Package 'spacodiR'

April 4, 2011

Type Package

Title Spatial and Phylogenetic Analysis of Community Diversity

Version 0.11.04012011

Date 2011-04-01

Author Jonathan Eastman, Timothy Paine, and Olivier Hardy

Maintainer Jonathan Eastman <jonathan.eastman@gmail.com>

Depends picante, colorspace, Rcpp

LinkingTo Rcpp

Description SPACoDi is primarily designed to characterise the
structure and phylogenetic diversity of communities using
abundance or presence-absence data of species among community plots.

License GPL (>=2)

LazyLoad yes

Archs i386, x86_64

R topics documented:

as.phylocom *converting between data formats for community phylogenetics*

Description

as.phylocom converts from spacodi or picante data formats into phylocom format.

Usage

```
as.phylocom(data, picante=FALSE, outfile = NULL)
```

Arguments

data	a community dataset in either **spacodiR** or **picante** format (with species as row names and plots as column names or vice versa)
picante	Boolean; if current data format is spacodi, argument should be picante=FALSE; otherwise, format is assumed to be for picante, with plots as row names
outfile	an optional text file to which to write output

Details

This utility converts a species-by-plots matrix into triplet format, which is readable by the external program phylocom. If picante=TRUE, the data are expected to be in the form used for **picante** (i.e., a plots-by-species matrix; picante-package). If the user selects picante=FALSE, the data are expected to be in the form used for **spacodiR** (i.e., a species-by-plots matrix). The user has the option to save an output file, defined by outfile.

Value

A named array, formatted for use in phylocom; note that while the R-object returned by this function has column names, if output is written to a file, the header is dropped (as appropriate for use in the external phylocom executable: http://www.phylodiversity.net/phylocom/).

Author(s)

Jonathan Eastman

References

WEBB CO, DD ACKERLY and SW KEMBEL. 2008. Phylocom: software for the analysis of phylogenetic community structure and trait evolution. Bioinformatics 24:2098-2100.

See Also

`as.spacodi` and `as.picante` for converting between `phylocom` and `SPACoDi` formats; see `picante-package` for an R-port of `phylocom`

Examples

```
# call example data from SPACoDi
data(sp.example)
attach(sp.example)
spl->d.spacodi
d.spacodi ## SPACoDi format

# convert to phylocom
as.phylocom(data=spl, picante=FALSE)->d.phylocom
d.phylocom ## phylocom format

# convert dataset to picante
as.picante(data=d.phylocom)->d.picante
d.picante ## picante format

# convert back to SPACoDi
as.spacodi(data=d.picante)
```

`as.picante` *converting between data formats for community phylogenetics*

Description

`as.picante` converts from `spacodi` or `phylocom` data formats into `picante` format.

Usage

```
as.picante(data, outfile = NULL)
```

Arguments

`data`	a community dataset in `phylocom` or `spacodi` format
`outfile`	an optional text file to which to write output

Details

This utility converts a community dataset (either from `phylocom` or **spacodiR**) into a format interpretable by **picante** (see `picante-package`). `phylocom` format is also referred to as `triplet`-formatting, where plots are within the first column, abundances in the second, and species names in the third column of the dataframe. The user has the option to save an output file, defined by `outfile`. `SPACoDi` format is similar to that for **picante**, where dataframes between these packages are transposed. `SPACoDi` format should have species as row names.

Value

An array, formatted for use in `picante`

Author(s)

Jonathan Eastman

See Also

as.spacodi and as.phylocom for converting between phylocom and SPACoDi formats; see picante-package for an R-port of phylocom

Examples

```
# call example data from SPACoDi
data(sp.example)
attach(sp.example)
spl->d.spacodi
d.spacodi ## SPACoDi format

# convert to phylocom
as.phylocom(data=spl, picante=FALSE)->d.phylocom
d.phylocom ## phylocom format

# convert dataset to picante
as.picante(data=d.phylocom)->d.picante
d.picante ## picante format

# convert back to SPACoDi
as.spacodi(data=d.picante)

# run standardized effect size mean nearest taxon distances in picante
ses.mntd(as.picante(spl), cophenetic(phy))
```

---

as.spacodi *converting between data formats for community phylogenetics*

Description

as.spacodi converts from picante or phylocom data formats into spacodi format.

Usage

```
as.spacodi(data, outfile = NULL)
```

Arguments

data	a community dataset in phylocom or picante format
outfile	an optional text file to which to write output

Details

This utility converts a community dataset (either from **phylocom** or **picante** (see `picante`) into a format interpretable by either this R-package **spacodiR** or the external program `SPACoDi`, a Windows executable (available at `http://ebe.ulb.ac.be/ebe/Software.html`). Note also that the community-dataset format used here is also that called for by the package **vegan**; see `vegandocs`.

`phylocom` format is also referred to as `triplet`-formatting, where plots are within the first column, abundances in the second, and species names in the third column of the dataframe. `picante` format is simply the transpose of `spacodiR`-formatting of the community dataset: in `spacodiR`, species are expected as the row names of the dataframe, where plots are represented as the column names. The user has the option to save an output file, defined by `outfile`.

