat.R <- Rentropy(ab, dphy)
dotchart(bat.R$diversity, labels = rownames(bat.R), xlab = "Entropy")


}

## End(Not run)
```

---

rlqESLTP   *Linking Patterns in Phylogeny, Traits, Abiotic Variables and Space*

Description

An extention of the RLQ approach to identify potential environmental filters in species' traits in a phylogenetic and spatial context.

Usage

```
rlqESLTP(dudiE, dudiS, dudiL, dudiT, dudiP, ...)

## S3 method for class 'rlqESLTP'
plot(x, which = NULL, phyl = NULL,
    xy = NULL, traits = NULL, env = NULL, type = NULL,
    ax = 1, disp = c("dots", "bars", "grid"), ...)

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

Arguments

dudiE
: an object of class dudi of package ade4 (e.g. a principal component applied to environmental variables or a principal coordinate analysis applied to environmental distances between plots). Plots must be weighted as in dudiL. dudiE can be set to NULL if dudiS is not.

dudiS
: an object of class dudi of package ade4 (e.g. an eigenvector decomposition applied to a spatial neighbourhood matrix (see Pavoine et al. 2011 for exemples)). Plots must be weighted as in dudiL. dudiS can be set to NULL if dudiE is not.

dudiL
: an object of class dudi obtained with function dudi.coa of ade4 applied to a plot (rows) by species (columns) data frame. dudiL cannot be set to NULL.

dudiT
: an object of class dudi of package ade4 (e.g. a principal component analysis applied to functional traits of species or a principal coordinate analysis applied to functional distances between species). Species must be weighted as in dudiL. dudiT can be set to NULL if dudiP is not.

dudiP
: an object of class dudi of package ade4 (e.g. a principal coordinate analysis applied to phylogenetic distances between species). Species must be weighted as in dudiL. dudiP can be set to NULL if dudiT is not.

x
: an object of class rlqESTP (obtained with function rqlESTP).

object
: an object of class rlqESTP (obtained with function rqlESTP).

which
: a character that might be "E" (in that case arguments env and type must be given); "S" (in that case argument xy must be given); "T" (in that case arguments traits and type must be given); or "P" (in that case argument phyl must be given).

phyl
: an object inheriting the class phylo (see package ape), phylo4 (see package phylobase) or hclust.

xy
: a data frame with two columns giving the coordinates of the plots (longitude, latitude in that order).

traits
: a list of data frames for the traits, each data frame contains variables from a single statistical type (see argument type below). NAs are allowed. However, in that case, the calculations depend, for each trait, on the subset of available data only.

env
: a list of data frames for the environmental variables, each data frame contains variables from a single statistical type (see argument type below). NAs are

allowed. However, in that case, the calculations depend, for each variable, on the subset of available data only.

type   a vector that provides the type of each table to be analysed in env (if environe-mental variables should be displayed) or traits (if the focus is on species' traits). The possible types are ″Q″ (quantitative, numeric), ″O″ (ordinal), ″N″ (nominal), ″D″ (dichotomous), ″F″ (fuzzy, or expressed as a proportion), ″B″ (multichoice nominal variables, coded by binary columns), ″C″ (circular). Values in type must be in the same order as in env or traits.

ax   a numeric indicating the axis of interest.

disp   a string indicating which method to use to display species scores in front of the phylogeny: one of ″dots″ (for a dot plot), ″bars″ (for a barplot), and ″grid″ (for a gridplot for vizualizing scores by a color gradient).

...   further arguments passed to or from other methods.

Value

Function rlqESLTP returns an object of classes rlqESLTP, rlq and dudi. It is a list of 26 objects:

tab   a data frame. Crossed Table (CT): crossing the columns of the merged trait and phylogenetic table with those of the merged environmental and spatial table.

cw   a vector of numerics. Weights attributed to the columns of the merged trait and phylogenetic table.

lw   a vector of numerics. weights attributed to the columns of the merged environ-mental and spatial table.

eig   a vector of numerics. The vector of eigenvalues.

rank   an integer. The total number of axes in the analysis.

nf   a numeric. The number of axes kept.

c1   a data frame. Principal axes. Normed scores for the columns of merged trait and phylogenetic table.

co   a data frame. Scores for the columns of merged trait and phylogenetic table.

l1   a data frame. Principal axes. Normed scores for the columns of merged envi-ronmental and spatial table.

