Package ‘adephylo’

October 19, 2022

Version 1.1-13

Date 2022-10-19

Title Exploratory Analyses for the Phylogenetic Comparative Method

Author Thibaut Jombart <`t.jombart@imperial.ac.uk`>, Stéphane Dray <`stephane.dray@univ-lyon1.fr`>, Anders Ellern Bilgrau <`abilgrau@math.aau.dk`>

Maintainer Stéphane Dray <`stephane.dray@univ-lyon1.fr`>

Depends methods, ade4 (>= 1.7-10)

Imports phylobase, ape, adegenet

Description Multivariate tools to analyze comparative data, i.e. a phylogeny and some traits measured for each taxa.

License GPL (>= 2)

LazyLoad yes

RoxygenNote 7.2.1

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-10-19 13:45:07 UTC

R topics documented:

`adephylo-package` *The adephylo package*

Description

This package is devoted to exploratory analysis of phylogenetic comparative data. It re-implements and extends phylogenetic procedures from the `ade4` package (which are now deprecated).

Details

Comparative data (phylogeny+traits) are handled as phylo4d objects, a canonical class implemented by the `phylobase` package. Trees are handled as `phylo` objects (from the `ape` package) or as phylo4 objects (`phylobase`'s extension of `phylo` objects).

Main functionalities of `adephylo` are summarized below.

=== TOPOLOGICAL INFORMATION ===
Several functions allow one to retrieve topological information from a tree; such information can be used, for instance, as a basis to compute distances or proximities between tips.

- `listDD`: lists the direct descendants from each node of a tree.

- `listTips`: lists the tips descending from each node of a tree.

- `.tipToRoot`: finds the set of nodes between a tip and the root of a tree.

- `sp.tips`: finds the shortest path between tips of a tree.

- `treePart`: defines partitions of tips reflecting the topology of a tree. This function can output non-independent dummy vectors, or alternatively an orthonormal basis used by the orthogram procedure.

=== PHYLOGENETIC PROXIMITIES/DISTANCES ===
Several phylogenetic proximities and distances are implemented. Auxiliary function easing the computation of other distances/proximities are also provided:

- `distRoot`: computes different distances of a set of tips to the root.

- `distTips`: computes different pairwise distances in a set of tips.

- `proxTips`: computes different proximities between a set of tips.

=== MEASURES/TESTS OF PHYLOGENETIC AUTOCORRELATION ===
Several procedures allow one to measure, and/or test phylogenetic signal in biological traits:

- `abouheif.moran`: performs Abouheif's test, designed to detect phylogenetic autocorrelation in a quantitative trait. This implementation is not based on original heuristic procedure, but on the exact formulation proposed by Pavoine et al. (2008), showing that the test is in fact a Moran's index test. This implementation further extends the procedure by allowing any measure of phylogenetic proximity (5 are proposed).

- `orthogram`: performs the orthonormal decomposition of variance of a quantitative variable on an orthonormal basis as in Ollier et al. (2005). It also returns the results of five non parametric tests associated to the variance decomposition.

- `moran.idx`: computes Moran's index of autocorrelation given a variable and a matrix of proximities among observations (no test).

=== MODELLING/INVESTIGATION OF PHYLOGENETIC SIGNAL ===
Rather than testing or measuring phylogenetic autocorrelation, these procedures can be used for further investigation of phylogenetic signal. Some, like `me.phylo`, can be used to remove phylogenetic autocorrelation. Others can be used to understand the nature of this autocorrelation (i.e., to ascertain which traits and tips are concerned by phylogenetic non-independence).

- `me.phylo`/`orthobasis.phylo`: these synonymous functions compute Moran's eigenvectors (ME) associated to a tree. These vectors model different observable phylogenetic signals. They can be used as covariables to remove phylogenetic autocorrelation from data.

- `orthogram`: the orthogram mentioned above also provides a description of how biological variability is structured on a phylogeny.

- `ppca`: performs a phylogenetic Principal Component Analysis (pPCA, Jombart et al. 2010). This multivariate method investigates phylogenetic patterns in a set of quantitative traits.

=== GRAPHICS ===
Some plotting functions are proposed, most of them being devoted to representing phylogeny and a quantitative information at the same time.

- `table.phylo4d`: fairly customisable way of representing traits onto the tips of a phylogeny. Several traits can be plotted in a single graphic.

- `bullseye`: an alternative to `table.phylo4d` based on fan-like representation, better for large trees.

- `scatter.ppca`, `screeplot.ppca`, `plot.ppca`: several plots associated to a phylogenetic principal component analysis (see `ppca`).

=== DATASETS ===
Several datasets are also proposed. Some of these datasets replace former version from `ade4`, which are now deprecated. Here is a list of available datasets: `carni19`, `carni70`, `lizards`, `maples`, `mjrochet`, `palm`, `procella`, `tithonia`, and `ungulates`.

To cite adephylo, please use the reference given by `citation("adephylo")`.

