Package ‘picante’

August 29, 2016

Type Package
Title R tools for integrating phylogenies and ecology
Version 1.6-2
Date 2014-03-05
Author Steven W. Kembel <steve.kembel@gmail.com>, David D. Ackery <dackerly@berkeley.edu>, Simon P. Blomberg <s.blomberg1@uq.edu.au>, Will K. Cornell <cornwell@zoology.ubc.ca>, Peter D. Cowan <pdc@berkeley.edu>, Matthew R. Helmus <mrhelmus@wisc.edu>, Helene Morlon <morlon.helene@gmail.com>, Campbell O. Webb <cwebb@oeb.harvard.edu>
Maintainer Steven W. Kembel <steve.kembel@gmail.com>
Depends ape, vegan, nlme
Suggests brglm, circular, corpcor, quantreg
Description Phylocom integration, community analyses, null-models, traits and evolution in R
License GPL-2
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-03-05 21:51:49

R topics documented:

picante-package .................................................. 2
color.plot.phylo .................................................. 3
comdist .......................................................... 4
comdistnt ......................................................... 5
comm.phylo.cor .................................................. 6
comm.phylo.qr .................................................... 8
cor.table ........................................................ 10
evol.distinct ..................................................... 10
expected.pd ..................................................... 11
IvesGodfray ....................................................... 13
Kcalc ............................................................. 14
match.phylo.data .................................................. 15
matrix2sample ..................................................... 16
mntd ................................................................. 17
mpd ................................................................. 18
multiPhylosignal .................................................... 19
pblm ................................................................. 19
pcd ................................................................. 22
pd ................................................................. 24
phyEstimate ......................................................... 25
phylocom ........................................................... 26
phylosignal .......................................................... 27
phylosor ............................................................ 28
phylosor.rnd ......................................................... 29
phylostruct .......................................................... 31
prune.sample ........................................................ 32
psd ................................................................. 33
randomizeMatrix ..................................................... 35
raoD ................................................................. 36
readsample ........................................................... 38
sample2matrix ....................................................... 38
ses.mntd ............................................................ 39
ses.mpd ............................................................ 41
ses.pd ............................................................. 42
specaccum.psr ....................................................... 44
species.dist ........................................................ 45
sppregs .............................................................. 46
tax.distinctiveness .................................................. 48
traitgram ............................................................ 49
unifrac .............................................................. 50
utility ............................................................... 51
writesample ......................................................... 52
writetraits .......................................................... 53

Index 54

picante-package picante: R tools for integrating phylogenies and ecology

Description

R tools for integrating phylogenies and ecology
Details

Package: picante
Type: Package
Version: 1.6-2
Date: 2014-03-05
License: GPL-2

Author(s)

Author: Steven W. Kembel <steve.kembel@gmail.com>, David D. Ackerly <dackerly@berkeley.edu>, Simon P. Blomberg <s.blomberg1@uq.edu.au>, Will K. Cornwell <cornwell@zoology.ubc.ca>, Peter D. Cowan <pdc@berkeley.edu>, Matthew R. Helmus <mrhelmus@wisc.edu>, Helene Morlon <morlon.helene@gmail.com>, Campbell O. Webb <cwebb@oeb.harvard.edu> Maintainer: Steven W. Kembel <steve.kembel@gmail.com>

color.plot.phylo

Color tip labels based on trait

Description

Plots a phylogeny with tip labels colored to indicate continuous or discrete trait values

Usage

```r
color.plot.phylo(phylo, df, trait, taxa.names, 
    num.breaks = ifelse(is.factor(df[,trait]), 
        length(levels(df[,trait])), 12), 
    col.names = rainbow(ifelse(length(num.breaks) > 1, 
        length(num.breaks) - 1, num.breaks)), 
    cut.labs = NULL, 
    leg.title = NULL, 
    main = trait, 
    leg.cex = 1, 
    tip.labs = NULL, 
    ...)```

Arguments

- `phylo`: An object of class `phylo`
- `df`: A dataframe containing the traits to be plotted
- `trait`: A string representing the name of column in the dataframe to be plotted
taxa.names  A string representing the name of column in the dataframe that contains the names of the taxa
num.breaks  For continuous traits, the number of bins to separate the data into
col.names   A vector of colors to use for tip labels
leg.title   A title for the tip color legend
main        A main title for the plot
cut.labs    A main title for the plot
leg.cex     A main title for the plot
tip.labs    A main title for the plot
...         Additional argument to pass to the plot.phylo function

Details

If if trait is a factor then each level of the factor is plotted with the corresponding col.names value (if length(num.breaks) > length(col.names) colors are recycled.) If trait is not a factor then it is assumed to be continuous and trait is evenly divided into num.breaks levels.

Value

The command is invoked for its side effect, a plot of the phylo with tips colored based on trait

Author(s)

Peter Cowan <pdc@berkeley.edu>

comdist  Calculates inter-community mean pairwise distance

Description

Calculates MPD (mean pairwise distance) separating taxa in two communities, a measure of phylogenetic beta diversity

Usage

comdist(comm, dis, abundance.weighted = FALSE)

Arguments

comm  Community data matrix
dis   Interspecific distance matrix
abundance.weighted  Should mean pairwise distances separating species in two communities be weighted by species abundances? (default = FALSE)
Details
This function calculates a measure of phylogenetic beta diversity: the expected phylogenetic distance separating two individuals or taxa drawn randomly from different communities.

Value
Distance object of MPD values separating each pair of communities.

Author(s)
Steven Kembel <steve.kembel@gmail.com>

References

See Also
mpd, ses.mpd

Examples
```r
data(phylocom)
comdist(phylocom$sample, cophenetic(phylocom$phylo), abundance.weighted=TRUE)
```

comdistnt
Calculates inter-community mean nearest taxon distance

Description
Calculates MNTD (mean nearest taxon distance) separating taxa in two communities, a measure of phylogenetic beta diversity

Usage
```r
comdistnt(comm, dis, abundance.weighted = FALSE, exclude.conspecifics = FALSE)
```

Arguments
```
comm                  Community data matrix
dis                   Interspecific distance matrix
abundance.weighted    Should mean nearest taxon distances from each species to species in the other community be weighted by species abundance? (default = FALSE)
exclude.conspecifics  Should conspecific taxa in different communities be exclude from MNTD calculations? (default = FALSE)
```

```
Details

This metric has also been referred to as MNND (mean nearest neighbour distance).

This function calculates a measure of phylogenetic beta diversity: the average phylogenetic distance to the most similar taxon or individual in the other community for taxa or individuals in two communities.

Value

Distance object of MNTD values separating each pair of communities.

Author(s)

Steven Kembel <steve.kembel@gmail.com>

References


See Also

`mndt, ses.mndt`

Examples

data(phylocom)
comdistnt(phylocom$sample, cophenetic(phylocom$phylo), abundance.weighted=FALSE)

---

Correlations between species co-occurrence and phylogenetic distances

Description

Calculates measures of community phylogenetic structure (correlation between co-occurrence and phylogenetic distance) to patterns expected under various null models.

Usage

```
comm.phylo.cor(samp, phylo, metric = c("cij", "checkerboard", "jaccard", "doij"),
null.model = c("sample.taxa.labels", "pool.taxa.labels",
"frequency", "richness", "independentswap","trialswap"), runs = 999, ...)
```
Arguments

- samp: Community data matrix
- phylo: Phylogenetic tree
- metric: Metric of co-occurrence to use (see `species.dist`)
- null.model: Null model to use (see Details section for description)
- runs: Number of runs (randomizations)
- ...: Additional arguments to `randomizeMatrix`

Details

Currently implemented null models (arguments to `null.model`):

- `sample.taxa.labels`: Shuffle phylogeny tip labels (only within set of taxa present in community data)
- `pool.taxa.labels`: Shuffle phylogeny tip labels (across all taxa included in phylogenetic tree)
- `frequency`: Randomize community data matrix abundances within species (maintains species occurrence frequency)
- `richness`: Randomize community data matrix abundances within samples (maintains sample species richness)
- `independentswap`: Randomize community data matrix maintaining species occurrence frequency and site richness using independent swap
- `trialsawp`: Randomize community data matrix maintaining species occurrence frequency and site richness using trial swap

Value

A list with elements:

- `obs.corr`: Observed co-occurrence/phylogenetic distance correlation
- `obs.corr.p`: P-value of observed correlation (standard P-value for correlation coefficient, not based on comparison with randomizations)
- `obs.rank`: Rank of observed correlation vs. random
- `runs`: Number of runs (randomizations)
- `obs.rand.p`: P-value of observed correlation vs. randomizations (= obs.rank / (runs + 1))
- `random.corr`: A vector of random correlation calculated for each run

Author(s)

Steven Kembel <steve.kembel@gmail.com>

References

comm.phylo.qr

See Also
randomizeMatrix

Examples

data(phylocom)
comm.phylo.cor(phylocom$sample, phylocom$phylo, metric="cij", null.model="sample.taxa.labels")

comm.phylo.qr Quantile regression slopes between species co-occurrence and phylogenetic distances

Description
Calculates measures of community phylogenetic structure (quantile regression between co-occurrence and phylogenetic distance) to patterns expected under various null models.