li   a data frame. Scores for the columns of merged environmental and spatial table.

call   call

lQ   data frame. Scores for the species.

mQ   data frame. Normed scores for the species.

lR   data frame. Scores of the plots.

mR   data frame. Normed scores for the plots.

aQ   data frame. Correlations trait/phylogeny axes / coinertia axes.

aR   data frame. Correlations environment/space axes / coinertia axes.

lR_givenE   data frame. Contributions of environmental information to plot scores.

lR_givenS   data frame. Contributions of spatial information to plot scores.

lQ_givenT	data frame. Contributions of trait information to species scores.
lQ_givenP	data frame. Contributions of phylogenetic information to species scores.
row.w	a vector. Weights attributed to plots.
col.w	a vector. Weights attributed to species.
dudiL	object of class dudi (dudiL).
dudiR	object of class dudi (merging information on environment and space).
dudiQ	object of class dudi (merging information on traits and phylogeny).

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Pavoine, S., Vela, E., Gachet, S., de Belair, G., Bonsall, M.B. (2011) Linking patterns in phylogeny, traits, abiotic variables and space: a novel approach to linking environmental filtering and plant community assembly. *Journal of Ecology*, **99**, 165–175.

Examples

```
## Not run:
if(require(ade4) && require(adegraphics) && require(ape)){
O <- adegpar()$plabels$optim
adegpar("plabels.optim" = TRUE)

data(mafragh, package="ade4")
xy <- mafragh$xy
#The object that defines the neighbourhood between plots is in
mneig <- mafragh$neig
#mneig is an object of class neig
s.label(mafragh$xy, nb = mafragh$nb, paxes.draw = FALSE)

#The environmental variables are in
env <- mafragh$env
names(env)
#The abundance data are in
flo <- mafragh$flo
# Adjustement of the writing of species names
names <- gsub(" ", "_", mafragh$spenames[colnames(flo), 1])
for(i in 1:26){
names <- gsub(LETTERS[i], letters[i], names)
}
names <- gsub("alisma_plantago", "alisma_plantago_aquatica", names)
colnames(flo) <- names
#The data on traits are in
traits <- lapply(mafragh$traits, function(x) x[colnames(flo), , drop=FALSE])
#The phylogenetic tree is in
phy <- read.tree(text=mafragh$tre)
plot(phy)
```

```
#Traits are separated by statistical type. The object traits is a list of data frames.
tabBinary <- prep.binary(traits$tabBinary, c(3, 4))
tabQuantitative <- traits$tabQuantitative
tabCircular <- prep.circular(traits$tabCircular, 1, 12)
tabOrdinal <- traits$tabOrdinal

#Analyses:
coaflo <- dudi.coa(flo, scan = FALSE, nf = 55)
vecspa <- scores.neig(mneig)
pcaspa <- dudi.pca(vecspa, coaflo$lw, scan = FALSE, nf = 96)
#We first removed environmental variables
env <- env[-(8:10)]
env[4:8] <- log(env[4:8])
pcaenv <- dudi.pca(env, coaflo$lw, scale = FALSE, scan = FALSE, nf = 8)
disT <- dist.ktab(ktab.list.df(list(tabBinary,
tabOrdinal[c("Spikiness", "Hairy leaves")])), c("B", "O"))
# The definition of the functional distances between species
# were slightly different in Pavoine et al. (2011).

pcotraits <- dudi.pco(disT, coaflo$cw, full = TRUE)
pcophy <- dudi.pco(sqrt(as.dist(cophenetic(phy)[names(flo), names(flo)])/2), coaflo$cw, full = TRUE)


rlqmix <- rlqESLTP(pcaenv, pcaspa, coaflo, pcotraits, pcophy, scan = FALSE, nf = 2)
barplot(rlqmix$eig)
rlqmix$eig[1]/sum(rlqmix$eig)
plot(rlqmix, xy=xy, ax=1, wh="S")
plot(rlqmix, phy=phy, ax=1, wh="P")
plot(rlqmix, traits=tabBinary, ax=1, type="B", wh="T")
plot(rlqmix, traits=tabOrdinal[2:3], ax=1, type="O", wh="T")
plot(rlqmix, env=env, ax=1, type="Q", wh="E")

adegpar("plabels.optim" = O)
}

## End(Not run)
```

---

rockfish	*Rockfish Phylogenetic Diversity in Southern California Bight*

Description

The data set was analyzed in Pavoine et al. (2009) to determine whether the observed decrease in the number of individuals and species in the Southern California Bight through years due to fishing activities was accompanied by changes in the phylogenetic structure of the community.