Package:	adephylo
Type:	Package
Version:	1.1-7
Date:	2014-11-10
License:	GPL (>=2)

Author(s)

Thibaut Jombart <tjombart@imperial.ac.uk>
with contributions Stephane Dray <stephane.dray@univ-lyon1.fr> and Anders Ellern Bilgrau <abilgrau@math.aau.dk>.
Parts of former code from `ade4` by Daniel Chessel and Sebastien Ollier.

See Also

The `ade4` package for multivariate analysis.

`.tipToRoot` *Low-level auxiliary functions for adephylo*

Description

These hidden functions are utils for adephylo, used by other functions. Regular users can use them as well, but no validity checks are performed for the arguments: speed is here favored over safety. Most of these functions handle trees inheriting phylo4 class.

Usage

```
.tipToRoot(x, tip, root, include.root = FALSE)
```

Arguments

`x`	A valid tree of class phylo4.
`tip`	An integer identifying a tip by its numbers.
`root`	An integer identifying the root of the tree by its number.
`include.root`	a logical stating whether the root must be included as a node of the path from tip to root (TRUE), or not (FALSE, default).

Details

`.tipToRoot` finds the set of nodes between a tip and the root of a tree.

Value

`.tipToRoot`: a vector of named integers identifying nodes.

Author(s)

Thibaut Jombart <`tjombart@imperial.ac.uk`>

Examples

```
if(require(ape) & require(phylobase)){
## make a tree
x <- as(rtree(20),"phylo4")
plot(x,show.node=TRUE)

## .tipToRoot
root <- rootNode(x)
.tipToRoot(x, 1, root)
lapply(1:nTips(x), function(i) .tipToRoot(x, i, root))
}
```

`abouheif.moran` *Abouheif's test based on Moran's I*

Description

The test of Abouheif (1999) is designed to detect phylogenetic autocorrelation in a quantitative trait. Pavoine *et al.* (2008) have shown that this tests is in fact a Moran's I test using a particular phylogenetic proximity between tips (see details). The function `abouheif.moran` performs basically Abouheif's test for several traits at a time, but it can incorporate other phylogenetic proximities as well.

Usage

```
abouheif.moran(
  x,
  W = NULL,
  method = c("oriAbouheif", "patristic", "nNodes", "Abouheif", "sumDD"),
  f = function(x) {
     1/x
 },
  nrepet = 999,
  alter = c("greater", "less", "two-sided")
)
```

Arguments

`x`	a data frame with continuous variables, or a phylo4d object (i.e. containing both a tree, and tip data). In the latter case, `method` argument is used to determine which proximity should be used.
`W`	a *n* by *n* matrix (*n* being the number rows in x) of phylogenetic proximities, as produced by `proxTips`.
`method`	a character string (full or unambiguously abbreviated) specifying the type of proximity to be used. By default, the proximity used is that of the original Abouheif's test. See details in `proxTips` for information about other methods.
`f`	a function to turn a distance into a proximity (see `proxTips`).
`nrepet`	number of random permutations of data for the randomization test
`alter`	a character string specifying the alternative hypothesis, must be one of "greater" (default), "less" or "two-sided"

Details

Note that the original Abouheif's proximity (Abouheif, 1999; Pavoine *et al.* 2008) unifies Moran's I and Geary'c tests (Thioulouse *et al.* 1995).

`abouheif.moran` can be used in two ways:
- providing a data.frame of traits (`x`) and a matrix of phylogenetic proximities (`W`)
- providing a phylo4d object (`x`) and specifying the type of proximity to be used (`method`).

`W` is a squared symmetric matrix whose terms are all positive or null.

`W` is firstly transformed in frequency matrix A by dividing it by the total sum of data matrix :

$$a_{ij} = \frac{W_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij}}$$

The neighbouring weights is defined by the matrix $D = diag(d_1, d_2, \ldots)$ where $d_i = \sum_{j=1}^{n} W_{ij}$. For each vector x of the data frame x, the test is based on the Moran statistic $x^t Ax$ where x is D-centred.

Value

Returns an object of class `krandtest` (randomization tests from ade4), containing one Monte Carlo test for each trait.

Author(s)

Original code from ade4 (gearymoran function) by Sebastien Ollier
Adapted and maintained by Thibaut Jombart <tjombart@imperial.ac.uk>.

References

Thioulouse, J., Chessel, D. and Champely, S. (1995) Multivariate analysis of spatial patterns: a unified approach to local and global structures. *Environmental and Ecological Statistics*, **2**, 1–14.

See Also

- `gearymoran` from the ade4 package
- `Moran.I` from the ape package for the classical Moran's I test.