Usage

comm.phylo.qr(samp, phylo, metric = c("cij", "checkerboard", "jaccard", "doij"),
null.model = c("sample.taxa.labels", "pool.taxa.labels",
"frequency", "richness", "independentswap", "trialswap"),
quant = 0.75, runs = 999, show.plot = FALSE, ...)

Arguments

samp Community data matrix
phylo Phylogenetic tree
metric Metric of co-occurrence to use (see species.dist)
null.model Null model to use (see Details section for description)
quant Quantile of slope to be fit (using rq)
runs Number of runs (randomizations)
show.plot Option to display a plot of co-occurrence versus phylogenetic distance with quantile regression slope fit
...
Additional arguments to randomizeMatrix

Details
This function fits a quantile regression of co-occurrence versus phylogenetic distances separating species, and compares observed patterns to the patterns expected under some null model. The quantile regressions are fit using the rq function from the quantreg package.

Currently implemented null models (arguments to null.model):

sample.taxa.labels Shuffle phylogeny tip labels (only within set of taxa present in community data)
**comm.phylo.qr**

**pool.taxa.labels**  Shuffle phylogeny tip labels (across all taxa included in phylogenetic tree)

**frequency**  Randomize community data matrix abundances within species (maintains species occurrence frequency)

**richness**  Randomize community data matrix abundances within samples (maintains sample species richness)

**independentswap**  Randomize community data matrix maintaining species occurrence frequency and site richness using independent swap

**trialswap**  Randomize community data matrix maintaining species occurrence frequency and site richness using trial swap

**Value**

A list with elements:

- `obs.qr.intercept`  Observed co-occurrence/phylogenetic distance quantile regression intercept
- `obs.qr.slope`  Observed co-occurrence/phylogenetic distance quantile regression slope
- `obs.qr.slope.p`  P-value of observed quantile regression slope significance versus null model (calculated based on comparison with randomizations)
- `obs.rank`  Rank of observed quantile regression slope vs. random
- `runs`  Number of runs (randomizations)
- `random.qr.slopes`  A vector of quantile regression slopes calculated for each randomization

**Author(s)**

Steven Kembel <steve.kembel@gmail.com>

**References**


**See Also**

`randomizematrix`

**Examples**

```r
data(phylocom)
comm.phylo.qr(phylocom$sample, phylocom$phylo, metric="cij",
  null.model="sample.taxa.labels", runs=99)
```
cor.table Table of correlations and P-values

Description
Table of correlations with associated P-values and df, can be used with regular or independent contrast data

Usage
cor.table(x, cor.method = c("pearson","spearman"), cor.type=c("standard","contrast"))

Arguments
x Data frame of data points or contrasts at nodes
cor.method Correlation method (as cor)
cor.type Are data standard or independent contrast values?

Value
r Correlation values
df Degrees of freedom
p P-values

Author(s)
Steven Kembel <steve.kembel@gmail.com>

References

evol.distinct Species’ evolutionary distinctiveness

Description
Calculates evolutionary distinctiveness measures for a suite of species by: a) equal splits (Redding and Mooers 2006) b) fair proportions (Isaac et al., 2007). Returns a datafram with species identifiers and species scores.
Usage

```
evol.distinct(tree, type = c("equal.splits", "fair.proportion"),
    scale = FALSE, use.branch.lengths = TRUE)
```

Arguments

- **tree**: an object of class phylo
- **type**: a) equal splits (Redding and Mooers 2006) or b) fair proportions (Isaac et al., 2007)
- **scale**: The scale option refers to whether or not the phylogeny should be scaled to a depth of 1 or, in the case of an ultrametric tree, scaled such that branch lengths are relative.
- **use.branch.lengths**: If use.branch.lengths=FALSE, then all branch lengths are changed to 1.

Note

This function will return a vector of evolutionary distinctiveness for every species in the given tree. If only a subset of values are needed there are two, conceptually distinct options: either prune the tree first and then pass the tree in or subset the resulting vector. These two options will provide very different outputs.

Author(s)

Karen Magnuson-Ford, Will Cornwell, Arne Mooers, Mark Vellend

References


Description

Calculates the expected phylogenetic diversity (Faith’s PD) and variance of PD under binomial sampling with a fixed probability of each tip being sampled, and the Edge-length Abundance Distribution of a phylogeny.
Usage

expected.pd(phy)
variance.pd(phy, upper.bound=TRUE)
ead(phy)

Arguments

phy phyl object
upper.bound Calculate upper bound of PD variance? (default = TRUE)

Details

The function `expected.pd` calculates the expected phylogenetic diversity (Faith’s PD - total branch length) for all subsets of a phylogeny, based on an analytic solution for expected PD.

The function `variance.pd` additionally calculates the variance of expected PD for all subsets of a phylogeny, based on an analytic solution for expected PD. If argument `upper.bound=TRUE`, a fast solution for the upper bound of the variance is returned. Otherwise, the exact solution for the variance is returned. Note that the exact solution is much slower than the upper bound solution.

The function `ead` calculates the edge abundance distribution (EAD), the length of edges with different numbers of descendant tips.

Value

n Expected Number of tips sampled
expected.pd Expected PD for a given n
variance.pd Variance of PD for a given n
num.children Number of tips descended from an edge
dge.length Total phylogenetic edge length for a given number of tips descended from an edge

Author(s)

Steven Kembel <steve.kembel@gmail.com> and James O’Dwyer <jodwyer@santafe.edu>

References


See Also

pd
Examples

```r
randtree <- rcoal(300)
randtree.pd.ub <- variance.pd(randtree, upper.bound=TRUE)
randtree.pd.exact <- variance.pd(randtree, upper.bound=FALSE)
plot(expected.pd(randtree), xlab="Number of tips",
     ylab="Phylogenetic diversity (PD)", type="l", log="xy")
lines(randtree.pd.exact$expected.pd+1.96*sqrt(randtree.pd.exact$variance.pd), lty=2)
lines(randtree.pd.ub$expected.pd+1.96*sqrt(randtree.pd.ub$variance.pd), lty=2)
lines(randtree.pd.ub$expected.pd+1.96*sqrt(randtree.pd.ub$variance.pd), lty=3)
lines(randtree.pd.ub$expected.pd-1.96*sqrt(randtree.pd.ub$variance.pd), lty=3)
legend("bottomright", lty=c(1,2,3), legend=c("Expected PD",
                                  "95 percent CI (exact)", "95 percent CI (upper bound)"))
```

---

**IvesGodfray**

*Host-parasitoid food web data*

**Description**

Data on the structure of a host-parasitoid food web from Ives & Godfray (2006). Includes information on phylogenetic covariances among 12 leaf-mining moth hosts and 27 species of parasitoid wasps.

**Usage**

```r
data(IvesGodfray)
```

**Format**

A list with three elements:

- host: Phylogenetic variance/covariance matrix for 12 leaf-mining moth hosts
- parasitoid: Phylogenetic variance/covariance matrix for 27 species of parasitoid wasps
- interactions: Matrix describing interactions between hosts and parasitoids

**Source**


**See Also**

pblm
**Kcalc**

*K statistic of phylogenetic signal*

Description

Calculates K statistic of phylogenetic signal

Usage

```r
Kcalc(x, phy, checkdata=TRUE)
```

Arguments

- `x`: Vector or data.frame of trait data (in phylo$tip.label order)
- `phy`: phylo object
- `checkdata`: Check for match between trait and phylogeny taxa labels using `match.phylo.data`? (default=TRUE)

Value

- `K`: K statistic

Author(s)

Simon Blomberg <s.blomberg1@uq.edu.au> and David Ackerly <dackerly@berkeley.edu>

References


See Also

`phylosignal`

Examples

```r
## Not run:
randtree <- rcoal(20)
randtraits <- rTraitCont(randtree)
Kcalc(randtraits[randtree$tip.label], randtree)

## End(Not run)
```
match.phylo.data  

**Match taxa in phylogeny and data**

**Description**

These functions compare taxa present in phylogenies with community or trait data sets, pruning and sorting the two kinds of data to match one another for subsequent analysis.