Usage

```
data("rockfish")
```

Format

The format is a list of 3 objects:

`tre`, a string: the phylogenetic tree in newick format.

`fau`, a data frame with years as rows, species as columns (genus: Sebastes), and abundance as entries.

`traits`, a data frame with species as rows (same as in `fau`) and two variables as columns: `MaxSize`, the maximum body size; and `Vul`, an index of species' vulnerability.

Details

The abundance data were obtained from the Marine Recreational Fishery Statistics Survey (MRFSS). We considered the compositions of rockfish assemblages caught by party and charter boats with hooks and lines from 1980-1986, 1993-1994, 1996, 1998-2007. The phylogenetic tree was obtained from Hyde and Vetter (2007).

Source

Appendixes of Pavoine et al. (2009).

References

Pavoine, S., Love, M., Bonsall, M. (2009) Hierarchical partitioning of evolutionary and ecological patterns in the organization of phylogenetically-structured species assemblages: application to rockfish (genus: *Sebastes*) in the Southern California Bight. *Ecology Letters*, **12**, 898–908.

Hyde, J.R. and Vetter, R.D. (2007). The origin, evolution, and diversification of rockfishes of the genus Sebastes (Cuvier). *Molecular Phylogenetics and Evolution*, **44**, 790–811.

Examples

```
## Not run:
if(require(ape)){
data(rockfish)
phy <- read.tree(text=rockfish$tre)
plot(phy)
}

## End(Not run)
```

RP15EI   *Theoretical Data Set used in Ricotta and Pavoine (2015) in Ecological Indicators*

Description

The data set was used in Ricotta and Pavoine (2015) to show how the coefficients of multi-site dissimilarity that they developed can be applied to data.

Usage

```
data("RP15EI")
```

Format

The format is a list of 4 data frames. Each data frame gives the presence (1) and absence (0) of 10 species (columns, S1-S10) in 8 plots (rows P1-P8). These data are characterized by different levels of nestedness and turnover in species compositions between sites.

M1: Intermediate nestedness and turnover.

M2: Perfectly nested pattern.

M3: Very high species turnover.

M4: Random configuration.

Source

Figure 1 of Ricotta and Pavoine (2015)

References

Ricotta, C. and Pavoine, S. (2015) A multiple-site dissimilarity measure for species presence/absence data and its relationship with nestedness and turnover. *Ecological Indicators*, **54**,203–206.

Examples

```
data(RP15EI)
betastatjac(RP15EI$M1)
```

RP15JVS   *Theoretical Data Set used in Ricotta and Pavoine (2015) in Journal of Vegetation Science*

Description

The data set was used in Ricotta and Pavoine (2015) to illustrate the relationships between the coefficients of similarity between communities that they developed. These data represent an artificial ecological gradient.

Usage

```
data("RP15JVS")
```

Format

The format is a list of four objects:

ab, a data frame with communities as rows, species as columns, and number of indivuduals as entries.

D1, a data frame of pairwise dissimilarities between species. In D1, dissimilarities reflect the species ecological differences along the artificial gradient of table ab: interspecies dissimilarities were set roughly proportional to the distance between the locations of the species optima and to the difference between their optimal abundances (see Ricotta and Pavoine 2015 for details).

D2, a data frame of pairwise dissimilarities between species. In D2, dissimilarities were randomly assigned using an even distribution.

D3, a data frame of pairwise dissimilarities between species. In D3, dissimilarities were uniformly set to 2/3.

Source

Appendixes of Ricotta and Pavoine (2015)

References

Ricotta, C. and Pavoine, S. (2015) Measuring similarity among plots including similarity among species: An extension of traditional approaches. *Journal of Vegetation Science*, **26**, 1061–1067

Examples

```
data(RP15JVS)
dissABC(RP15JVS$ab, RP15JVS$D1, method="J", option=1)
```

RPP16EE   *Theoretical Data Set used in Ricotta et al. (2016) in Ecology and Evolution*

Description

The data set was used in Ricotta et al. (2016) to show how the coefficients of plot-to-plot dissimilarity that they developed can be applied to data.