Value

An array, formatted for use in `spacodiR`

Author(s)

Jonathan Eastman

See Also

`as.phylocom`; `as.picante`

Examples

```
# call example data from spacodiR
data(sp.example)
attach(sp.example)
spl->d.spacodi
d.spacodi ## SPACoDi format

# convert to phylocom
as.phylocom(data=d.spacodi, picante=FALSE)->d.phylocom
d.phylocom ## phylocom format

# convert dataset to picante
as.picante(data=d.phylocom)->d.picante
d.picante ## picante format

# convert back to SPACoDi
as.spacodi(data=d.picante)
```

`match.spacodi.data` *pruning a tree to match datasets used in SPACoDi*

Description

`match.spacodi.data` is a utility for automating the process of pruning datasets for use in **spacodiR**. Note that nearly all **spacodiR** functions require that trees have present (at least) all species sampled in the community dataset.

Usage

```
match.spacodi.data(sp.plot, phy=NULL, sp.traits=NULL, prune=TRUE, verbose=FALSE)
```

Arguments

`sp.plot` a community dataset in **spacodiR** format (see `as.spacodi`)

`phy` a phylogenetic tree of class `phylo`; see `read.tree`

`sp.traits` a species-by-trait(s) dataframe

`prune` Boolean; whether plots with fewer than two species sampled are pruned

`verbose` Boolean; whether dropped plots are written to the console

Value

A list of pruned dataset(s):

- `sp.plot`
- `sp.tree`
- `sp.traits`

Note: if `prune=TRUE`, the list of plots pruned will be printed to the console.

Author(s)

Jonathan Eastman

Examples

```
# load a species-by-plots matrix, along with a tree
data(sp.example)
attributes(sp.example)
attach(sp.example)
spl
phy
trt

# prune out undersampled plots
spl[,2]=0
match.spacodi.data(spl) -> sp.plot.new
sp.plot.new


# match datasets where sp.traits is smaller than the remainder
match.spacodi.data(sp.plot=spl, phy=phy, sp.traits=trt[1:6,])
```

phy.deresolve *collapse phylogenetic nodes into multichotomies*

Description

collapses phylogenetic dichotomies within a temporal range into multichotomies

Usage

```
phy.deresolve(phy, time.range = c(0, 0), relative = TRUE)
```

Arguments

phy	a phylogenetic tree of class phylo; see read.tree
time.range	a supplied temporal range, within which nodes will be collapsed
relative	whether relative branch lengths are used (e.g., root is 1; present is 0)

Details

If time.range is supplied as a single value, phy.deresolve will assume that the user intended a range whose minimum is zero (i.e., present; see **Examples** below). If no nodes are present within the supplied time.range, the tree will be returned unadulterated.

Value

A phylogenetic tree of class phylo

Author(s)

Jonathan Eastman, based on di2multi by Emmanuel Paradis

Examples

```
data(sp.example)
attach(sp.example)

## An example with an internal temporal bin
# arbitrarily transform tree to a root age of 100
phy$edge.length=phy$edge.length*(100/max(branching.times(phy)))

# collapse nodes within the middle half of the tree's history
mulphy=phy.deresolve(phy=phy, time.range=c(0.05, 0.95))

# compare result with original tree
plot(mulphy, show.tip.label=FALSE)
mtext("multifurcations in the middle nine-tenths")
dev.new()
plot(phy, show.tip.label=FALSE)
mtext("original phylogeny")

## An example with time.range given by a single value
mulphy=phy.deresolve(phy=phy, time.range=0.25) # affects the most recent quarter of the t
dev.new()
```

```
plot(mulphy, show.tip.label=FALSE)
mtext("multifurcations in the most recent quarter")

## An example showing effect of 'relative'
phy$edge.length=phy$edge.length*(100/max(branching.times(phy)))
dev.new()
plot(dd<-phy.deresolve(phy=phy, time.range=c(0,1), relative=FALSE))
mtext("relative = FALSE")
dev.new()
plot(rr<-phy.deresolve(phy=phy, time.range=c(0,1), relative=TRUE))
mtext("relative = TRUE")
```

phy.dotplot *EXPERIMENTAL: plotting members of communities (or plots) on a phylogeny*

Description

`phy.dotplot` is used to visually display species membership in communities, generating one plot per community

Usage

```
phy.dotplot(sp.plot, phy, edge.width = 0.2, lab.adj = c(0, 0), tips.adj = c(0.5,
```

Arguments

`sp.plot` a community dataset in **spacodiR** format (see `as.spacodi`)

`phy` a phylogenetic tree of class `phylo`; see `read.tree`

`edge.width` a value specifying the thickness of branches on the plotted phylogen(ies)

`lab.adj` a vector of two values, adjusting plotted group names, with respect to the x- and y-axes

`tips.adj` a vector of two values, adjusting the position of plotted symbols (referencing the x- and y-axes)

`tips.cex` a value for character expansion of plotted symbols for present species

`pch` a value for the plotting character for species-presence symbols: see `par`

`print.labs` Boolean; returns array of tip labels, group membership of species, and tree to the console

`outfile` an optional .pdf file to which to write output

`...` additional plotting options to be specified

Details

For several groups (fewer than sixteen, for optimal visibility), this function will generate a plot of phylogenies, group labels for each tree, and symbols demarcating a particular species that is present within the group. This may be useful in visual representations of phylogenetic overdispersion or underdispersion within plots (or other groups). Particularly the `tips.adj` option may require fiddling in order for symbols to appear *precisely* where they should.