**Usage**

```
match.phylo.comm(phy, comm)
match.phylo.data(phy, data)
match.comm.dist(comm, dis)
```

**Arguments**

- `phy` A phylogeny object of class phylo
- `comm` Community data matrix
- `data` A data object - a vector (with names matching phy) or a data.frame or matrix (with row names matching phy)
- `dis` A distance matrix - a dist or matrix object

**Details**

A common pitfall in comparative analyses in R is that taxa labels are assumed to match between phylogenetic and other data sets. These functions prune a phylogeny and community or trait data set to match one another, reporting taxa that are missing from one data set or the other.

Taxa names for phylogeny objects are taken from the phylogeny’s tip labels. Taxa names for community data are taken from the column names. Taxa names for trait data are taken from the element names (vector) or row names (data.frame or matrix). Taxa names for distance data are taken from column/row names of the distance matrix/dist object.

If community data lack taxa names, the function will issue a warning and no result will be returned, since the community-phylogenetic analyses in `picante` require named taxa in the community data set.

If trait data or distance matrix lack names, a warning is issued and the data are assumed to be sorted in the same order as the phylogeny’s tip labels or community’s column labels.

These utility functions are used by several functions that assume taxa labels in phylogeny and data match, including `Kcalc`, `phylosignal`, and `raoD`.

**Value**

A list containing the following elements, pruned and sorted to match one another:

- `phy` A phylogeny object of class phylo
- `comm` Community data matrix
data A data object (vector, data.frame or matrix)
dist A distance matrix - a dist or matrix object

Author(s)
Steven Kembel <steve.kembel@gmail.com>

See Also
prune.missing, prune.sample

Examples
data(phylocom)
matrix2sample(phylocom$sample)
matrix2sample(phylocom$traits[1:10,])

matrix2sample  Convert community data matrix to Phylocom sample

Description
Converts a community data matrix to a Phylocom database-format community sample

Usage
matrix2sample(z)

Arguments
z Community data matrix

Value
Phylocom database-format community sample

Author(s)
Steven Kembel <steve.kembel@gmail.com> and Cam Webb <cwebb@oeb.harvard.edu>

References

Examples
data(phylocom)
matrix2sample(phylocom$sample)
### Description

Calculates MNTD (mean nearest taxon distance) for taxa in a community

### Usage

```r
mnd(samp, dis, abundance.weighted=FALSE)
```

### Arguments

- **samp**: Community data matrix
- **dis**: Interspecific distance matrix
- **abundance.weighted**: Should mean nearest taxon distances for each species be weighted by species abundance? (default = FALSE)

### Details

This metric has also been referred to as MNND (mean nearest neighbour distance), and the function was named `mnnd` in picante versions < 0.7.

### Value

Vector of MNTD values for each community.

### Author(s)

Steven Kembel <steve.kembel@gmail.com>

### References


### See Also

- `ses.mnd`

### Examples

```r
data(phylocom)
mnd(phylocom$sample, cophenetic(phylocom$phylo), abundance.weighted=TRUE)
```
**mpd**

*Mean pairwise distance*

**Description**

Calculates mean pairwise distance separating taxa in a community

**Usage**

```r
mpd(samp, dis, abundance.weighted=FALSE)
```

**Arguments**

- `samp` Community data matrix
- `dis` Interspecific distance matrix
- `abundance.weighted` Should mean pairwise distances be weighted by species abundance? (default = FALSE)

**Value**

Vector of MPD values for each community

**Author(s)**

Steven Kembel <steve.kembel@gmail.com>

**References**


**See Also**

`ses.mpd`

**Examples**

```r
data(phylocom)
mpd(phylocom$sample, cophenetic(phylocom$phylo), abundance.weighted=TRUE)
```
multiPhylosignal

Description

Calculates phylogenetic signal for data.frame of traits. Traits may have missing values in which case the tree will be pruned prior to calculating phylogenetic signal for each trait.

Usage

multiPhylosignal(x, phy, checkdata=TRUE, ...)

Arguments

x Data frame of trait data (traits in columns) with row names corresponding to tip.labels
phy phylo object
checkdata Check for match between trait and phylogeny taxa labels using match.phylo.data? (default=TRUE)
... Additional arguments to phylosignal

Value

Returns a data frame with phylogenetic signal results for each trait

Author(s)

Steven Kembel <steve.kembel@gmail.com>

pblm

Phylogenetic Bipartite Linear Model

Description

Fits a linear model to the association strengths of a bipartite data set with or without phylogenetic correlation among the interacting species

Usage

pblm(assocs, tree1=NULL, tree2=NULL, covars1=NULL, covars2=NULL, bootstrap=FALSE, nreps=10, maxit=10000, pstart=c(.5,.5))
pblmpredict(x, tree1.w.novel=NULL, tree2.w.novel=NULL, predict.originals=FALSE)
Arguments

- **assoc**: A matrix of association strengths among two sets of interacting species.
- **tree1**: A phylo tree object or a phylogenetic covariance matrix for the rows of `assoc`.
- **tree2**: A phylo tree object or a phylogenetic covariance matrix for the columns of `assoc`.
- **covars1**: A matrix of covariates (e.g., traits) for the row species of `assoc`.
- **covars2**: A matrix of covariates (e.g., traits) for the column species of `assoc`.
- **bootstrap**: logical, bootstrap confidence intervals of the parameter estimates.
- **nreps**: Number of bootstrap replicated data sets to estimate parameter CIs.
- **maxit**: as in `optim`.
- **pstart**: starting values of the two phylogenetic signal strength parameters passed to `optim`.
- **x**: object of class `pblm`.
- **tree1.w.novel**: A phylo tree object or a phylogenetic covariance matrix which corresponds to `tree1` of `x` with species to predict associations.
- **tree2.w.novel**: A phylo tree object or a phylogenetic covariance matrix which corresponds to `tree2` of `x` with species to predict associations.
- **predict.originals**: if TRUE then the associations of each original species in the two phylogenies is predicted.

Details

Fit a linear model with covariates using estimated generalized least squares to the association strengths between two sets of interacting species. Associations can be either binary or continuous. If phylogenies of the two sets of interacting species are supplied, two phylogenetic signal strength parameters ($d_1$ and $d_2$), one for each species set, based on an Ornstein-Uhlenbeck model of evolution with stabilizing selection are estimated. Values of $d=1$ indicate no stabilizing selection and correspond to the Brownian motion model of evolution; $0<d<1$ represents stabilizing selection; $d=0$ depicts the absence of phylogenetic correlation (i.e., a star phylogeny); and $d>1$ corresponds to disruptive selection where phylogenetic signal is amplified. Confidence intervals for these and the other parameters can be estimated with bootstrapping.

The function `pblm.predict` predicts the associations of novel species following the methods given in appendix B of Ives and Godfray (2006).

Value

- **MSE**: total, full (each $d$ estimated), star ($d=0$), and base ($d=1$) mean squared errors.
- **signal.strength**: two estimates of phylogenetic signal strength.
- **coefficients**: estimated intercept and covariate coefficients with approximate 95 percent CIs for the three model types (full, star, base).
- **CI.boot**: 95 percent CIs for all parameters.
`pblm`

- **variates**: matrix of model variates (can be used for plotting)
- **residuals**: matrix of residuals from the three models (full, star and base)
- **predicted**: predicted associations
- **bootvalues**: matrix of parameters estimated from the `nreps` bootstrap replicated data sets used to calculate CIs
- **phylocovs**: phylogenetic covariance matrices scaled by the estimated d1 and d2
- **cors.1**: correlations among predicted and observed associations for species of `tree1`, NA if `predict.originals=FALSE`
- **cors.2**: correlations among predicted and observed associations for species of `tree2`, NA if `predict.originals=FALSE`
- **pred.novels1**: predicted associations for the novel species of `tree1`
- **pred.novels2**: predicted associations for the novel species of `tree2`

**Note**

Covariates that apply to both species sets (e.g., sampling site) should be supplied in the covariate matrix of the set with the most species.

Bootstrapping CIs is slow due to the function `optim` used to estimate the model parameters. See appendix A in Ives and Godfray (2006) for a discussion about this bootstrapping procedure.

If `pblm.predict=TRUE` the function does not first remove each species in turn when predicting the associations of the original species as is done in Ives and Godfray (2006).