Usage

```
data("RPP16EE")
```

Format

The format is a list of 2 objects:

Com, a data frame. Artificial data table composed of 15 species (S1-S15) (columns) and 9 plots (P1-P9) (rows).

Dis, a data frame with the artificial dissimilarities between species.

Source

Appendixes 1 and 2 of Ricotta et al. (2016)

References

Ricotta, C., Podani, J., Pavoine, S. (2016) A family of functional dissimilarity measures for presence and absence data. *Ecology and Evolution*, **6**, 5383–5389.

Examples

```
data(RPP16EE)
RPP16EE$Com
Jac(RPP16EE$Com)
## Not run:
if(require(ade4) && require(adegraphics)){
oldparamadeg <- adegpar()
adegpar("plegend.drawKey" = FALSE)
table.value(RPP16EE$Com)
adegpar(oldparamadeg)
# In this graph, black squares indicate
# which species (S1-S15) are present in which plot (P1-P9)
}

## End(Not run)
```

RutorGlacier	*Functional and phylogenetic composition of plant communities along a primary succession on glacial deposits.*

Description

59 plots, each of about 25 m2 in size, were sampled above the tree line at the foreland of the Rutor Glacier (Italy). The abundance of plant species were collected by Caccianiga et al. (2006); Ricotta et al. (2016) established trait data using Grime's (1974) plant strategy theory, Caccianiga et al. (2006) established morpho-functional traits, and Ricotta et al. (2015) the phylogenetic tree using the dated Daphne phylogeny (Durka and Michalski 2012) for European flora.

Usage

```
data("RutorGlacier")
```

Format

`RutorGlacier` is a list of five components:

`Abund`, a data frame with plots as rows, species as columns and abundance of species in plots as entries.

`Traits`, a data frame with species as rows and three traits (named C, S, and R) as columns. Species were classified in terms of Grime's (1974) plant strategy theory, as competitors (C), stress tolerators

(S) and ruderals (R) with fuzzy-coded values in the range 0-100, such that the sum of C+S+R was equal to 100.

`Traits2`, a data frame with species as rows and morpho-functional traits as columns. Traits are: ″`C_height`″ = canopy height (mm), ″`LDMC`″ = leaf dry matter content (%), ″`LDW`″ = leaf dry weight (mg), ″`SLA`″ = specific leaf area (mm^2 mg^-1), ″`LNC`″ = leaf nitrogen content (%), ″`LCC`″ = leaf carbon content (%), ″`Flowering_period`″ = flowering period length (months), ″`Flowering_start`″ = flowering start (1 to 6 denote March to August, respectively), ″`Lateral_spread`″ = lateral spread (1 = plant short-lived, 2 = loosely tufted ramets (graminoids), compactly tufted, lacking thickened rootstock (non-graminoids), 3 = compactly tufted ramets (gram.), compactly tufted with thickened rootstock (non-gram.), 4 = shortly creeping (<40 mm), 5 = creeping (40-79 mm), 6 = widely creeping (>79 mm)), CHMH = canopy height/maximum height ratio, Succulence = succulence index, `Clonality` = clonality (0 = sparse individuals, 1 = limited clonal growth, 2 = extensive clonal patches), ″`NecroPersi`″ = necromass persistence (0 = necromass absent, 1 = fibrous sheaths, 2 = entire persistent sheaths or leaves).

`TreeNW`, a phylogenetic tree in newick format for all plant species in `Abund` and `traits`.

`Fac`, a vector of strings defining in which successional stage each plot belongs to: "early" = early-successional stage, "mid" = mid-successional stage and "late" = late-successional stage.

References

Caccianiga, M., Luzzaro, A., Pierce, S., Ceriani, R.M., Cerabolini, B.E.L. (2006) The functional basis of a primary succession resolved by CSR classification. *Oikos*, **112**, 10–20.

Durka, W., Michalski, S.G. (2012) Daphne: a dated phylogeny of a large European flora for phylogenetically informed ecological analyses. *Ecology*, **93**, 2297.

Grime, J.P. (1974) Vegetation classification by reference to strategies. *Nature*, **250**, 26–31.

Ricotta, C., Bacaro, G., Caccianiga, M., Cerabolini, B.E.L., Moretti, M. (2015) A classical measure of phylogenetic dissimilarity and its relationship with beta diversity. *Basic and Applied Ecology*, **16**, 10–18.