Value

A collection of trees (or a single tree), where species present within a predefined group are represented with a dark circle at the tip (if `pch=21`).

Author(s)

Jonathan Eastman

Examples

```
data(sp.example)
attach(sp.example)

phy.dotplot(sp.plot=spl[,1:9], phy=phy, tips.adj=c(0.50,0.55), lab.adj=c(0,1))
```

`phy.nodetimes` *finding splitting times for nodes*

Description

`phy.nodetimes` is a function to collect all nodes (and times) that are within a particular time fraction of the total temporal extent of a phylogeny

Usage

```
phy.nodetimes(phy, time.range = c(0, 0), proportion = TRUE)
```

Arguments

`phy`	a phylogenetic tree of class `phylo`; see `read.tree`
`time.range`	the temporal slice of the tree defining the bounds for extracted nodes
`proportion`	Boolean; whether `time.range` is regarded as proportions of tree height

Details

Given a temporal fraction of a phylogeny, this function will return all contained nodes (and branching times if desired). If used with `spacodi.by.nodes`, one could conduct significance testing of `Bst` through particular time slices of the phylogeny. Note: when using `proportion=TRUE`, '0' is regarded as the present, '1' as the root of the tree (e.g., `start.time=1.0` and `stop.time=0.8` gathers nodes for the basalmost or most rootward 20 percent of the temporal extent of the tree).

Value

A named vector of times (whose names are node labels; see `nodelabels` for the function used to label nodes).

Author(s)

Jonathan Eastman

Examples

```
## an example of comparing observed and expected Bst at nodes ##
data(sp.example)
attach(sp.example)
s=spl
p=phy

# get nodes for a time-slice
n.t=phy.nodetimes(phy=p,time.range=c(0.25,0.75))

# get Bsts for nodes
o=spacodi.by.nodes(sp.plot=s, phy=p, obs.only=TRUE)[[1]]
obs=o[!is.na(match(o[,"node.ID"], names(n.t))),1]

# compile expected Bst-values from data reshuffling
spacodi.by.nodes(sp.plot=s, phy=p, n.rep=5, method="1s")[[2]]->exps
e=exps[!is.na(match(row.names(exps), names(n.t))),]
exp=unname(unlist(e))

# randomization test of observed and expected Bst
resamp.test(obs=obs, exp=exp, iter=100, two.tailed=TRUE)
```

`resamp.1a` *randomizing a community phylogenetics matrix: '1a' of Hardy (2008)*

Description

`resamp.1a` is used for resampling data within a community dataset

Usage

```
resamp.1a(obj, abund.class.ratio = 4)
```

Arguments

`obj`	a community dataset in **spacodiR** format (see `as.spacodi`)
`abund.class.ratio`	a ratio defining abundance classes

Details

A resampling procedure for a species-by-plots matrix, where species are shuffled within abundance classes. Species are grouped into distinct abundance classes characterized by a fixed ratio: `abund.class.ratio` = maximal abundance / minimal abundance. For instance, if `abund.class.ratio = 4`, the limits between abundance classes could be 1, 4, 16, ... Species are randomly permuted within each class, which maintains most of the abundance phylogenetic structure originally present in a dataset.

Value

A shuffled dataset

Author(s)

Timothy Paine and Jonathan Eastman

References

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

See Also

`spacodi.by.nodes` for permutation tests of community diversity and additional randomization methods; see `spacodi.calc` for the main underlying function; `randomizeMatrix`

Examples

```
data(sp.example)
attach(sp.example)
spl

# shuffle dataset
resamp.1a(obj=spl, abund.class.ratio=3)
```

`resamp.1s` *randomizing a community phylogenetics matrix: '1s' of Hardy (2008)*

Description

`resamp.1s` is used for resampling data within a community dataset

Usage

```
resamp.1s(obj)
```

Arguments

`obj`	a community dataset in **spacodiR** format (see `as.spacodi`)

Details

A resampling procedure for a species-by-plots matrix, where observed abundances are shuffled across species and plots.

Value

A shuffled dataset

Author(s)

Timothy Paine and Jonathan Eastman

References

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

See Also

`spacodi.by.nodes` for permutation tests of community diversity and additional randomization methods; see `spacodi.calc` for the main underlying function; `randomizeMatrix`

Examples

```
data(sp.example)
attach(sp.example)
spl

# shuffle dataset
resamp.1s(obj=spl)
```

`resamp.2s` *randomizing a community phylogenetics matrix: '2s' of Hardy (2008)*

Description

`resamp.2s` is used for resampling data within a community dataset

Usage

```
resamp.2s(obj)
```

Arguments

`obj`	a community dataset in **spacodiR** format (see `as.spacodi`)

Details

A resampling procedure for a species-by-plots matrix, where observed abundances are shuffled across species but within plots.

Value

A shuffled dataset

Author(s)

Jonathan Eastman

References

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

See Also

spacodi.by.nodes for permutation tests of community diversity and additional randomization methods; see spacodi.calc for the main underlying function; randomizeMatrix

Examples

```
data(sp.example)
attach(sp.example)
spl

# shuffle dataset
resamp.2s(obj=spl)
```

resamp.2x *randomizing a community phylogenetics matrix: '2x' of Hardy (2008)*

Description

`resamp.2x` is used for resampling data within a community dataset

Usage

```
resamp.2x(obj, level=0.1)
```

Arguments

`obj`	a community dataset in **spacodiR** format (see `as.spacodi`)
`level`	a proportion specifying the extent of data shuffling

Details

A resampling procedure for a species-by-plots matrix, based on Gotelli swapping. Shuffles abundances within a pair of plots and for a pair of species. The `level` defines the degree of sampling, with larger values dictating a higher level of reshuffling. For instance, if `level = 0.4` and the dataset involves 5 species and 10 plots, a total of 20 (`0.4x5x10`) Gotelli swaps are performed.