Examples

```
if(require(ade4)&& require(ape) && require(phylobase)){
## load data
data(ungulates)
tre <- read.tree(text=ungulates$tre)
x <- phylo4d(tre, ungulates$tab)

## Abouheif's tests for each trait
myTests <- abouheif.moran(x)
myTests
plot(myTests)
```

```
## a variant using another proximity
plot(abouheif.moran(x, method="nNodes") )

## Another example

data(maples)
tre <- read.tree(text=maples$tre)
dom <- maples$tab$Dom

## Abouheif's tests for each trait (equivalent to Cmean)
W1 <- proxTips(tre,method="oriAbouheif")
abouheif.moran(dom,W1)

## Equivalence with moran.idx

W2 <- proxTips(tre,method="Abouheif")
abouheif.moran(dom,W2)
moran.idx(dom,W2)
}
```

`bullseye` *Fan-like phylogeny with possible representation of traits on tips*

Description

This function represents a phylogeny as a fan, using circles to provide a legend for distances and optionally colored symbols to represent traits associated to the tips of the tree. This function uses and is compatible with ape's `plot.phylo`.

Usage

```
bullseye(
  phy,
  traits = NULL,
  col.tips.by = NULL,
  col.pal = spectral,
  circ.n = 6,
  circ.bg = transp("royalblue", 0.1),
  circ.unit = NULL,
  legend = TRUE,
  leg.posi = "bottomleft",
  leg.title = "",
  leg.bg = "white",
  traits.inset = 1.1,
  traits.space = 0.05,
  traits.pch = 19,
  traits.cex = 1,
  alpha = 1,
```

```
    axis = TRUE,
    ...
)
```

Arguments

`phy`	a tree in `phylo`, phylo4 or phylo4d format.
`traits`	an optional data.frame of traits.
`col.tips.by`	an optional vector used to define colors for tip labels; if unamed, must be ordered in the same order as `phy$tip.label`.
`col.pal`	a function generating colors according to a given palette; several palettes can be provided as a list, in the case of several traits; the first palette is always reserved for the tip colors; this argument is recycled.
`circ.n`	the number of circles for the distance annotations.
`circ.bg`	the color of the circles.
`circ.unit`	the unit of the circles; if NULL, determined automatically from the data.
`legend`	a logical specifying whether a legend should be plotted; only one legend is displayed, with priority to tip colors first, and then to the first trait.
`leg.posi, leg.title, leg.bg`	position, title and background for the legend.
`traits.inset`	inset for positioning the traits; 1 corresponds to the circle crossing the furthest tip, 0 to the center of the plot.
`traits.space`	a coefficient indicating the spacing between traits.
`traits.pch, traits.cex`	type and size of the symbols used for the traits; recycled if needed.
`alpha`	alpha value to be used for the color transparency, between 0 (invisible) and 1 (plain).
`axis`	a logical indicating whether an axis should be displayed.
`...`	further arguments to be passed to plot methods from `ape`. See `plot.phylo`.

Author(s)

Thibaut Jombart `<tjombart@imperial.ac.uk>`

See Also

`table.phylo4d` for non-radial plots.

The phylo4d class for storing `phylogeny+data`.

`plot.phylo` from the `ape` package.

`dotchart.phylog`.

Examples

```
if(require(ape) && require(phylobase) && require(adegenet)){

data(lizards)
tre <- read.tree(text=lizards$hprA) # make a tree

## basic plots
bullseye(tre)
bullseye(tre, lizards$traits)

## customized
par(mar=c(6,6,6,6))
bullseye(tre, lizards$traits, traits.cex=sqrt(1:7), alpha=.7,
         legend=FALSE, circ.unit=10, circ.bg=transp("black",.1),
         edge.width=2)

}
```

`carni19` *Phylogeny and quantative trait of carnivora*

Description

This data set describes the phylogeny of carnivora as reported by Diniz-Filho et al. (1998). It also gives the body mass of these 19 species.

Format

`carni19` is a list containing the 2 following objects :

tre is a character string giving the phylogenetic tree in Newick format.

bm is a numeric vector which values correspond to the body mass of the 19 species (log scale).

Note

This dataset replaces the former version in ade4.

Source

Diniz-Filho, J. A. F., de Sant'Ana, C.E.R. and Bini, L.M. (1998) An eigenvector method for estimating phylogenetic inertia. *Evolution*, **52**, 1247–1262.

Examples

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

data(carni19)
tre <- read.tree(text=carni19$tre)
x <- phylo4d(tre, data.frame(carni19$bm))
table.phylo4d(x, ratio=.5, center=FALSE)
}