**Author(s)**

Matthew Helmus <mrhelmus@gmail.com>

**References**


**Examples**

```r
## Not run:
# load example data from Ives & Godfray (2006)
data(IvesGodfray)

# net attack rate of parasitoid on host eq.4 in Ives and Godfray
A<-(-1*xlog(1-IvesGodfray$interactions[,28])/t(IvesGodfray$interactions[28])))

# Make tips of the phylogenetic trees contemporaneous by extending tips
p<-dim(IvesGodfray$host)[1]
q<-dim(IvesGodfray$parasitoid)[1]
```
Phylogenetic Community Dissimilarity

**Description**

Pairwise dissimilarity in phylogenetic community composition that is partitioned into a nonphylogenetic and a phylogenetic component.

**Usage**

```r
pcd(comm, tree, PSVmncd=NULL, PSVpool=NULL, reps=10^4)
```

**Arguments**

- **comm**: Community data matrix
- **tree**: Object of class phylo or a phylogenetic covariance matrix
- **PSVmncd**: Vector of null mean conditional phylogenetic species variability (PSV) values
- **PSVpool**: The standard, unconditional PSV calculated for the species pool
- **reps**: The number of random draws from the species pool used to produce PSVmncd

**Details**

*Phylogenetic community dissimilarity (PCD)* is the pairwise differences between communities derived by asking how much of the variance among species in the values of a hypothetical nonselected trait in one community can be predicted by the known trait values of species in another community.
PCD is partitioned into a nonphylogenetic component that reflects shared species between communities (PCDc) and a phylogenetic component that reflects the evolutionary relationships among nonshared species (PCDp). In order to compare communities that vary in species richness, the metric is standardized under the assumption that the species in communities are selected at random from the species pool. The analyses here define the species pool as the list of all species in the set of communities in comm, but the species pool can be defined under any hypothesis of community assembly either by manipulating the code or inputting a user defined PSV mncd and PSV pool.

**Value**

The function returns a list with items:

- PCD: A square matrix of PCD values
- PCDc: A square matrix of PCDc values
- PCDp: A square matrix of PCDp values
- PSV mncd: A vector of null mean conditional PSV values used to calculate PCD
- PSV pool: The unconditional PSV of the species pool used to calculate PCD

**Note**

The sampling procedure used to standardize PCD and produce PSV mncd and PSV pool can be slow.

**Author(s)**

Anthony Ives <arives@wisc.edu> and Matthew Helmus <mrhelmus@gmail.com>

**References**


**See Also**

psv, phylosor, unifrac

**Examples**

```r
## Not run:
data(phylocom)
pcd(phylocom$sample, phylocom$phylo)
## End(Not run)
```
Calculate Faith’s Phylogenetic Diversity

Description

Calculate the sum of the total phylogenetic branch length for one or multiple samples.

Usage

\[
\text{pd}(\text{samp}, \text{tree}, \text{include.root}=\text{TRUE})
\]

Arguments

\begin{itemize}
\item \text{samp} Community data matrix
\item \text{tree} A phylo tree object
\item \text{include.root} Should the root node be included in all PD calculations (default = TRUE)
\end{itemize}

Value

Returns a dataframe of the PD and species richness (SR) values for all samples

Warning

If the root is to be included in all calculations (\text{include.root}=\text{TRUE}), the PD of all samples will include the branch length connecting taxa in those samples and the root node of the supplied tree. The root of the supplied tree may not be spanned by any taxa in the sample. If you want the root of your tree to correspond to the most recent ancestor of the taxa actually present in your sample, you should prune the tree before running \text{pd}:

\[
\text{prunedTree} \leftarrow \text{prune.sample}(\text{sample}, \text{tree})
\]

Note

The data sets need not be species-community data sets but may be any sample data set with an associated phylogeny. PD is not statistically independent of species richness, it positively correlates with species richness across samples. The function \text{ses.pd} compares observed PD to the values expected under various randomizations and allows a way to standardize for unequal richness across samples.

If the root is to be included in all calculations of PD (\text{include.root}=\text{TRUE}), the tree must be rooted. Single-species samples will be assigned a PD value equal to the distance from the root to the present. If the root is not included in all calculations by default (\text{include.root}=\text{FALSE}), the tree need not rooted, but in the case of single-species samples the PD will be equal to NA and a warning will be issued.

Author(s)

Matthew Helmus <mrhelmus@gmail.com>, Jonathan Davies <davies@nceas.ucsb.edu>, Steven Kembel <steve.kembel@gmail.com>
References

See Also
psr, ses.pd

Examples
data(phylocom)
pd(phylocom$sample, phylocom$phylo)

phyEstimate  Phylogenetic estimation of traits for unobserved taxa

Description
Uses phylogenetic ancestral state reconstruction to estimate trait values for unobserved taxa.

Usage
phyEstimate(phy, trait, method="pic", ...)
phyEstimateDisc(phy, trait, best.state=TRUE, cutoff=0.5, ...)

Arguments

phy  phylo object
trait  vector or data.frame containing trait values
method  ancestral state estimation method used by ace (default="pic")
best.state  estimate best-supported trait state for discrete variables? (default=TRUE)
cutoff  support cutoff required to declare a best.state
...  Additional arguments passed to ace

Details
These functions use phylogenetic ancestral state estimation to infer trait values for novel taxa on a phylogenetic tree, for continuous (phyEstimate) and discrete (phyEstimateDisc) traits. The required input is a phylogenetic tree object plus a vector or data.frame containing estimated trait values for a subset of the taxa in the phylogenetic tree. Trait values for taxa that are present in the tree but not the trait data will be estimated using ancestral state estimation (Garland and Ives 2000). Briefly, for each taxon present in the tree but not the trait data, the phylogeny is rerooted at the most recent common ancestor of the novel taxon and the rest of the phylogeny, and the trait value of the novel taxon is estimated from the reconstructed trait value at the root of the rerooted phylogeny.

For phyEstimateDisc, the state with the highest support will be reported if argument best.state=TRUE. If the best-supported state’s support is less than the specified cutoff, no best state is reported and a NA value will be returned.
Value

phyEstimate produces a data frame with columns:

est          Estimated trait value
se           Standard error of estimated trait value

phyEstimateDisc produces a data frame with columns:

states  1..N  A column with statistical support is produced for each discrete trait state
estimated.state
  If best.state=TRUE, a column with the state with the highest support
estimated.state.support
  Statistical support for the state with the highest support

Author(s)

Steven Kembel <steve.kembel@gmail.com>

References


Examples

#generate random phylogeny
randtree <- rcoal(50)
#simulate trait evolution for a subset of taxa on phylogeny
randtraits <- sample(rTraitCont(randtree, sigma=10, root.value=100), 40)
#estimate trait values for "missing" taxa using PIC method
phyEstimate(randtree, randtraits, method="pic")

phylocom          Phylocom default data

Description

Phylogeny, community and trait data from the Phylocom 4.0 distribution

Usage

data(phylocom)
phylosignal

**Format**

A list with three elements:

- **phylo** Phylogenetic tree (an object of class phylo)
- **sample** Community data (a data.frame with samples in rows and species in columns)
- **traits** Trait data (a data.frame with species in rows and traits in columns)

**Source**


---

**phylosignal**  
*Measure phylogenetic signal*

**Description**

Calculates K statistic of phylogenetic signal as well as P-value based on variance of phylogenetically independent contrasts relative to tip shuffling randomization.

**Usage**

`phylosignal(x, phy, reps = 999, checkdata = TRUE, ...)`

**Arguments**

- `x` Trait vector (same order as phy\$tip.label)
- `phy` phylo object
- `reps` Number of randomizations
- `checkdata` Check for match between trait and phylogeny taxa labels using `match.phylo.data`? (default=TRUE)
- `...` Additional arguments passed to `pic`

**Value**

Data frame with columns:

- **K** K statistic
- **PIC.variance** Mean observed PIC variance
- **PIC.variance.P** P-value of observed vs. random variance of PICs
- **PIC.variance.z** Z-score of observed vs. random variance of PICs
Author(s)

Steven Kembel <steve.kembel@gmail.com>

References


See Also

Kcalc

Examples

## Not run:
randtree <- rcoal(20)
randtraits <- rTraitCont(randtree)
phylosignal(randtraits[randtree$tip.label],randtree)

## End(Not run)

phylosor

Phylogenetic index of beta-diversity PhyloSor

Description

Fraction of branch-length shared between two communities

Usage

phylosor(samp, tree)

Arguments

samp Community data matrix
tree Object of class phylo - a rooted phylogeny

Value

A distance object of the PhyloSor index of similarity between communities, the fraction of PD (branch-length) shared between two samples
Warning

The phylosor of all samples will include the branch length connecting taxa in those samples and the root of the supplied tree. The root of the supplied tree may not be spanned by any taxa in the sample. If you want the root of your tree to correspond to the most recent ancestor of the taxa actually present in your sample, you should prune the tree before running phylosor:

```r
prunedTree <- prune.sample(sample, tree)
```

Note

The root of the supplied tree is included in calculations of PhyloSor. The supplied tree must be rooted. Single-species samples will be assigned a PD value equal to the distance from the root to the present.