Value

A shuffled dataset

Author(s)

Jonathan Eastman

References

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

GOTELLI NJ. 2000. Null model analysis of species co-occurrence patterns. Ecology 81:2606-2621.

See Also

`spacodi.by.nodes` for permutation tests of community diversity and additional randomization methods; see `spacodi.calc` for the main underlying function; `randomizeMatrix`

Examples

```
data(sp.example)
attach(sp.example)
spl

# shuffle dataset
resamp.2x(obj=spl, level=0.2)
```

`resamp.3i` *randomizing a community phylogenetics matrix: '3i' of Hardy (2008)*

Description

`resamp.3i` is used for resampling data within a community dataset

Usage

```
resamp.3i(obj)
```

Arguments

`obj`	a community dataset in **spacodiR** format (see `as.spacodi`)

Details

A resampling procedure for a species-by-plots matrix, where observed abundances are shuffled within species and among plots.

Value

A shuffled dataset

Author(s)

Jonathan Eastman

References

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

See Also

`spacodi.by.nodes` for permutation tests of community diversity and additional randomization methods; see `spacodi.calc` for the main underlying function; `randomizeMatrix`

Examples

```
data(sp.example)
attach(sp.example)
spl

# shuffle dataset
resamp.3i(obj=spl)
```

`resamp.3t` *randomizing a community phylogenetics matrix: '3t' of Hardy (2008)*

Description

`resamp.3t` is used for resampling data within a community dataset

Usage

```
resamp.3t(obj, dmat=NULL)
```

Arguments

`obj`	a community dataset in **spacodiR** format (see `as.spacodi`)
`dmat`	an optional dataframe of distances between plots; row names and column names should be identical

Details

A resampling procedure for a species-by-plots matrix, where observed abundances within species are shuffled to adjacent plots. This procedure thus assumes meaningful arrangement of plots in space. If a distance matrix is supplied, the likelihood of shuffling to a particular plot is proportional to the distance between the plots.

Value

A shuffled dataset

Author(s)

Jonathan Eastman

References

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

See Also

`spacodi.by.nodes` for permutation tests of community diversity and additional randomization methods; see `spacodi.calc` for the main underlying function; `randomizeMatrix`

Examples

```
# define a community dataset
data(sp.example)
attach(sp.example)
spl

# define a distance matrix
foo <- matrix(runif((ncol(spl)->ss)^2,0,100),ss,ss)
foo[upper.tri(foo)] <- foo[lower.tri(foo)]
diag(foo) <- 0
dmat <- as.data.frame(foo)
row.names(dmat) <- names(spl)
names(dmat) <- row.names(dmat)


# shuffle dataset
resamp.3t(obj=spl, dmat=dmat)
spl ## comparison with original
```

`resamp.3x` *randomizing a community phylogenetics matrix: '3x' of Hardy (2008)*

Description

`resamp.3x` is used for resampling data within a community dataset

Usage

```
resamp.3x(obj, level=0.1)
```

Arguments

`obj`	a community dataset in **spacodiR** format (see `as.spacodi`)
`level`	a proportion specifying the extent of data shuffling

Details

A resampling procedure for a species-by-plots matrix, based on Gotelli swapping. Shuffles abundances within species and for a pair of plots. The `level` defines the degree of sampling, with larger values dictating a higher level of reshuffling. For instance, if `level = 0.1` and the dataset involves 20 species and 20 plots, a total of 40 (`0.1x20x20`) Gotelli swaps are performed.

Value

A shuffled dataset

Author(s)

Jonathan Eastman

References

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

GOTELLI NJ. 2000. Null model analysis of species co-occurrence patterns. Ecology 81:2606-2621.

See Also

`spacodi.by.nodes` for permutation tests of community diversity and additional randomization methods; see `spacodi.calc` for the main underlying function; `randomizeMatrix`

Examples

```
data(sp.example)
attach(sp.example)
spl

# shuffle dataset
resamp.3x(obj=spl, level=0.2)
```

`resamp.phy` *partial phylogeny randomization for tips*

Description

`resamp.phy` is used to shuffle tips subtended by a set of internal nodes determining by divergence time or specified by the user

Usage

```
resamp.phy(phy, node = NULL, time.threshold = 1, proportion = TRUE)
```

Arguments

`phy`	a phylogenetic tree of class `phylo`
`node`	numeric value(s), specifying the internal node(s) whose tips to shuffle; see `nodelabels`
`time.threshold`	either absolute or relative divergence times, used to define clades within which tips will be shuffled
`proportion`	if `TRUE`, divergence times are treated as relative (`0: present; 1: root`)

Details

Either a numeric vector is supplied for `node` or a `time.threshold`. If given a set of `nodes`, this function will naively shuffle tips descended from the nodes in the order supplied (without regard to whether any internal node in the vector is a descendant of any other node in the `node` vector). If given a `time.threshold`, tips will be reshuffled within non-nested clades that have a rootmost node that occurs within the range `[0, time.threshold]`. Note that regard to absolute divergence times can be enforced with `proportion=FALSE`. Note further that `resamp.phy(phy=phy, node=NULL, node.threshold=1, proportion=TRUE)` achieves the same effect as `resamp.1s`.