## End(Not run)
```

`carni70` *Phylogeny and quantitative traits of carnivora*

Description

This data set describes the phylogeny of 70 carnivora as reported by Diniz-Filho and Torres (2002). It also gives the geographic range size and body size corresponding to these 70 species.

Format

`carni70` is a list containing the 2 following objects:

tre is a character string giving the phylogenetic tree in Newick format. Branch lengths are expressed as divergence times (millions of years)

tab is a data frame with 70 species and two traits: size (body size (kg)) ; range (geographic range size (km)).

Note

This dataset replaces the former version in ade4.

Source

Diniz-Filho, J. A. F., and N. M. Torres. (2002) Phylogenetic comparative methods and the geographic range size-body size relationship in new world terrestrial carnivora. *Evolutionary Ecology*, **16**, 351–367.

Examples

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

data(carni70)
rownames(carni70$tab) <- gsub("_", ".", rownames(carni70$tab))
```

```
tre <- read.tree(text=carni70$tre)
x <- phylo4d(tre, carni70$tab)
table.phylo4d(x)

par(mar=rep(.1,4))
table.phylo4d(x,cex.lab=.5, show.n=FALSE, ratio=.5)


## transform size in log and test for a phylogenetic signal
size <- log(carni70$tab)[,1]
names(size) <- row.names(carni70$tab)
orthogram(size, tre)

## transform range and test for a phylogenetic signal
yrange <- scale(carni70$tab)[,2]
names(yrange) <- row.names(carni70$tab)
orthogram(yrange, tre)
}

## End(Not run)
```

`dibas` *DIstance-Based Assignment*

Description

These functions are under development. Please do not use them unless asked by the author.

Usage

```
dibas(x, ...)

## S3 method for class 'matrix'
dibas(x, grp, method = c("default", "leaveOneOut"), ...)

## S3 method for class 'vector'
dibas(x, grp, method = c("default", "leaveOneOut"), n.items = NULL, ...)

## S3 method for class 'phylo'
dibas(
  x,
  grp,
  method = c("default", "leaveOneOut"),
  fromRoot = FALSE,
  metric = c("Abouheif", "nNodes", "patristic", "sumDD"),
  n.items = NULL,
  ...
)
```

```
## S3 method for class 'dist'
dibas(x, grp, method = c("default", "leaveOneOut"), ...)
```

Arguments

`x`	a `phylo` object, or a symmetric matrix of pairwise distances of class `matrix` or `dist`.
`...`	further arguments passed to other methods. Can be used to provide arguments to `table.phylo4d` in `plot` method.
`grp`	a `factor` indicating the groups of observations.
`method`	a character string indicating the method to be used for estimating the distribution of pairwise distances within groups. The default method ("default") uses all observations, while the "leaveOneOut" estimates separate group distributions for each individual, leaving this one out in the estimation process.
`n.items`	a vector of integers of the same length as x, stating how many times each items in 'x' should be repeated; used to take into account differences in abundances of the different items (e.g. sequences in multiple copies).
`fromRoot`	a logical indicating if distances from the root, rather than between groups, should be used.
`metric`	a character string matching "nNodes", "patristic", "Abouheif", or "sumDD" indicating the distance measure to be used. See `distTips` for more information. Note that patristic distances should be avoided in presence of one or more highly diverse group because of the 'hand fan' syndrome (see example).

Author(s)

Thibaut Jombart <`tjombart@imperial.ac.uk`>

Examples

```
## Not run:
if(require(ape)){
#### SIMPLE SIMULATED DATA ####
## 50 variables, 2 groups, 30 indiv
dat <- simDatGroups(k=2, p=50, n=c(15,15), mu=c(0,1))
names(dat)

## make a tree
tre <- nj(dist(dat$dat))
plot(tre,type="unr", tip.col=c("blue","red")[as.integer(dat$grp)],
   main="simulated data - tree")

## use dibas method
res <- dibas(tre, dat$grp, metric="nNodes")
res

barplot(t(res$prob), main="group membership probabilities")
```

```
#### NON-PARAMETRIC TEST BASED ON MEAN SUCCESSFUL ASSIGNMENT ####
## use dibas method
distHo <- replicate(100,
   dibas(tre, sample(dat$grp), metric="patristic")$mean.ok)
pval <- mean(res$mean.ok<=c(distHo,res$mean.ok))
pval

hist(c(distHo,res$mean.ok), col="grey",
   main="Mean successful assignement - permuted values")
abline(v=res$mean.ok, col="red")
mtext(side=3, text="Observed value in red")



#### HAND FAN SYNDROME ####
## 50 variables, 2 groups, 30 indiv
dat <- simDatGroups(k=2, p=50, n=c(15,15), mu=c(0,1), sigma=c(2,4))
names(dat)

## make a tree
tre <- nj(dist(dat$dat))
plot(tre,type="unr", tip.col=c("blue","red")[as.integer(dat$grp)],
   main="simulated data - tree")
mtext(side=3, text="hand-fan syndrome")

## use dibas method
res.patri <- dibas(tre, dat$grp, metric="patristic")
res.patri$grp.tab # poor results
plot(table(res.patri$groups), main="Group assignment - dibas patristic")

res <- dibas(tre, dat$grp, metric="nNodes")
res$grp.tab # results OK
plot(table(res$groups), main="Group assignment - dibas nNodes")




#### MORE COMPLEX DATASET ####
if(require(adegenet)){

dat <- simDatGroups(k=5, p=50, n=c(5,10,10,30,60), mu=sample(1:5, 5,
   replace=TRUE), sigma=sample(1:5)/2)
names(dat)

## make a tree
tre <- nj(dist(dat$dat))
plot(tre,type="unr", tip.col=fac2col(dat$grp),main="simulated data - tree")

## use dibas method
res <- dibas(tre, dat$grp, metric="Abouheif")
```

```
res

plot(table(res$groups), main="Group assignment - dibas Abouheif")

}
}

## End(Not run)
```

`distRoot` *Compute the distance of tips to the root*

Description

The function `distRoot` computes the distance of a set of tips to the root. Several distances can be used, defaulting to the sum of branch lengths.