Author(s)

Helene Morlon <morlon.helene@gmail.com> and Steven Kembel <steve.kembel@gmail.com>

References


See Also

phylosor.rnd, pd

Examples

```r
data(phylocom)
phylosor(phylocom$sample, phylocom$phylo)
```

phylosor.rnd

Null PhyloSor values of phylogenetic beta-diversity

Description

PhyloSor values obtained by randomization for different choices of null models

Usage

```r
phylosor.rnd(samp, tree, cstSor=TRUE, null.model=c("taxa.labels", "frequency","richness","independentswap","trialswap"), runs=999, iterations=1000)
```
Arguments

samp  Community data matrix
tree  Object of class phylo - a rooted phylogeny
cstSor  TRUE if the Sorensen similarity should be kept constant across communities. FALSE otherwise
null.model  Null model to use (see Details section)
runs  Number of randomizations
iterations  Number of iterations to use for each randomization (for independent swap and trial null models)

Details

Currently implemented null models (arguments to null.model):

  taxa.labels  Shuffle community data matrix labels. Maintains species richness in each community and species shared between communities. Should be used with cstSor=TRUE

  frequency  Randomize community data matrix abundances within species (maintains species occurrence frequency). Does not maintain species richness in communities nor species shared between communities. Can only be used with cstSor=FALSE

  richness  With cstSor=TRUE: For each pair of community, maintains species richness in each community and species shared between communities. Sample in the species pool with equal probability; With cstSor=FALSE: Maintains species richness in each community, does not maintain species shared between communities. Sample in the species pool with equal probability

  independentswap  Randomize community data matrix with the independent swap algorithm (Gotelli 2000) maintaining species occurrence frequency and sample species richness. Can only be used with cstSor=FALSE

  trialswap  Randomize community data matrix with the trial-swap algorithm (Miklos & Podani 2004) maintaining species occurrence frequency and sample species richness. Can only be used with cstSor=FALSE

Value

A list of length the number of runs. Each element of the list is a distance matrix containing the PhyloSor values of phylogenetic beta-diversity obtained by randomization

Author(s)

Helene Morlon <morlon.helene@gmail.com> and Steven Kembel <steve.kembel@gmail.com>

References

phylostruct

See Also

phylosor, randomizematrix

Examples

```r
data(phylocom)
phylosor.rnd(phylocom$sample, phylocom$phylo, cstSor=TRUE, null.model="richness", runs=5)
```

---

**phylostruct**

**Permutations to Test for Phylogenetic Signal in Community Composition**

**Description**

Randomize sample/community data matrices to create null distributions of given metrics

**Usage**

```r
phylostruct(samp, tree, env=NULL, metric=c("psv", "psr", "pse", "psc", "sppregs"),
null.model=c("frequency", "richness", "independentswap", "trialswap"),
runs=100, it=1000, alpha=0.05, fam="binomial")
```

**Arguments**

- **samp**: community data matrix, species as columns, communities as rows
- **tree**: phylo tree object or a phylogenetic covariance matrix
- **env**: environmental data matrix
- **metric**: if metric="psv", "psr", "pse", or "psc" compares the observed mean of the respective metric to a null distribution at a given alpha; if metric="sppregs" compares the three correlations produced by sppregs to null distributions
- **null.model**: permutation procedure used to create the null distribution, see randomizematrix
- **runs**: the number of permutations to create the distribution, a rule of thumb is (number of communities)/alpha
- **it**: the number of swaps for the independent and trial-swap null models, see randomizematrix
- **alpha**: probability value to compare the observed mean/correlations to a null distribution
- **fam**: as in sppregs

**Details**

The function creates null distributions for the psd set of metrics and for the correlations of sppregs from observed community data sets.
**Value**

- **metric**: metric used
- **null.model**: permutation used
- **runs**: number of permutations
- **it**: number of swaps if applicable
- **obs**: observed mean value of a particular metric or the three observed correlations from `sppregs`
- **mean.null**: mean(s) of the null distribution(s)
- **quantiles.null**: quantiles of the null distribution(s) to compare to obs; determined by alpha
- **phylo.structure**: if obs less than (alpha/2), phylo.structure="underdispersed"; if obs greater than (1-alpha/2), phylo.structure="overdispersed"; otherwise phylo.structure="random" and NULL if metric="sppregs"
- **nulls**: null values of the distribution(s)

**Author(s)**

Matthew Helmus <mrhelmus@gmail.com>

**References**


**See Also**

`psd`, `sppregs`, `randomizeMatrix`

---

**prune.sample**

*Prune tree to match community data or trait data*

**Description**

Prune a phylogenetic tree to include only species present in a community data set or with non-missing trait data

**Usage**

```r
prune.sample(samp, phylo)
prune.missing(x, phylo)
```
**psd**  

**Phylogenetic Species Diversity Metrics**

**Description**

Calculate the bounded phylogenetic biodiversity metrics: phylogenetic species variability, richness, evenness and clustering for one or multiple samples.

**Usage**

```r
psv(samp, tree, compute.var=TRUE)
psr(samp, tree, compute.var=TRUE)
pse(samp, tree)
psc(samp, tree)
psd(samp, tree, compute.var=TRUE)
psv.spp(samp, tree)
```

**Arguments**

- `samp`: Community data matrix
- `tree`: A phylo tree object or a phylogenetic covariance matrix
- `compute.var`: Computes the expected variances for PSV and PSR for each community

**Details**

*Phylogenetic species variability (PSV)* quantifies how phylogenetic relatedness decreases the variance of a hypothetical unselected/neutral trait shared by all species in a community. The expected value of PSV is statistically independent of species richness, is one when all species in a sample are unrelated (i.e., a star phylogeny) and approaches zero as species become more related. PSV is directly related to mean phylogenetic distance. The expected variance around PSV for any sample of a particular species richness can be approximated. To address how individual species contribute to the mean PSV of a data set, the function `psv.spp` gives signed proportions of the total deviation from the mean PSV that occurs when all species are removed from the data set one at a time. The
absolute values of these “species effects” tend to positively correlate with species prevalence.

*Phylogenetic species richness (PSR)* is the number of species in a sample multiplied by PSV. It can be considered the species richness of a sample after discounting by species relatedness. The value is maximum at the species richness of the sample, and decreases towards zero as relatedness increases. The expected variance around PSR for any sample of a particular species richness can be approximated.

*Phylogenetic species evenness (PSE)* is the metric PSV modified to incorporate relative species abundances. The maximum attainable value of PSE (i.e., 1) occurs only if species abundances are equal and species phylogeny is a star. PSE essentially grafts each individual of a species onto the tip of the phylogeny of its species with branch lengths of zero.

*Phylogenetic species clustering (PSC)* is a metric of the branch tip clustering of species across a sample’s phylogeny. As PSC increases to 1, species are less related to one another the tips of the phylogeny. PSC is directly related to mean nearest neighbor distance.

**Value**

Returns a dataframe of the respective phylogenetic species diversity metric values

**Note**

These metrics are bounded either between zero and one (PSV, PSE, PSC) or zero and species richness (PSR); but the metrics asymptotically approach zero as relatedness increases. Zero can be assigned to communities with less than two species, but conclusions drawn from assigning communities zero values need be carefully explored for any data set. The data sets need not be species-community data sets but may be any sample data set with an associated phylogeny.

**Author(s)**

Matthew Helmus <mrhelmus@gmail.com>

**References**


**See Also**

`mpd`, `mnnd`, `specaccum.psr`
randomizeMatrix

Null models for community data matrix randomization

Description

Various null models for randomizing community data matrices

Usage

randomizeMatrix(samp, null.model = c("frequency", "richness", "independentswap", "trialswap"), iterations = 1000)

Arguments

samp Community data matrix
null.model Null model to use (see Details section for description)
iterations Number of independent or trial-swaps to perform

Details

Currently implemented null models (arguments to null.model):

frequency Randomize community data matrix abundances within species (maintains species occurrence frequency)
richness Randomize community data matrix abundances within samples (maintains sample species richness)
independentswap Randomize community data matrix with the independent swap algorithm (Gotelli 2000) maintaining species occurrence frequency and sample species richness
trialswap Randomize community data matrix with the trial-swap algorithm (Miklos & Podani 2004) maintaining species occurrence frequency and sample species richness

Value

Randomized community data matrix

Author(s)

Steven Kembel <steve.kembel@gmail.com>

References

raoD

Rao’s quadratic entropy

Examples

data(phylocom)
randomizematrix(phylocom$sample, null.model="richness")

Description

Calculates Rao’s quadratic entropy, a measure of within- and among-community diversity taking species dissimilarities into account.