Value

a phylogenetic tree whose tips have been shuffled (without any modification of the underlying topology)

Author(s)

Jonathan Eastman

References

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

See Also

see `nodelabels` for values given for internal nodes and how these are retrieved

Examples

```
# load a tree
data(sp.example)
attach(sp.example)
plot(phy)
mtext("original phylogeny")

# reshuffle within a time range
time=1/3
bb=branching.times(phy)
bb=bb/max(bb)
nodes=(Ntip(phy)+1):max(phy$edge)
nodes[bb<=time]=1

dev.new()
plot(resamp.phy(phy, time.threshold=time, proportion=TRUE))
mtext("reshuffled phylogeny showing affected nodes")
nodelabels(cex=ifelse(nodes==1, 2, NA), col=ifelse(nodes==1, 1, NA), pch=19)
```

`resamp.test` *statistical comparison of sets of values by randomization*

Description

compares means by bootstrap resampling of differences between empirical distributions

Usage

```
resamp.test(obs = obs, exp = exp, mu = 0, iter = 10000, two.tailed = FALSE, na.r
```

Arguments

`obs`	a vector of numeric values
`exp`	a vector of numeric values
`mu`	the true difference in means
`iter`	number of randomization comparisons to perform
`two.tailed`	Boolean; as default, the test is performed under a one-tailed assumption
`na.rm`	whether to remove NA data

Details

If a single value is supplied for `obs`, this test equates to finding the quantile in `exp` in which `obs` would be found (under a one-tailed test); see **Examples** and also `ecdf`

Value

A list, whose contents are determined by the above argument:

`unnamed value`	if `two.tailed=TRUE`, this is the two-tailed p-value
`diffs`	the full resampling distribution of differences between `obs` and `exp`, given `mu`
`greater`	if `two.tailed=FALSE`, this is the p-value associated with the righthand tail
`lesser`	if `two.tailed=FALSE`, this is the p-value associated with the lefthand tail

Author(s)

Jonathan Eastman

Examples

```
## A GENERAL EXAMPLE
# a comparison between two distributions
a=rnorm(n=1000, mean=1, sd=0.5)
b=rnorm(n=1000, mean=0, sd=1)
resamp.test(obs=a, exp=b, two.tailed=FALSE)

# a comparison of a single value to a normal distribution
a=3
b=rnorm(n=1000, mean=0, sd=1)
resamp.test(obs=a, exp=b, two.tailed=FALSE)

# compare above result with ecdf(), in which we compute an empirical
f=ecdf(b)
print(1-f(a)) # analogous to a one-tailed test as above

## A COMMUNITY PHYLOGENETIC EXAMPLE
## an example of comparing observed and expected Bst at nodes ##
data(sp.example)
attach(sp.example)
s=spl
p=phy

# get nodes for a time-slice
```

```
n.t=phy.nodetimes(phy=p,time.range=c(0.25, 0.75))

# get Bsts for nodes
o=spacodi.by.nodes(sp.plot=s, phy=p, obs.only=TRUE)[[1]]
obs=o[!is.na(match(o[,"node.ID"], names(n.t))),1]

# compile expected Bst-values from data reshuffling
spacodi.by.nodes(sp.plot=s, phy=p, n.rep=5, method="1s")[[2]]->exps
e=exps[!is.na(match(row.names(exps), names(n.t))),]
exp=unname(unlist(e))

# randomization test of observed and expected Bst
resamp.test(obs=obs, exp=exp, iter=100, two.tailed=TRUE)
```

sp.example *example data for spacodiR*

Description

spacodi-package provides example community phylogenetics data, a phylogeny, and trait data

Usage

```
data(sp.example)
```

Details

This object is a list of example datasets for **spacodiR**: a community dataset (spl), phylogeny (phy), species-level traits (trt), and matrix of distances between plots (dis). The data can be called into R by 'data(sp.example)', where each item is indexed as an element of a list (e.g., for the tree, use sp.example$phy). More generally, documented examples can be run from the console: for instance, try 'example(spacodi.calc)'.

spacodi *main: Spatial and Phylogenetic Analysis of Community Diversity*

Description

spacodiR is primarily designed to characterise the structure and phylogenetic diversity of communities using abundance or presence-absence data of species among community plots. For estimation of community structure, see spacodi.calc. More information on SPACoDi can be found at http://ebe.ulb.ac.be/ebe/Software.html

Details

Package:	spacodi
Type:	Package
Version:	0.11
License:	GPL
LazyLoad:	yes

Author(s)

Jonathan Eastman, Timothy Paine, and Olivier Hardy <ohardy@ulb.ac.be>

References

OLIVIER J. HARDY and BRUNO SENTERRE. 2007. Characterizing the phylogenetic structure of communities by an additive partitioning of phylogenetic diversity. Journal of Ecology 95:493-506.

OLIVIER J. HARDY. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

`spacodi-internal` *internal SPACoDi functions*

Description

`spacodi-internal` are not typically called by the user

Details

This is an internal **spacodiR** function, not intended to be called directly by the user.