Usage

```
distRoot(
  x,
  tips = "all",
  method = c("patristic", "nNodes", "Abouheif", "sumDD")
)
```

Arguments

`x`	a tree of class `phylo`, phylo4 or phylo4d.
`tips`	A vector of integers identifying tips by their numbers, or a vector of characters identifying tips by their names.
`method`	a character string (full or abbreviated without ambiguity) specifying the method used to compute distances ; possible values are: - `patristic`: patristic distance, i.e. sum of branch lengths - `nNodes`: number of nodes on the path between the nodes - `Abouheif`: Abouheif's distance (see details) - `sumDD`: sum of direct descendants of all nodes on the path (see details)

Details

`Abouheif` distance refers to the phylogenetic distance underlying the test of Abouheif (see references). Let P be the set of all the nodes in the path going from `node1` to `node2`. Let DDP be the number of direct descendants from each node in P. Then, the so-called 'Abouheif' distance is the product of all terms in DDP.

`sumDD` refers to a phylogenetic distance quite similar to that of Abouheif. We consider the same sets P and DDP. But instead of computing the product of all terms in DDP, this distance computes the sum of all terms in DDP.

Value

A numeric vector containing one distance value for each tip.

Author(s)

Thibaut Jombart <`tjombart@imperial.ac.uk`>

References

Pavoine, S.; Ollier, S.; Pontier, D. & Chessel, D. (2008) Testing for phylogenetic signal in life history variable: Abouheif's test revisited. *Theoretical Population Biology*: **73**, 79-91.

See Also

`distTips` which computes the same phylogenetic distances, but between tips.

Examples

```
if(require(ape) & require(phylobase)){
## make a tree
x <- as(rtree(50),"phylo4")
## compute 4 different distances
met <- c("patristic","nNodes","Abouheif","sumDD")
D <- lapply(met, function(e) distRoot(x, method=e) )
names(D) <- met
D <- as.data.frame(D)

## plot these distances along with the tree
temp <- phylo4d(x, D)
table.phylo4d(temp, show.node=FALSE, cex.lab=.6)
}
```

`distTips` *Compute some phylogenetic distance between tips*

Description

The function `distTips` computes a given distance between a set of tips of a phylogeny. A vector of tips is supplied: distances between all possible pairs of these tips are computed. The distances are computed from the shortest path between the tips. Several distances can be used, defaulting to the sum of branch lengths (see argument `method`).

Usage

```
distTips(
  x,
  tips = "all",
  method = c("patristic", "nNodes", "Abouheif", "sumDD"),
  useC = TRUE
)
```

Arguments

`x`	a tree of class `phylo`, phylo4 or phylo4d.
`tips`	A vector of integers identifying tips by their numbers, or a vector of characters identifying tips by their names. Distances will be computed between all possible pairs of tips.
`method`	a character string (full or abbreviated without ambiguity) specifying the method used to compute distances ; possible values are: - `patristic`: patristic distance, i.e. sum of branch lengths - `nNodes`: number of nodes on the path between the nodes - `Abouheif`: Abouheif's distance (see details) - `sumDD`: sum of direct descendants of all nodes on the path (see details)
`useC`	a logical indicating whether computations should be performed using compiled C code (TRUE, default), or using a pure R version (FALSE). C version is several orders of magnitude faster, and R version is kept for backward compatibility.

Details

An option (enabled by default) allows computations to be run using compiled C code, which is much faster than pure R code. In this case, a matrix of all pairwise distances is returned (i.e., `tips` argument is ignored).

`Abouheif` distance refers to the phylogenetic distance underlying the test of Abouheif (see references). Let P be the set of all the nodes in the path going from `node1` to `node2`. Let DDP be the number of direct descendants from each node in P. Then, the so-called 'Abouheif' distance is the product of all terms in DDP.

`sumDD` refers to a phylogenetic distance quite similar to that of Abouheif. We consider the same sets P and DDP. But instead of computing the product of all terms in DDP, this distance computes the sum of all terms in DDP.

Value

An object of class `dist`, containing phylogenetic distances.

Author(s)

Thibaut Jombart <`tjombart@imperial.ac.uk`>

References

Pavoine, S.; Ollier, S.; Pontier, D. & Chessel, D. (2008) Testing for phylogenetic signal in life history variable: Abouheif's test revisited. *Theoretical Population Biology*: **73**, 79-91.

See Also

`distTips` which computes several phylogenetic distances between tips.