Usage

raoD(comm, phy=NULL)

Arguments

- **comm**: Community data matrix
- **phy**: Object of class phylo - an ultrametric phylogenetic tree (optional)

Details

Rao’s quadratic entropy (Rao 1982) is a measure of diversity in ecological communities that can optionally take species differences (e.g. phylogenetic dissimilarity) into account. This method is conceptually similar to analyses of genetic diversity among populations (Nei 1973), but instead of diversity of alleles among populations, it measures diversity of species among communities.

If no phylogeny is supplied, $D_{kk}$ is equivalent to Simpson’s diversity (probability that two individuals drawn from a community are from different taxa), $D_{kl}$ is a beta-diversity equivalent of Simpson’s diversity (probability that individuals drawn from each of two communities belong to different taxa), and $H$ is $D_{kl}$ standardized to account for within-community diversity.

If an ultrametric phylogeny is supplied, $D_{kk}$ is equivalent to the mean pairwise phylogenetic distance (distance to MRCA) between two individuals drawn from a community, $D_{kl}$ is the mean pairwise phylogenetic distance between individuals drawn from each of two communities, and $H$ is $D_{kl}$ standardized to account for within-community diversity.

\[
D_{kl} = \sum(t_{ij} \cdot x_{ki} \cdot x_{lj})
\]

where $x_{ki}$ is the relative abundance of taxon $i$ in community $k$ and $t_{ij}$ is a matrix of weights for all pairs of taxa $i,j$. Without a phylogeny, when $i=j$, $t_{ij}=0$, otherwise $t_{ij}=1$. With a phylogeny, $t_{ij}$ is the phylogenetic distance to MRCA for taxa $i,j$.

\[
H_{kl} = D_{kl} - (D_{kk} + D_{ll})/2
\]

Alpha, beta and total measure the average diversity within, among, and across all communities based on $D_{kk}$ and $H$ values taking variation in number of individuals per community into account. A Fst-like measure is calculated by dividing beta by the total diversity across all samples.
Value

A list of results

- **DKk** Within-community diversity
- **DKl** Among-community diversity
- **H** Among-community diversities excluding within-community diversity
- **Total** Total diversity across all samples
- **Alpha** Alpha diversity - average within-community diversity
- **Beta** Beta diversity - average among-community diversity
- **Fst** Beta diversity / total diversity

Warning

Alpha, beta, and total diversity components and Fst should not be interpreted as a measure of relative differentiation among versus within communities. See Jost (2007) for a detailed description of this problem. Hardy and Jost (2008) suggest Fst can be interpreted as 'local species identity excess' or 'local phylogenetic similarity excess' rather than as a measure of among-community differentiation.

Author(s)

Steven Kembel <steve.kembel@gmail.com>

References


See Also

- mpd, comdist

Examples

```r
data(phylocom)
raoD(phylocom$sample)
raoD(phylocom$sample, phylocom$phylo)
```
readsample  

**Read Phylocom sample**

**Description**
Reads a Phylocom sample file and converts to a community data matrix.

**Usage**
```r
readsample(filename = "")
```

**Arguments**
- `filename`  Phylocom sample file path

**Value**
Community data matrix

**Author(s)**
Steven Kembel <skembel> and Cam Webb <cwebb@oeb.harvard.edu>

**References**

sample2matrix  

**Convert Phylocom sample to community data matrix**

**Description**
Convert a Phylocom database-format sample to community data matrix.

**Usage**
```r
sample2matrix(x)
```

**Arguments**
- `x`  Phylocom sample formatted data frame, a data frame with three columns:
  - Column 1 Community name
  - Column 2 Species abundance
  - Column 3 Species name
ses.mntd

**Author(s)**
Steven Kembel <steve.kembel@gmail.com> and Cam Webb <cwebb@oeb.harvard.edu>

**References**
et/phylocom/.

---

**ses.mntd**

*Standardized effect size of MNTD*

**Description**
Standardized effect size of mean nearest taxon distances in communities. When used with a phylo-
genetic distance matrix, equivalent to -1 times the Nearest Taxon Index (NTI).

**Usage**
```
ses.mntd(samp, dis, null.model = c("taxa.labels", "richness", "frequency",
  "sample.pool", "phylogeny.pool", "independentswap", "trialswap"),
  abundance.weighted=FALSE, runs = 999, iterations = 1000)
```

**Arguments**
- `samp`: Community data matrix
- `dis`: Distance matrix (generally a phylogenetic distance matrix)
- `null.model`: Null model to use (see Details section for description)
- `abundance.weighted`: Should mean nearest taxon distances for each species be weighted by species
  abundance? (default = FALSE)
- `runs`: Number of randomizations
- `iterations`: Number of iterations to use for each randomization (for independent swap and
  trial null models)

**Details**
The metric used by this function has also been referred to as MNND (mean nearest neighbour
distance), and the function was named `ses.mnnd` in picante versions < 0.7.

Currently implemented null models (arguments to `null.model`):
- `taxa.labels`: Shuffle distance matrix labels (across all taxa included in distance matrix)
- `richness`: Randomize community data matrix abundances within samples (maintains sample species
  richness)
frequency Randomize community data matrix abundances within species (maintains species occurrence frequency)

sample.pool Randomize community data matrix by drawing species from pool of species occurring in at least one community (sample pool) with equal probability

phylogeny.pool Randomize community data matrix by drawing species from pool of species occurring in the distance matrix (phylogeny pool) with equal probability

independentswap Randomize community data matrix with the independent swap algorithm (Gotelli 2000) maintaining species occurrence frequency and sample species richness

trialsawp Randomize community data matrix with the trial-swap algorithm (Miklos & Podani 2004) maintaining species occurrence frequency and sample species richness

Value

A data frame of results for each community

ntaxa Number of taxa in community
mntd.obs Observed MNTD in community
mntd.rand.mean Mean MNTD in null communities
mntd.rand.sd Standard deviation of MNTD in null communities
mntd.obs.rank Rank of observed MNTD vs. null communities
mntd.obs.z Standardized effect size of MNTD vs. null communities (= (mntd.obs - mntd.rand.mean) / mntd.rand.sd, equivalent to -NTI)
mntd.obs.p P-value (quantile) of observed MNTD vs. null communities (= mntd.obs.rank / runs + 1)
runs Number of randomizations

Author(s)

Steven Kembel <steve.kembel@gmail.com>

References


See Also

mntd, randomizeMatrix

Examples

data(phylocom)

ses.mntd(phylocom$sample, cophenetic(phylocom$phylo),null.model="taxa.labels")
Standardized effect size of MPD

Description

Standardized effect size of mean pairwise distances in communities. When used with a phylogenetic distance matrix, equivalent to -1 times the Nearest Relative Index (NRI).

Usage

```r
ses.mpd(samp, dis, null.model = c("taxa.labels", "richness", "frequency", "sample.pool", "phylogeny.pool", "independentswap", "trialswap"), abundance.weighted = FALSE, runs = 999, iterations = 1000)
```

Arguments

- **samp**: Community data matrix
- **dis**: Distance matrix (generally a phylogenetic distance matrix)
- **null.model**: Null model to use (see Details section for description)
- **abundance.weighted**: Should mean nearest taxon distances for each species be weighted by species abundance? (default = FALSE)
- **runs**: Number of randomizations
- **iterations**: Number of iterations to use for each randomization (for independent swap and trial null models)

Details

Currently implemented null models (arguments to null.model):

- **taxa.labels**: Shuffle distance matrix labels (across all taxa included in distance matrix)
- **richness**: Randomize community data matrix abundances within samples (maintains sample species richness)
- **frequency**: Randomize community data matrix abundances within species (maintains species occurrence frequency)
- **sample.pool**: Randomize community data matrix by drawing species from pool of species occurring in at least one community (sample pool) with equal probability
- **phylogeny.pool**: Randomize community data matrix by drawing species from pool of species occurring in the distance matrix (phylogeny pool) with equal probability
- **independentswap**: Randomize community data matrix with the independent swap algorithm (Gotelli 2000) maintaining species occurrence frequency and sample species richness
- **trialswap**: Randomize community data matrix with the trial-swap algorithm (Miklos & Podani 2004) maintaining species occurrence frequency and sample species richness
Value

A data frame of results for each community

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ntaxa</td>
<td>Number of taxa in community</td>
</tr>
<tr>
<td>mdp.obs</td>
<td>Observed mdp in community</td>
</tr>
<tr>
<td>mdp.rand.mean</td>
<td>Mean mdp in null communities</td>
</tr>
<tr>
<td>mdp.rand.sd</td>
<td>Standard deviation of mdp in null communities</td>
</tr>
<tr>
<td>mdp.obs.rank</td>
<td>Rank of observed mdp vs. null communities</td>
</tr>
<tr>
<td>mdp.obs.z</td>
<td>Standardized effect size of mdp vs. null communities (= (mpd.obs - mdp.rand.mean) / mdp.rand.sd, equivalent to -NRI)</td>
</tr>
<tr>
<td>mdp.obs.p</td>
<td>P-value (quantile) of observed mdp vs. null communities (= mdp.obs.rank / runs + 1)</td>
</tr>
<tr>
<td>runs</td>
<td>Number of randomizations</td>
</tr>
</tbody>
</table>

Author(s)

Steven Kembel <steve.kembel@gmail.com>

References


See Also

mpd, randomizematrix

Examples

data(phylocom)

ses.mpd(phylocom$sample, cophenetic(phylocom$phylo), null.model="taxa.labels")

ses.pd

Standardized effect size of PD

Description

Standardized effect size of phylogenetic diversity (Faith’s PD) in communities.