Ricotta, C., de Bello, F., Moretti, M., Caccianiga, M., Cerabolini, B.E.L., Pavoine, S. (2016) Measuring the functional redundancy of biological communities: a quantitative guide. *Methods in Ecology and Evolution*, **7**, 1386–1395.

Ricotta, C., Laroche, F., Szeidl, L., Pavoine, S. (2020) From alpha to beta functional and phylogenetic redundancy. *Methods in Ecology and Evolution*. In press.

Examples

```
## Not run:
if(require(ape) && require(ade4)){
data(RutorGlacier)
phy <- read.tree(text=RutorGlacier$TreeNW)
plot(phy)
ab <- RutorGlacier$Abund[, phy$tip.label]
plot(abgevodivparam(phy, ab, q=0:4))

# Phylogenetic dissimilarities between plots
# (Ricotta et al. 2020)
Dp <- DP(phy, ab, tol=0.00001)
```

```
pcoDp <- dudi.pco(sqrt(Dp), full=TRUE)
s.class(pcoDp$li, as.factor(RutorGlacier$Fac))

# Phylogenetic beta uniqueness (Ricotta et al. 2020)
Up <- betaTreeUniqueness(phy, ab, tol=0.00001)
# Average uniqueness between two plots at each successional stage
fac <- as.factor(RutorGlacier$Fac)
mean(Up[fac == "early", fac == "early"])
mean(Up[fac == "mid", fac == "mid"])
mean(Up[fac == "late", fac == "late"])
}

## End(Not run)
```

---

speciesdiv	*Indices of Species Diversity*

Description

The function speciesdiv calculates diversity indices that rely on relative or absolute species abundance.

Usage

```
speciesdiv(comm, method = "full", tol = 1e-8)
```

Arguments

comm   a data frame or a matrix typically with communities as rows, species as columns and presence/absence or abundance as entry. Note that with presence/absence (0/1) data, only species richness will be calculated correctly.

method   a string or a vector of strings: one or several of "richness", "GiniSimpson", "Simpson", "Shannon", "Margalef", "Menhinick", "McIntosh", "full". See details.

tol   a tolerance threshold (a value between -tol and tol is considered equal to zero)

Details

Let S_i be the number of species in community i, n_{ij} be the absolute abundance of species j in community i, N_i the sum of all species abundance in community i ($N_i = \sum_j n_{ij}$; the sum of row i in comm), p_{ij} the relative abundance of species j in community i ($p_{ij} = n_{ij}/N_i$). If method="richness", the diversity index is the number of species. If method="GiniSimpson", the diversity index is that of Gini (1912) and Simpson (1949): $1 - \sum_j p_{ij}^2$. If method="Simpson", the diversity index is (Simpson 1949): $1/\sum_j p_{ij}^2$. If method="Shannon", the diversity index is that of Shannon (1948) with neperian logarithm: $-\sum_j p_{ij}log(p_{ij})$. If method="Margalef", the diversity index is that of Margalef (1972): $(S_i - 1)/log(N_i)$. If method="Menhinick", the diversity index is that of Menhinick (1964): $S_i/\sqrt{N_i}$. If method="McIntosh", the diversity index is that of McIntosh (1967): $(N_i - \sqrt{\sum_j n_{ij}^2})/(N_i - \sqrt{N_i})$. If one of the strings is "full", then all indices are calculated.

Value

Function `speciesdiv` returns a matrix with communities as rows and the diversity indices as columns.

Author(s)

Sandrine Pavoine `<sandrine.pavoine@mnhn.fr>`

References

Gini, C. (1912) Variabilita e mutabilita. Studi economicoaguridici delle facoltta di giurizprudenza dell, Universite di Cagliari III, Parte II.

Magurran, A.E. (2004) Measuring biological diversity. Blackwell Publishing, Oxford, U.K.

Margalef, R. (1972) Homage to Evelyn Hutchinson, or why is there an upper limit to diversity? *Transactions of the Connecticut Academy of Arts and Sciences*, **44**, 211–235.

McIntosh, R.P. (1967) An index of diversity and the relation of certain concepts to diversity. *Ecology*, **48**, 392–404.

Menhinick, E.F. (1964) A Comparison of some species-individuals diversity indices applied to samples of field insects. *Ecology*, **45**, 859–861.

Shannon, C.E. (1948) A mathematical theory of communication. *Bell System technical journal*, **27**, 379–423, 623–656.