Examples

```
if(require(ape) & require(phylobase)){
## make a tree
x <- as(rtree(10),"phylo4")
plot(x, show.node=TRUE)
axisPhylo()
## compute different distances
distTips(x, 1:3)
distTips(x, 1:3, "nNodes")
distTips(x, 1:3, "Abouheif")
distTips(x, 1:3, "sumDD")

## compare C and pure R code outputs
x <- rtree(10)
all.equal(as.matrix(distTips(x)), as.matrix(distTips(x, useC=FALSE)))
all.equal(as.matrix(distTips(x, meth="nNode")),
   as.matrix(distTips(x, meth="nNode", useC=FALSE)))
all.equal(as.matrix(distTips(x, meth="Abou")),
   as.matrix(distTips(x, meth="Abou", useC=FALSE)))
all.equal(as.matrix(distTips(x, meth="sumDD")),
   as.matrix(distTips(x, meth="sumDD", useC=FALSE)))

## compare speed
x <- rtree(50)
tim1 <- system.time(distTips(x, useC=FALSE)) # old pure R version
tim2 <- system.time(distTips(x)) # new version using C
tim1[c(1,3)]/tim2[c(1,3)] # C is about a thousand time faster in this case
}
```

`listDD` *List direct descendants for all nodes of a tree*

Description

The function `listDD` lists the direct descendants from each node of a tree. The tree can be of class `phylo`, phylo4 or phylo4d.

Usage

```
listDD(x, nameBy = c("label", "number"))
```

Arguments

`x`	A tree of class `phylo`, phylo4 or phylo4d.
`nameBy`	a character string indicating whether the returned list must be named by node labels ("label") or by node numbers ("number").

Value

A list whose components are vectors of named nodes (or tips) for a given internal node.

Author(s)

Thibaut Jombart <`tjombart@imperial.ac.uk`>

See Also

`listTips` which lists the tips descending from each node.

`treePart` which defines partitions of tips according to the tree topology.

Examples

```
if(require(ape) & require(phylobase)){
## make a tree
x <- as(rtree(20),"phylo4")
plot(x,show.node=TRUE)
listDD(x)
}
```

`listTips` *List tips descendings from all nodes of a tree*

Description

The function `listTips` lists the tips descending from each node of a tree. The tree can be of class `phylo`, phylo4 or phylo4d.

Usage

```
listTips(x)
```

Arguments

`x`	A tree of class `phylo`, phylo4 or phylo4d.

Value

A list whose components are vectors of named tips for a given node.

Author(s)

Thibaut Jombart `<tjombart@imperial.ac.uk>`

See Also

`listDD` which lists the direct descendants for each node.

`treePart` which defines partitions of tips according to the tree topology.

Examples

```
if(require(ape) & require(phylobase)){
## make a tree
x <- as(rtree(20),"phylo4")
plot(x,show.node=TRUE)
listTips(x)
}
```

`lizards` *Phylogeny and quantitative traits of lizards*

Description

This data set describes the phylogeny of 18 lizards as reported by Bauwens and D\'iaz-Uriarte (1997). It also gives life-history traits corresponding to these 18 species.

Format

`lizards` is a list containing the 3 following objects :

traits is a data frame with 18 species and 8 traits.

hprA is a character string giving the phylogenetic tree (hypothesized phylogenetic relationships based on immunological distances) in Newick format.

hprB is a character string giving the phylogenetic tree (hypothesized phylogenetic relationships based on morphological characteristics) in Newick format.

Details

Variables of `lizards$traits` are the following ones : mean.L (mean length (mm)), matur.L (length at maturity (mm)), max.L (maximum length (mm)), hatch.L (hatchling length (mm)), hatch.m (hatchling mass (g)), clutch.S (Clutch size), age.mat (age at maturity (number of months of activity)), clutch.F (clutch frequency).

Note

This dataset replaces the former version in ade4.

References

Bauwens, D., and D\'iaz-Uriarte, R. (1997) Covariation of life-history traits in lacertid lizards: a comparative study. *American Naturalist*, **149**, 91–111.

See a data description at `http://pbil.univ-lyon1.fr/R/pdf/pps063.pdf` (in French).

Examples

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

## see data
data(lizards)
liz.tr <- read.tree(tex=lizards$hprA) # make a tree
liz <- phylo4d(liz.tr, lizards$traits) # make a phylo4d object
table.phylo4d(liz)

## compute and plot principal components
```

```
if(require(ade4)){
liz.pca1 <- dudi.pca(lizards$traits, cent=TRUE,
   scale=TRUE, scannf=FALSE, nf=2) # PCA of traits
myPC <- phylo4d(liz.tr, liz.pca1$li) # store PC in a phylo4d object
varlab <- paste("Principal \ncomponent", 1:2) # make labels for PCs
table.phylo4d(myPC, ratio=.8, var.lab=varlab) # plot the PCs
add.scatter.eig(liz.pca1$eig,2,1,2,posi="topleft", inset=c(0,.15))
title("Phylogeny and the principal components")