Usage

ses.pd(samp, tree, null.model = c("taxa.labels", "richness", "frequency", "sample.pool", "phylogeny.pool", "independentswap", "trialswap"), runs = 999, iterations = 1000, ...)
Arguments

samp  Community data matrix
tree  Phylogeny (phylo object)
null.model  Null model to use (see Details section for description)
runs  Number of randomizations
iterations  Number of iterations to use for each randomization (for independent swap and trial null models)
...  Additional arguments to pd function

Details

Currently implemented null models (arguments to null.model):

taxa.labels  Shuffle taxa labels across tips of phylogeny (across all taxa included in phylogeny)
richness  Randomize community data matrix abundances within samples (maintains sample species richness)
frequency  Randomize community data matrix abundances within species (maintains species occurrence frequency)
sample.pool  Randomize community data matrix by drawing species from pool of species occurring in at least one community (sample pool) with equal probability
phylogeny.pool  Randomize community data matrix by drawing species from pool of species occurring in the phylogeny (phylogeny pool) with equal probability
independentswap  Randomize community data matrix with the independent swap algorithm (Gotelli 2000) maintaining species occurrence frequency and sample species richness
trialsawp  Randomize community data matrix with the trial-swap algorithm (Miklos & Podani 2004) maintaining species occurrence frequency and sample species richness

Value

A data frame of results for each community

ntaxa  Number of taxa in community
pd.obs  Observed PD in community
pd.rand.mean  Mean PD in null communities
pd.rand.sd  Standard deviation of PD in null communities
pd.obs.rank  Rank of observed PD vs. null communities
pd.obs.z  Standardized effect size of PD vs. null communities (= (pd.obs - pd.rand.mean) / pd.rand.sd)
pd.obs.p  P-value (quantile) of observed PD vs. null communities (= mpd.obs.rank / runs + 1)
runs  Number of randomizations

Author(s)

Steven Kembel <steve.kembel@gmail.com>
References


See Also

pd, randomizematrix

Examples

data(phylocom)
ses.pd(phylocom$sample, phylocom$phylo, null.model="taxa.labels", runs=99)

specaccum.psr  Phylogenetic Species Richness Sample-Based Rarefaction Curve

Description

Finds a sample-based rarefaction curve for phylogenetic species richness for a set of samples.

Usage

specaccum.psr(samp, tree, permutations = 100, method = "random", ...)

Arguments

samp  Community data matrix

tree  A phylo tree object or a phylogenetic covariance matrix

permutations  Number of permutations with method method= "random"

method  Species accumulation method, currently only "random" is supported which adds samples in random order.

...  Other parameters to functions

Value

The function returns an object of class "specaccum" with items:

call  Function call.

method  Accumulator method.

sites  Number of sites/samples.

richness  The mean phylogenetic species richness corresponding to number of sites/samples.
species.dist

sd  The standard deviation of phylogenetic species accumulation curve (or its standard error) estimated from permutations in method = "random".

perm  Permutation results with method = "random" and NULL in other cases. Each column in perm holds one permutation.

Author(s)
Matthew Helmus <mrhelmus@gmail.com> based on the vegan package specaccum function by Roeland Kindt and Jari Oksanen.

References


See Also
psr, specaccum

Examples

data(phylocom)
accum.sr<-specaccum(phylocom$sample, permutations = 100, method = "random")
plot(accum.sr, col="blue")
points(accum.sr$sites, accum.sr$richness, pch=19, col="blue")

accum.psr<-specaccum.psr(phylocom$sample, phylocom$phylo, permutations = 100, method = "random")
plot(accum.psr, add=TRUE, col = "red")
points(accum.psr$sites, accum.psr$richness, pch=19, col="red")

legend(5,5,legend=c("SR","PSR"),pch=c(19,19),col=c("blue","red"))

Description
Compute interspecific distances based on patterns of species co-occurrence in communities.

Usage

species.dist(x, metric = c("cij", "jaccard", "checkerboard", "doij"))
sppregs

Arguments

x Community data matrix
metric Co-occurrence metric to use (see Details section for description)

Details

Currently implemented co-occurrence measures (arguments to metric):

cij Schoener’s index of co-occurrence
jaccard Jaccard index of co-occurrence
checkerboard Checkerboard index of co-occurrence
doij DOij index of co-occurrence

Value

A dist object with co-occurrences among all species pairs

Author(s)

Steven Kembel <steve.kembel@gmail.com>

References


See Also

vegdist

sppregs  Regressions to Separate Phylogenetic Attraction and Repulsion

Description

Fit regressions on species abundance or presence/absence across communities and calculate phylogenetic correlations

Usage

sppregs(samp, env, tree=NULL, fam="gaussian")
sppregs.plot(sppreg, rows=c(1,3), cex.mag=1, x.label="phylogenetic correlations", y.label=c("occurrence correlations w/ env", "occurrence correlations wo/ env", "change in correlations"))
Arguments

- **samp**: community data matrix, species as columns, communities as rows
- **env**: environmental data matrix
- **tree**: phylo tree object or a phylogenetic covariance matrix
- **fam**: with `fam = "gaussian"` fits with `glm`; with `fam = "binomial"` fit logistic regressions with Firth’s bias-reduction using `brglm`
- **sppreg**: object from function `sppregs`
- **rows**: `rows = c(1,3)` plots in a row; `rows = c(3,1)` in a column
- **cex.magn**: value for `cex` in `par`
- **x.label**: x axis labels
- **y.label**: y axis labels

Details

For each species in `samp`, the function fits regressions of species presence/absence or abundances on the environmental variables supplied in `env`; and calculates the \((n^2-n)/2\) pairwise species correlations between the residuals of these fits and pairwise species phylogenetic correlations. The residuals can be thought of as the presence/absence of species across sites/communities after accounting for how species respond to environmental variation across sites. Each set of coefficients can be tested for phylogenetic signal with, for example, the function `phylosignal`.

The function `sppregs.plot` produces a set of three plots of the correlations of pairwise species phylogenetic correlations versus: the observed pairwise correlations of species across communities, the residual correlations, and the pairwise differences between (i.e., the change in species co-occurrence once the environmental variables are taken into account). The significance of these correlations can be tested via permutation with the function `phylostruct`.

Value

- **family**: the regression error distribution
- **residuals**: the residuals from each species regression
- **coefficients**: the estimated coefficients from each species regression
- **std.errors**: the standard errors of the coefficients
- **correlations**: correlations of pairwise species phylogenetic correlations between: the observed pairwise correlations of species across communities, the residual correlations, and the pairwise differences between the two
- **cors.pa**: the observed pairwise correlations of species across communities
- **cors.resid**: the residual pairwise correlations of species across communities
- **cors.phylo**: the phylogenetic pairwise correlations among species

Note

The function requires the library `brglm` to perform logistic regressions
Author(s)
Matthew Helmus <mrhelmus@gmail.com>

References

See Also
phylostruct, phylosignal

tax.distinctiveness  

---

Description
Taxic diversity: Vane-Wright et al., 1991 and May 1990 which accounts for polytomies by counting the number of branches descending from each node that lies on the path from a spp tip to the root (not just counting the number of nodes).

Usage
tax.distinctiveness(tree, type = c("Vane-Wright", "May"))

Arguments
tree an object of class phylo
type specify "Vane-Wright" or "May"

Author(s)
Karen Magnuson-Ford, Will Cornwell, Arne Mooers, Mark Vellend

References


**traitgram**

*Draw phylogeny with nodes at trait positions*

**Description**

Draws a phylogeny where x position of nodes and tips corresponds to value of a continuous trait variable, and y position corresponds to node depth (i.e. age).