`spacodi.by.nodes` *conducting randomization tests of turnover in diversity through time*

Description

`spacodi.by.nodes` is a function for testing the null expectation of uniform diversity turnover across nodes of a phylogeny.

Usage

```
spacodi.by.nodes(sp.plot, phy, sp.parm = "Bst", obs.only = FALSE, return.all = T
```

Arguments

`sp.plot` a community dataset in **spacodiR** format (see `as.spacodi`)

`phy` a phylogenetic tree of class `phylo`; see `read.tree`

`sp.parm` a parameter of interest (one of `"Ist"`,`"Pst"`,`"Bst"`,`"PIst"` or analogs for trait data); see **Details**

`obs.only` Boolean; whether only estimated `sp.parms`s for the empirical data are returned

`return.all` Boolean; whether results for all nodes are returned for the empirical data (regardless NA estimates)

`n.rep` number of permutations to perform on the dataset `method` reshuffling or simulation procedure to perform (see **Details** for optional methods)

`parm` additional parameters, required for methods `"1a", "2x", and "3x"` (see **Details**)

`dmat` an optional matrix of spatial distances between plots, for use only with method `"3t"`

`rand.test` Boolean; whether a randomization test of observed and expected `sp.parms` is performed

`r.rep` number of randomization iterations to perform

`...` additional settings to be set for functions called by `spacodi.matrices`

Details

`spacodi.by.nodes` is a function to conduct randomization tests on estimated and observed `sp.parmss` through time. See `spacodi.calc` for further information on the possible estimators to use, whether considering phylogenetic or trait diversity.

`spacodi.by.nodes` computes measures of community structure for nodes of a phylogeny that subtend greater than two species and where nodes subtend species that are sampled in multiple plots.

Note: whether for the empirical data or for permuted datasets, if fewer than two species are sampled in a given plot, the plot will be dynamically removed from the dataset. This behavior can be controlled directly in `spacodi.calc` with the `prune` option.

For significance testing, permuted datasets may be generated by one of several `methods`:

- `method = "1a"` shuffling based on abundance classes of species (see `resamp.1a`); requires specification of `abund.class.ratio` as a `parm`
- `method = "1s"` shuffling of abundances across entire dataset (see `resamp.1s`)
- `method = "2s"` shuffling of abundances across species but within plots (see `resamp.2s`)
- `method = "2x"` Gotelli swapping of abundances for pairs of species and within a pair of plots (see `resamp.2x`); requires specification of `level` as a `parm`
- `method = "3i"` shuffling of abundances within species and among plots (see `resamp.3i`)
- `method = "3t"` shuffling of abundances to adjacent plots but within species (see `resamp.3t`); if supplied, `dmat` is used to condition abundance shuffling, where closer plots, on average, have higher likelihoods of abundances being swapped
- `method = "3x"` Gotelli swapping of abundances for pairs of plots and within a pair of species (see `resamp.3x`); requires specification of `level` as a `parm`

Value

`spacodi.by.nodes` returns a list of three elements:

- `observed.sp.parm`: estimated `sp.parm` for nodes of the tree (either all nodes, if `return.all=TRUE` or for only those nodes for which `sp.parm` is non-null)
- `expected.sp.parm`: expected `sp.parm` for all nodes of the emprical data for which `sp.parm` is calculable; nodes in this element of the output array are row names
- `randomization.test`: results from randomization tests of `sp.parm` for each node. Note: if randomization test is used, some randomization iterations may return NA for particular nodes (see details above). The tally of valid expected `sp.parms` from permutation is recorded as a column in this element (i.e., `randomization.test$valid.comparisons`). Be wary of significant results that are based off few valid comparisons.

Author(s)

Jonathan Eastman and Timothy Paine

References

HARDY OJ and B SENTERRE. 2007. Characterizing the phylogenetic structure of communities by an additive partitioning of phylogenetic diversity. Journal of Ecology 95:493-506.

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

See Also

`spacodi.calc` for interpretation of results; `resamp.test`; `randomizeMatrix`

Examples

```
# load a species-by-plots matrix, along with a tree
data(sp.example)
attributes(sp.example)
attach(sp.example)
spl
phy

# generate a plot of observed and expected Bst
sp.permut<-spacodi.by.nodes(sp.plot=spl, phy=phy, n.rep=35)
sp.permut$randomization.test

# plot observed and expected Bst
spacodi.permutplot(spacodi.permutations=sp.permut, envelope=FALSE)
```

`spacodi.calc` *measuring spatial and phylogenetic structuring of diversity in communities*

Description

Considering species-, phylogenetic-, or trait-diversities, this function measures diversity structuring of community samples.

Usage

```
spacodi.calc(sp.plot, phy = NULL, sp.traits = NULL, all.together = TRUE, prune =
```

Arguments

`sp.plot` a community dataset in **spacodiR** format (see `as.spacodi`)

`phy` a phylogenetic tree of class `phylo` or evolutionary distance matrix between species (see `cophenetic.phylo`)

`sp.traits` a species-by-trait(s) dataframe or a species traits distance matrix (see `dist`)

`all.together`	Boolean; whether to treat all traits together or separately
`prune`	Boolean; whether to dynamically prune datasets if mismatches occur
`pairwise`	Boolean; whether to return pairwise diversity measures amongst all plots
`...`	additional arguments to be passed to `match.spacodi.data`

Details

`spacodi.calc` **requires** a community dataset (species-by-plots matrix; `sp.plot`) of numerical abundance, relative abundance, or presence | absence for plots. `spacodi.calc` returns statistics of diversity partitioning of plots, considering species diversity and, if additional information is provided, **either** trait or phylogenetic diversities among plots. If `phy=NULL` and `sp.traits=NULL`, a measure of partitioning for species diversity will be returned.