## compute a pPCA ##
## remove size effect
temp <- lapply(liz.pca1$tab, function(e) residuals(lm(e~-1+liz.pca1$li[,1])) )
temp <- data.frame(temp)
row.names(temp) <- tipLabels(liz)

## build corresponding phylo4d object
liz.noSize <- phylo4d(liz.tr, temp)
ppca1 <- ppca(liz.noSize, method="Abouheif", scale=FALSE, scannf=FALSE)
plot(ppca1)

}
}

## End(Not run)
```

`maples` *Phylogeny and quantitative traits of flowers*

Description

This data set describes the phylogeny of 17 flowers as reported by Ackerly and Donoghue (1998). It also gives 31 traits corresponding to these 17 species.

Format

`tithonia` is a list containing the 2 following objects : - tre: a character string giving the phylogenetic tree in Newick format.
- tab: a data frame with 17 species and 31 traits.

Note

This dataset replaces the former version in ade4.

References

Ackerly, D. D. and Donoghue, M.J. (1998) Leaf size, sappling allometry, and Corner's rules: phylogeny and correlated evolution in Maples (Acer). *American Naturalist*, **152**, 767–791.

Examples

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

data(maples)

## see the tree
tre <- read.tree(text=maples$tre)
plot(tre)
axisPhylo()

## look at two variables
dom <- maples$tab$Dom
bif <- maples$tab$Bif
plot(bif,dom,pch = 20)
abline(lm(dom~bif)) # a strong negative correlation ?
summary(lm(dom~bif))
cor.test(bif,dom)

## look at the two variables onto the phylogeny
temp <- phylo4d(tre, data.frame(dom,bif, row.names=tre$tip.label))
table.phylo4d(temp) # correlation is strongly linked to phylogeny

## use ape's PIC (phylogenetic independent contrasts)
pic.bif <- pic(bif, tre)
pic.dom <- pic(dom, tre)
cor.test(pic.bif, pic.dom) # correlation is no longer significant
}

## End(Not run)
```

`mjrochet` *Phylogeny and quantitative traits of teleos fishes*

Description

This data set describes the phylogeny of 49 teleos fishes as reported by Rochet et al. (2000). It also gives life-history traits corresponding to these 49 species.

Format

`mjrochet` is a list containing the 2 following objects :

tre is a character string giving the phylogenetic tree in Newick format.

tab is a data frame with 49 rows and 7 traits.

Details

Variables of `mjrochet$tab` are the following ones : tm (age at maturity (years)), lm (length at maturity (cm)), l05 (length at 5 per cent survival (cm)), t05 (time to 5 per cent survival (years)), fb (slope of the log-log fecundity-length relationship), fm (fecundity the year of maturity), egg (volume of eggs (mm^3)).

Note

This dataset replaces the former version in ade4.

References

Rochet, M. J., Cornillon, P-A., Sabatier, R. and Pontier, D. (2000) Comparative analysis of phylogenic and fishing effects in life history patterns of teleos fishes. *Oikos*, **91**, 255–270.

Examples

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

data(mjrochet)
tre <- read.tree(text=mjrochet$tre) # make a tree
traits <- log((mjrochet$tab))

## build a phylo4d
mjr <- phylo4d(tre, traits)

## see data
table.phylo4d(mjr,cex.lab=.5,show.node=FALSE,symb="square")

## perform Abouheif's test for each trait
mjr.tests <- abouheif.moran(mjr, nrep=499)
mjr.tests

}

## End(Not run)
```

`moran.idx` *Computes Moran's index for a variable*

Description

This simple function computes Moran's index of autocorrelation given a variable and a matrix of proximities among observations.

Usage

```
moran.idx(x, prox, addInfo = FALSE)
```

Arguments

`x`	a numeric vector whose autocorrelation is computed.
`prox`	a matrix of proximities between observations, as computed by the `proxTips`. Off-diagonal terms must be positive or null, while diagonal terms must all equal zero.
`addInfo`	a logical indicating whether supplementary info (null value, minimum and maximum values) should be returned (TRUE) or not (FALSE, default); if computed, these quantities are returned as attributes.

Value

The numeric value of Moran's index.

Author(s)

Thibaut Jombart `<tjombart@imperial.ac.uk>`

References

Moran, P.A.P. (1948) The interpretation of statistical maps. *Journal of the Royal Statistical Society, B* **10**, 243–251.

Moran, P.A.P. (1950) Notes on continuous stochastic phenomena. *Biometrika*, **37**, 17–23.

de Jong, P. and Sprenger, C. and van Veen, F. (1984) On extreme values of Moran's I and Geary's c. *Geographical Analysis*, **16**, 17–24.

See Also

`proxTips` which computes phylogenetic proximities between tips of a phylogeny.

Examples

```
## Not run:
## use maples dataset
data(maples)
tre <- read.tree(text=maples$tre)
dom <- maples$tab$Dom
bif <- maples$tab$Bif