Simpson, E.H. (1949) Measurement of diversity. *Nature*, **163**, 688.

Examples

```
data(batcomm)
ab <- batcomm$ab
speciesdiv(ab)
```

---

specieseve   *Indices of Species Evenvess*

Description

The function `specieseve` calculates evenness indices that rely on relative or absolute species abundance.

Usage

```
specieseve(comm, method = "full", tol = 1e-8)
```

Arguments

`comm`	a data frame or a matrix typically with communities as rows, species as columns and abundance as entry.
`method`	a string or a vector of strings: one or several of "GiniSimpson", "Simpson", "Shannon", "Heip", "McIntosh", "SmithWilson", "full". See details.
`tol`	a tolerance threshold (a value between `-tol` and `tol` is considered equal to zero)

Details

Let S_i be the number of species in community i, n_{ij} be the absolute abundance of species j in community i, N_i the sum of all species abundance in community i ($N_i = \sum_j n_{ij}$; the sum of row i in `comm`), p_{ij} the relative abundance of species j in community i ($p_{ij} = n_{ij}/N_i$). If `method="GiniSimpson"`, the evenness index is that associated with Gini (1912) and Simpson (1949) diversity index: $(1 - \sum_j p_{ij}^2) * S_i/(S_i - 1)$. If `method="Simpson"`, the evenness index is (Simpson 1949; Magurran 2004): $(1/\sum_j p_{ij}^2)/S_i$. If `method="Shannon"`, the evenness index is that associated with Shannon (1948) diversity index with neperian logarithm: $(-\sum_j p_{ij}ln(p_{ij}))/ln(S_i)$. If `method="Heip"`, the evenness index is that of Heip (1974) (Magurran 2004): $[exp(-\sum_j p_{ij}log(p_{ij})) - 1]/(S_i - 1)$. If `method="McIntosh"`, the evenness index is that of Pielou (1975) associated with McIntosh (1967) index of diversity: $(N_i - \sqrt{\sum_j n_{ij}^2})/(N_i - N_i/\sqrt{S_i})$. If `method="SmithWilson"`, the Smith and Wilson (1996) evenness index is calculated (Magurran 2004): $1 - [(2/\pi)(arctan(\sum_{j=1}^{S_i}(logn_{ij} - \sum_{k=1}^{S_i} logn_{ik}/S_i)^2/S_i))]$. The function uses neperian logarithm for all indices. If one of the strings is "full", then all indices are calculated.

Value

Function `specieseve` returns a matrix with communities as rows and the evenness indices as columns.

Author(s)

Sandrine Pavoine `<sandrine.pavoine@mnhn.fr>`

References

Gini, C. (1912) *Variabilita e mutabilita*. Studi economicoaguridici delle facoltta di giurizprudenza dell, Universite di Cagliari III, Parte II.

Heip, C. (1974) A new index measuring evenness. *Journal of the Marine Biological Association UK*, **54**, 555–557.

Magurran, A.E. (2004) *Measuring biological diversity*. Oxford, UK: Blackwell Publishing.

McIntosh, R.P. (1967) An index of diversity and the relation of certain conepts to diversity. *Ecology*, **48**, 392–404.

Pielou, E.C. (1975) *Ecological diversity*. New York: Wiley InterScience.

Shannon, C.E. (1948) A mathematical theory of communication. *Bell System technical journal*, **27**, 379–423, 623–656.

Simpson, E.H. (1949) Measurement of diversity. *Nature*, **163**, 688.

Smith, B. and Wilson, J.B. (1996) A consumer's guide to evenness measures. *Oikos*, **76**, 70–82.

Examples

```
data(batcomm)
ab <- batcomm$ab
specieseve(ab)
```

---

treeUniqueness *Community-level phylogenetic (or functional) redundancy*

Description

The function `treeUniqueness` calculates community-level phylogenetic (or tree-based) redundancy taking into account the branching pattern of the underlying phylogenetic tree (or any other tree, like a functional dendrogram).

Usage

```
treeUniqueness(phyl, comm, index =
    c("richness", "GiniSimpson", "Shannon"), tol = 0.001)
```

Arguments

phyl	an object inheriting the class `phylo` (see package ape), `phylo4` (see package phylobase), or `hclust`.
comm	a data frame or a matrix typically with communities as rows, species as columns and presence/absence (1/0) or an index of abundance as entries. Species should be labeled as in the tree named `phyl` where they are the tips.
index	a string. `treeUniqueness` computes tree-based (phylogenetic) species richness ("richness") or the tree-based analogue of the Shannon (shannon) or Gini-Simpson diversity ("GiniSimpson"); "richness" is assumed as the default value.
tol	a numeric. A value between -tol and tol is considered as zero. See details.