**Usage**

```r
traitgram(x, phy, xaxt = 's', underscore = FALSE, show.names = TRUE,
show.xaxis.values = TRUE, method = c('ML','pic'), ...)```

**Arguments**

- **x**: Trait vector (same order as phy$tip.label, or with taxon names in names)
- **phy**: phylo object
- **xaxt**: x axis default type
- **underscore**: if FALSE remove underscore from taxonomic names
- **show.names**: if TRUE show taxon names across tips of phylogeny
- **show.xaxis.values**: if TRUE show values for trait on x=axis
- **method**: method for calculation of internal trait values. 'ML' = maximum likelihood method; 'pic' = independent contrasts method. pic option can be used when ML fails to converge or otherwise seems to fail to correctly reconstruct ancestral values
- **...**: Additional arguments passed to plot

**Value**

Plots a traitgram, no values returned.

**Author(s)**

David Ackerly <dackerly@berkeley.edu>

**References**


Examples

```r
randtree <- rcoal(20)
randtraits <- rTraitCont(randtree)
traitgram(randtraits, randtree)
traitgram(randtraits, randtree, method = 'pic')
```

---

**unifrac**

*Unweighted UniFrac distance between communities*

### Description

Calculates unweighted UniFrac, a phylogenetic beta diversity metric of the unique (non-shared) fraction of total phylogenetic diversity (branch-length) between two communities.

### Usage

```r
unifrac(comm, tree)
```

### Arguments

- `comm` Community data matrix
- `tree` Object of class phylo - a rooted phylogeny

### Value

A dist object of the unweighted UniFrac distances between communities (the unique (non-shared) fraction of total phylogenetic diversity (branch-length) between two communities).

### Warning

The UniFrac distance between samples will include the branch length connecting taxa in those samples and the root of the supplied tree. The root of the supplied tree may not be spanned by any taxa in the sample. If you want the root of your tree to correspond to the most recent ancestor of the taxa actually present in your samples, you should prune the tree before running `unifrac`:

```r
prunedTree <- prune.sample(sample, tree)
```

### Note

The supplied tree must be rooted. Single-species samples will be assigned a PD value equal to the distance from the root to the present.

### Author(s)

Steven Kembel <steve.kembel@gmail.com>
utility

References

See Also
pd

Examples
data(phylocom)
unifrac(phylocom$sample, phylocom$phylo)

utility  Picante utility functions

Description
Picante utility functions for tree and data manipulation

Usage

df2vec(x, colID=1)
internal2tips(phy, int.node, return.names = FALSE)
node.age(phy)
pic.variance(x, phy, scaled = TRUE)
sortColumns(x)
sortRows(x)
taxaShuffle(x)
tipShuffle(phy)

Arguments

phy      phylo object
x        A data.frame, matrix or dist object
colID    Numeric or character ID of column to include
int.node internal node number
return.names TRUE or FALSE
scaled   Scale contrasts by branch length
...      Additional arguments

Details
Various utility functions for manipulating trees, data, etc.
writesample

Value

df2vec        A named vector
internal2tips Vector of tips descended from a node
node.age      Phylo object with phylo$ages vector of node ages corresponding to phylo$edge
pic.variance  Variance of independent contrasts
sortColumns   A data.frame or matrix with columns sorted by name
sortRows      A data.frame or matrix with rows sorted by name
taxaShuffle   Matrix with taxa names shuffled
tipShuffle    Phylo object with taxa names shuffled

Author(s)
Steven Kembel <steve.kembel@gmail.com>, Peter Cowan <pdc@berkeley.edu>, David Ackerly <dackerly@berkeley.edu>

writesample  Write a Phylocom community sample file

Description
Write a community data matrix to a Phylocom community sample file

Usage
writesample(community, filename = "")

Arguments
community    Community data matrix
filename     Filename path

Author(s)
Steven Kembel <steve.kembel@gmail.com> and Cam Webb <cwebb@oeb.harvard.edu>

References
writetraits

**writetraits** Write a Phylocom traits formatted file

**Description**
Write a Phylocom traits formatted file

**Usage**

writetraits(trt, file = "", bin = NULL, sigd = 3)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trt</td>
<td>Data frame containing trait data</td>
</tr>
<tr>
<td>file</td>
<td>Filename path</td>
</tr>
<tr>
<td>bin</td>
<td>Vector index of trait columns to be treated as binary</td>
</tr>
<tr>
<td>sigd</td>
<td>Significant digits for output</td>
</tr>
</tbody>
</table>

**Author(s)**

David Ackerly <dackerly@berkeley.edu> and Steven Kembel <steve.kembel@gmail.com>

**References**

# Index

*Topic **IO**
  - readsample, 38
  - sample2matrix, 38

*Topic **color**
  - color.plot.phylo, 3

*Topic **datasets**
  - IvesGodfray, 13
  - phylocom, 26

*Topic **file**
  - writesample, 52
  - writetraits, 53

*Topic **manip**
  - matrix2sample, 16
  - prune.sample, 32
  - randomizeMatrix, 35
  - utility, 51

*Topic **package**
  - picante-package, 2

*Topic **univar**
  - comdist, 4
  - comdistnt, 5
  - comm.phylo.cor, 6
  - comm.phylo.qr, 8
  - cor, 10
  - cor.table, 10
  - ead(expected.pd), 11
  - evol.distinct, 10
  - expected.pd, 11
  - glm, 47
  - internal2tips(utility), 51
  - IvesGodfray, 13
  - Kcalc, 14, 15, 28
  - match.comm.dist(match.phylo.data), 15
  - match.phylo.comm(match.phylo.data), 15
  - matrix2matrix, 14, 15, 19, 27
  - matrix2sample, 16
  - mnd, 34
  - mmnd(mntd), 17
  - mntd, 6, 17, 40
  - raoD, 36
  - ses.mntd, 39
  - ses.mpd, 41
  - ses.pd, 42
  - specaccum.psr, 44
  - species.dist, 45
  - sppregs, 46
  - traitgram, 49
  - unifrac, 50
  - brglm, 47
  - color.plot.phylo, 3
  - comdist, 4, 37
  - comdistnn(comdistnt), 5
  - comm.phylo.cor, 6
  - comm.phylo.qr, 8
  - cor, 10
  - cor.table, 10
  - df2vec(utility), 51
  - ead(expected.pd), 11
  - evol.distinct, 10
  - expected.pd, 11
  - glm, 47
  - internal2tips(utility), 51
  - IvesGodfray, 13
  - Kcalc, 14, 15, 28
  - match.comm.dist(match.phylo.data), 15
  - match.phylo.comm(match.phylo.data), 15
  - matrix2matrix, 14, 15, 19, 27
  - matrix2sample, 16
  - mnd, 34
  - mmnd(mntd), 17
  - mntd, 6, 17, 40
INDEX

mpd, 5, 18, 34, 37, 42
multiPhylosignal, 19
node.age (utility), 51

optim, 20, 21

pblm, 13, 19
pblmpredict (pblm), 19
PCD (pcd), 22
pcd, 22
pd, 12, 24, 29, 43, 44, 51
phyEstimate, 25
phyEstimateDisc (phyEstimate), 25
phylocom, 26
phylosignal, 14, 15, 19, 27, 47, 48
phylosor, 23, 28, 31
phylosor.rnd, 29, 29
phylostruct, 31, 47, 48
pic.variance (utility), 51
picante (picante-package), 2
picante-package, 2
prune.missing, 16
prune.missing (prune.sample), 32
prune.sample, 16, 32
psc (psd), 33
psd, 31, 32, 33
pse (psd), 33
psr, 25, 45
psr (psd), 33
PSV, 22
psv, 23
psv (psd), 33

randomizeMatrix, 7–9, 31, 32, 35, 40, 42, 44
raoD, 15, 36
reads sampled, 38
rq, 8

class2matrix, 38
ses.mnnd (ses.mntd), 39
ses.mntd, 6, 17, 39
ses.mpd, 5, 18, 41
ses.pd, 24, 25, 42
sortColumns (utility), 51
sortRows (utility), 51
specaccum, 45
specaccum.psr, 34, 44
species.dist, 7, 8, 45

sppregs, 31, 32, 46, 47
taxaShuffle (utility), 51
tipShuffle (utility), 51
traitgram, 49
unifrac, 23, 50
utility, 51

variance.pd (expected.pd), 11
vegdist, 46

writesample, 52
writetraits, 53