In treating each pair of plots as a community unto its own, `pairwise=TRUE` will return estimates for diversity structuring for all pairwise combinations of plots.

If a phylogeny or trait dataset is supplied with species that are not present in the community dataset (i.e., `sp.plot`) or vice versa, the user has the option to dynamically prune these datasets to match (`prune=TRUE`). If `prune=FALSE` and dataset mismatches occur, the function will inevitably return `NaN` where plots have fewer than two distinct species sampled.

Value

A named list of at least one element (`Ist`) is returned. The size of the returned list is wholly dependent upon given arguments.

SPECIES DIVERSITY STRUCTURING

- `Ist`: a measure of local species identity excess between individuals, expressing species turnover. It is a form of spatial partition of Gini-Simpson diversity (equivalent to `Fst` in population genetics). `Ist` considers **only** abundances (or presences) in the species-by-plots matrix.

PHYLOGENETIC DIVERSITY STRUCTURING

- `Pst`: a measure of local phyletic proximity excess between individuals, expressing species + phylogenetic turnover. It is a form of spatial partition of Rao's quadratic entropy (equivalent to `Nst` in population genetics). `Tst` is the analogue for trait data, estimating the spatial partitioning of mean trait-divergence between individuals.
- `Bst`: a measure of local phyletic proximity excess between individuals of distinct species, expressing phylogenetic turnover (independently of species turnover). `Ust` is the analogue for trait data, estimating the spatial partitioning of mean trait-divergence between individuals that belong to distinct species.
- `PIst`: Pst analogue for presence/absence data, expressing phylogenetic turnover (independently of species turnover). `TAUst` is the analogue for trait data, estimating mean trait-divergence between distinct species.
- `spatial phylogenetic clustering`: species within plots are more related on average than species from distinct plots where `Pst > Ist`, `Bst > 0`, or `PIst > 0`.
- `spatial phylogenetic overdispersion`: species within plots are less related on average than species from distinct plots where `Pst < Ist`, `Bst < 0`, or `PIst < 0`.

TRAIT DIVERSITY STRUCTURING

- Measures analogous to those under PHYLOGENETIC DIVERSITY STRUCTURING can be computed from trait data. For trait data, these analogues are Tst (see Pst), Ust (see Bst), and TAUst (see PIst). Note: elsewhere, Ust will be referred to as T*st but here has been renamed to avoid issues of indexing in R. Trait values are not assumed to follow any particular model of evolution; rather, distances between observed species traits are expected to be uniform in distribution.
- If all.together=TRUE, all traits will be used to generate distance a distance matrix for sampled species. Where all.together=FALSE is used, output is generated for each trait independently.

Author(s)

Olivier Hardy, Timothy Paine, and Jonathan Eastman

References

HARDY OJ and B SENTERRE. 2007. Characterizing the phylogenetic structure of communities by an additive partitioning of phylogenetic diversity. Journal of Ecology 95:493-506.

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

HARDY OJ and L JOST. 2008. Interpreting and estimating measures of community phylogenetic structuring. Journal of Ecology 96:849-852.

See Also

match.spacodi.data; as.spacodi

Examples

```
# load a species-by-plots matrix, along with a tree
data(sp.example)
attributes(sp.example)
attach(sp.example)
spl
phy

# community diversity statistics of Hardy and Senterre (2007): tree-based
spacodi.calc(sp.plot = spl, phy = phy)

# community diversity statistics: trait-based with pairwise comparisons
spacodi.calc(sp.plot = spl, phy = phy, pairwise=TRUE)

# community diversity for a pair of traits
spacodi.calc(sp.plot = spl, sp.traits = trt, all.together=TRUE)

# community diversity for a pair of traits, each singly
spacodi.calc(sp.plot = spl, sp.traits = trt, all.together=FALSE)

# Ist: using abundance data only
spacodi.calc(sp.plot = spl)

# calculations with missing taxa between tree and sp.plot
# excluding the last five species in sp.plot,
spacodi.calc(sp.plot = spl[1:15,], phy = phy, prune=TRUE)

# as before but with 'manual' pruning of the datasets
match.spacodi.data(sp.plot=spl[1:15,],phy=phy) -> prn.data
spacodi.calc(sp.plot=prn.data$sp.plot, phy=prn.data$sp.tree)
prn.data$sp.plot
prn.data$sp.tree
```

`spacodi.permutplot` *EXPERIMENTAL: plotting observed and expected community structure across branching times of a phylogeny*

Description

`spacodi.permutplot` is a function used to plot departures from null expectations of diversity turnover, nodewise for a phylogeny.