Details

The tolerance threshold `tol` is particularly important if your tree is not exactly ultrametric due to approximation problems. In that case, the distance from tip to root varies according to the tip considered, although it should not (variations are due to approximation problems). A difference smaller than tol in the distance to root for two species will thus be considered as null.

Value

An object of class `data.frame` is returned containing the following statistics:

DK	the present-day diversity of all plots in the data frame.
DP	the tree-based (phylogenetic) diversity D_P of all plots in the data frame obtained by averaging the diversities D_k over the corresponding tree periods after rescaling the tip-to-root length of the tree to unit (for details, see Ricotta et al. 2018).

U	tree-based (phylogenetic) uniqueness calculated as the ratio between tree-based (phylogenetic) diversity and present-day diversity D_P/D_K.
R	tree-based (phylogenetic) redundancy, calculated as $1 - U$.

Author(s)

Ricotta et al. (2018) with modifications by Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

References

Ricotta, C., Bacaro, G., Caccianiga, M., Cerabolini, B.E.L., Pavoine, S. (2018) A new method for quantifying the phylogenetic redundancy of biological communities. *Oecologia*, **186**, 339–346.

See Also

uniqueness

Examples

```
## Not run:
if(require(ape)){
data(rockfish)
phy <- read.tree(text = rockfish$tre)
R <- treeUniqueness(phy, rockfish$fau, index = "Shannon")
}

## End(Not run)
```

---

twoHmax   *Maximization of the Diversity Index TwoH*

Description

The R function twoHmax maximizes function 2H (Pavoine and Izsak 2014) for a given matrix **C** of (functional or phylogenetic) similarities between species. It is based on function divcmax of the ade4 package in R. As function divcmax of package ade4, function twoHmax uses an optimization technique based on Rosen's projection gradient algorithm and is verified using the Kuhn-Tucker conditions.

Usage

```
twoHmax(C, epsilon = 1e-08, smooth = TRUE, comment = FALSE)
```

Arguments

C	a matrix that contains measures of the chosen intraspecific components (see functions CFprop, multiCFprop, CFbinary, multiCFbinary) on the diagonal and measures of interspecific components off diagonal. These interspecific components reflect functional or phylogenetic similarities among species.
epsilon	a numeric tolerance threshold: a frequency is non null if it is higher than epsilon.
smooth	a logical value: if TRUE, the estimated positive proportions are checked and re-estimated with exact matrix products.
comment	a logical value indicating whether or not comments on the optimization technique should be printed.

Value

A list of two objects:

value	the maximum value of index 2H (see function qHdiv);
vector	a data frame with the vector pmax that maximizes index 2H (see function qHdiv).

Author(s)

Sandrine Pavoine <sandrine.pavoine@mnhn.fr>

The code is a modification of function divcmax of package ade4 written by Stephane Champely.

References

Pavoine, S. and Izsak, J. (2014) New biodiversity measure that includes consistent interspecific and intraspecific components. *Methods in Ecology and Evolution*, **5**, 165–172.