Usage

```
spacodi.permutplot(spacodi.permutations, cex=list(pch=1.5, rand=0.1, node=0.5,
bg = list("white", "lightgray", "black"), all.points = TRUE, add.id = TRUE, si
```

Arguments

`spacodi.permutations`
a list-object generated by `spacodi.by.nodes`

`cex` a named list of scalings for node symbols (`pch`), permutations (`rand`), `node` labels, and `legend`

`transp` degree of color transparency: `transp=1` is opaque

`col` a list of two elements: colors to be used for plotting observed and expected community structure

`bg` a list of background (fill) colors to be used for plotting observed and expected community structure

`all.points` Boolean; whether all permutation estimates are to be plotted

`add.id` Boolean; whether `node.labels` are placed near nodewise estimates of community structure

`sig.plot` Boolean; whether nodes are colored by significance of observed and expected estimates of structure

`cut.off` a value, if `sig.plot=TRUE`, distinguishing observed from expected estimates of structure

`envelope` Boolean; whether a smooth-spline envelope is drawn around permutation estimates of structure

`outfile` an optional .pdf file to which to write output

`...` additional plotting parameters to be optionally supplied

Details

This function generates a plot of diversity turnover, evaluated for each subtree of the supplied phylogeny. If `all.points=TRUE`, empirical estimates of structure as well as those expected from permuted community datasets are plotted together. Several methods are available for data permutation when using `spacodi.by.nodes`, typically the function used to generate `spacodi.permutations`.

Value

a plot of diversity structure estimates for a phylogeny, with the option to direct output to a .pdf file

Author(s)

Jonathan Eastman

See Also

`spacodi.calc` for further details on diversity structure; `spacodi.by.nodes`; `smooth.spline`

Examples

```
# load a species-by-plots matrix, along with a tree
data(sp.example)
attach(sp.example)
spl
phy

# generate a plot of observed and expected Bst
sp.permut=spacodi.by.nodes(sp.plot=spl, phy=phy, n.rep=10)
spacodi.permutplot(sp.permut,bty="n")
```

`spacodi.treeplot` *EXPERIMENTAL: plotting diversity turnover on trees*

Description

`spacodi.treeplot` is used to plot departures from expectations for diversity turnover on a phylogeny

Usage

```
spacodi.treeplot(spacodi.permutations, phy, cex=list(pch = 1.5, tip = 0.5, legen
```

Arguments

`spacodi.permutations`	a list-object generated by `spacodi.by.nodes`
`phy`	a phylogenetic tree of class `phylo`; see `read.tree`
`cex`	a named list of character scalings for node-associated symbols (`pch`), tip labels (`tip`), and the legend (`legend`)
`transp`	degree of color transparency: `transp=1` is opaque

`sig.plot` Boolean; whether nodes are colored by significance of observed and expected estimates of structure

`cut.off` a value, if `sig.plot=TRUE`, distinguishing observed from expected estimates of structure

`cols` a list of three elements if using `sig.plot`: the first color is for values not different than expected; the second and third for values greater and less than expected

`main` Boolean; whether a title is plotted

`outfile` an optional .pdf file to which to write output

`add.id` Boolean; whether `node.labels` are placed near nodes; see `nodelabels`

`...` additional plotting parameters to be optionally supplied

Details

This function will compute and plot estimates of diversity structure on the tree, with color-coded values.

Note: this function requires a `spacodi.permutations` with data for all nodes. If using, for instance, `spacodi.by.nodes` to generate these data, the option `obs.only=TRUE` will ensure that data are returned for all nodes (see **Examples** below).

Value

a plot of diversity structure estimates for a phylogeny, with the option to direct output to a .pdf file

Author(s)

Timothy Paine and Jonathan Eastman

References

HARDY OJ and B SENTERRE. 2007. Characterizing the phylogenetic structure of communities by an additive partitioning of phylogenetic diversity. Journal of Ecology 95:493-506.

HARDY OJ. 2008. Testing the spatial phylogenetic structure of local communities: statistical performances of different null models and test statistics on a locally neutral community. Journal of Ecology 96:914-926.

See Also

see `spacodi.by.nodes` for estimating phylogenetic turnover on trees with community-level sampling; `spacodi.permutplot`; `phy.dotplot`

Examples

```
data(sp.example)
attach(sp.example)

# plot PIst
PI=spacodi.by.nodes(sp.plot=spl, sp.parm="PIst", phy=phy, return.all=TRUE, method="1s")
spacodi.treeplot(PI, phy, sig.plot=TRUE, add.id=FALSE)
spacodi.treeplot(PI, phy, sig.plot=FALSE, add.id=FALSE)
```

write.spacodi.data *saving community phylogenetics datasets to a file*

Description

write.spacodi.data generates community phylogenetic datasets to be used in the external program SPACoDi

Usage

```
write.spacodi.data(sp.plot, outfile)
```

Arguments

sp.plot	a community dataset formatted for the R-package **spacodiR** (see as.spacodi)
outfile	a formatted file for the Windows executable SPACoDi

Details

This utility writes a species-by-plots matrix into a format readable by the external program SPACoDi, a Windows executable (available at http://ebe.ulb.ac.be/ebe/Software.html).

Author(s)

Jonathan Eastman

See Also

as.spacodi for converting from phylocom to **spacodiR** formats; see spacodi.calc for community diversity measures of empirical data

Examples

```
# generate a community-phylogenetics dataset
data(sp.example)
attach(sp.example)

# save the dataset to working directory
write.spacodi.data(sp.plot=spl, outfile="spacodi.formatted.txt")
```

Index