See Also

qHdiv, CFprop, CFbinary

Examples

```
## Not run:
if(require(ape)){

tre <-"((((((sA:4,sB:1):1,sC:3):2,((sD:2,sE:1):1,sF:1):2):1,sG:7):1,sH:1):3,(sI:2,sJ:1):2):0;"
#The number of tips is kept in parameter n:
n<-10
# Next we need to obtain matrix CP.
phyape <- read.tree(text = tre)
plot(phyape)
CP <- vcv(phyape)
WP <- diag(diag(CP))
# With this particular illustration, a maximizing vector,
# for 2H used with CP, that does not contain any zero can be found.
# This maximizing vector can thus be obtained directly,
# instead of being estimated. Two equivalent equations
```

```
# have been given to obtain the maximizing vector in
# Appendix S1 of Pavoine and Izsak (2014).
# We use the first one below
Pmax<-(solve(CP^2)%*%diag(CP))/sum(solve(CP^2)%*%diag(CP))
Pmax

# The second equation equivalently provides
Z <- ((diag(1/sqrt(diag(CP))))%*%CP%*%(diag(1/sqrt(diag(CP)))))^2
Pmax<-(solve(WP)%*%solve(Z)%*%rep(1,n))/sum(solve(WP)%*%solve(Z))
Pmax

# Applied to our case study, the function twoHmax provides good approximations
twoHmax(CP)

# Redundancy among variables:
data(rhone, package="ade4")
V <- rhone$tab
# First consider the covariances among the variables:
C <- cov(V)
# A vector that maximizes 2H applied to C is estimated as follows:
pmax_covariances <- twoHmax(C)$vector
dotchart(as.matrix(pmax_covariances))

# If we apply 2H only to the diagonal matrix with the variances
# of the variables, the vector that maximizes 2H is:
W <- diag(diag(C))
rownames(W)<-colnames(W)<-rownames(C)
pmax_variances <- twoHmax(W)$vector
dotchart(as.matrix(pmax_variances))

# If C contains the correlations among variables,
# a vector that maximizes 2H applied to C is estimated as follows:
C <- cor(V)
pmax_correlations <- twoHmax(C)$vector
dotchart(as.matrix(pmax_correlations))


# By attributing equal weights to the variables,
# 2H applied to the correlation matrix measures
# the number of effective variables:
# from 0 if all variables are completely correlated
# with each other to n if they are not correlated.
# Similarly, by attributing equal weights to the variables,
# 2H applied to the covariance matrix measures
# the effective amount of variation:
# from 0 if all variables are completely correlated
# with each other to n if they are not correlated
# and have similar variances.

#Even if the data set contains 15 variables,
# the effective number of variables is lower.
C <- cor(V)
equalproportions <- cbind.data.frame(rep(1/ncol(C), ncol(C)))
```

```
names(equalproportions) <- "equalprop"
equalproportions <- t(equalproportions)
qHdiv(equalproportions, C)

# When considering the covariances among species,
# instead of the correlations, the effective number
# of variables is even lower, indicating also an imbalance
# in the variances of the variables.
C <- cov(V)
qHdiv(equalproportions, C)

}

## End(Not run)
```

---

uniqueness	*Functional Uniqueness and Functional Redundancy of Biological Communities*

Description

The function `Uniqueness` calculates community-level functional uniqueness and redundancy.

Usage

```
uniqueness(comm, dis, tol = 1e-08, abundance = TRUE)
```

Arguments

comm	a matrix or a data frame containing the abundance or incidence (0/1) of species in communities (or plots). Columns are species and communities are rows.
dis	an object of class `dist` containing the functional distances among species. Values in dis must be bounded between 0 and 1. If they are not bounded, the function divides all values in dis by the highest observed value in dis.
tol	a tolerance threshold (a value between `-tol` and `tol` is considered as null).
abundance	a logical. If `TRUE`, abundance data are used when available; if `FALSE`, incidence (0/1) data are used.

Value

The function uniqueness returns a list of three data frames:

kbar: this first data frame gives values for Ricotta et al. (2016) coefficient $\bar{K}_i$'s per species (rows) and community (columns).

V: this second data frame gives values for Ricotta et al. (2016) coefficient V_i's per species (rows) and community (columns).

red: this third data set gives values, per community, for Ricotta et al. (2016) coefficients N (species richness), Q (quadratic diversity), D (Simpson diversity), U=Q/D (uniqueness), R=1-U (redundancy),

and Pavoine and Ricotta (2019) `Ustar=(1-D)/(1-Q)` (uniqueness) and `Rstar=1-Ustar` (redundancy); in this third data frame, coefficients are columns and communities are rows, the coefficients are thus calculated per community only.

Author(s)

Sandrine Pavoine `<sandrine.pavoine@mnhn.fr>`

References

Ricotta, C., de Bello, F., Moretti, M., Caccianiga, M., Cerabolini, B.E., Pavoine, S. (2016). Measuring the functional redundancy of biological communities: A quantitative guide. *Methods in Ecology and Evolution*, **7**, 1386–1395.

Pavoine, S., Ricotta, C. (2019). A simple translation from indices of species diversity to indices of phylogenetic diversity. *Ecological Indicators*, **101**, 552–561.

See Also

QE, treeUniqueness

Examples

```
data(RDMCCP16)
uniqueness(RDMCCP16$ab, as.dist(RDMCCP16$dis))
```

Index