## get a proximity matrix between tips
W <- proxTips(tre, met="Abouheif")

## compute Moran's I for two traits (dom and bif)
moran.idx(dom, W)
```

```
moran.idx(bif, W)
moran.idx(rnorm(nTips(tre)), W)

## build a simple permutation test for 'bif'
sim <- replicate(499, moran.idx(sample(bif), W)) # permutations
sim <- c(moran.idx(bif, W), sim)

pval <- mean(sim>=sim[1]) # right-tail p-value
pval

hist(sim, col="grey", main="Moran's I Monte Carlo test for 'bif'") # plot
mtext("Histogram of permutations and observation (in red)")
abline(v=sim[1], col="red", lwd=3)


## End(Not run)
```

`orthobasis.phylo` *Computes Moran's eigenvectors from a tree or a phylogenetic proximity matrix*

Description

The function `orthobasis.phylo` (also nicknamed `me.phylo`) computes Moran's eigenvectors (ME) associated to a tree. If the tree has 'n' tips, (n-1) vectors will be produced. These vectors form an orthonormal basis: they are centred to mean zero, have unit variance, and are uncorrelated. Each vector models a different pattern of phylogenetic autocorrelation. The first vectors are those with maximum positive autocorrelation, while the last vectors are those with maximum negative autocorrelation. ME can be used, for instance, as regressors to remove phylogenetic autocorrelation from data (see references).

Usage

```
orthobasis.phylo(
  x = NULL,
  prox = NULL,
  method = c("patristic", "nNodes", "oriAbouheif", "Abouheif", "sumDD"),
  f = function(x) {
     1/x
 }
)
```

Arguments

`x`	A tree of class `phylo`, phylo4 or phylo4d.
`prox`	a matrix of phylogenetic proximities as returned by `proxTips`.

`method`	a character string (full or abbreviated without ambiguity) specifying the method used to compute proximities; possible values are: - `patristic`: (inversed sum of) branch lengths - `nNodes`: (inversed) number of nodes on the path between the nodes - `oriAbouheif`: original Abouheif's proximity, with diagonal (see details in `proxTips`) - `Abouheif`: Abouheif's proximity (see details in `proxTips`) - `sumDD`: (inversed) sum of direct descendants of all nodes on the path (see details in `proxTips`).
`f`	a function to change a distance into a proximity.

Details

ME can be obtained from a tree, specifying the phylogenetic proximity to be used. Alternatively, they can be obtained directly from a matrix of phylogenetic proximities as constructed by `proxTips`.

Value

An object of class `orthobasis`. This is a data.frame with Moran's eigenvectors in column, with special attributes:
- attr(...,"values"): Moran's index for each vector - attr(...,"weights"): weights of tips; current implementation uses only uniform weights

Author(s)

Thibaut Jombart `<tjombart@imperial.ac.uk>`

References

Peres-Neto, P. (2006) A unified strategy for estimating and controlling spatial, temporal and phylogenetic autocorrelation in ecological models *Oecologica Brasiliensis* **10**: 105-119.

Dray, S.; Legendre, P. and Peres-Neto, P. (2006) Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbours matrices (PCNM) *Ecological Modelling* **196**: 483-493.

Griffith, D. and Peres-Neto, P. (2006) Spatial modeling in ecology: the flexibility of eigenfunction spatial analyses *Ecology* **87**: 2603-2613.

See Also

- `proxTips` which computes phylogenetic proximities between tips.

- `treePart` which can compute an orthobasis based on the topology of a phylogeny.

Examples

```
if(require(ape) && require(phylobase)){

## SIMPLE EXAMPLE ##
## make a tree
x <- rtree(50)

## compute Moran's eigenvectors
ME <- me.phylo(x, met="Abouheif")
ME

## plot the 10 first vectors
obj <- phylo4d(x, as.data.frame(ME[,1:10]))
table.phylo4d(obj, cex.sym=.7, cex.lab=.7)


## Not run:
## REMOVING PHYLOGENETIC AUTOCORRELATION IN A MODEL ##
## use example in ungulates dataset
data(ungulates)
tre <- read.tree(text=ungulates$tre)
plot(tre)

## look at two traits
afbw <- log(ungulates$tab[,1])
neonatw <- log((ungulates$tab[,2]+ungulates$tab[,3])/2)
names(afbw) <- tre$tip.label
names(neonatw) <- tre$tip.label
plot(afbw, neonatw) # relationship between traits
lm1 <- lm(neonatw~afbw)
abline(lm1)

lm1
resid1 <- residuals(lm1)
orthogram(resid1, tre) # residuals are autocorrelated

## compute Moran's eigenvectors (ME)
myME <- me.phylo(tre, method="Abou")
lm2 <- lm(neonatw ~ myME[,1] + afbw) # use for ME as covariable
resid2 <- residuals(lm2)
orthogram(resid2, tre) # there is no longer phylogenetic autocorrelation

## see the difference
table.phylo4d(phylo4d(tre, cbind.data.frame(resid1, resid2)))

## End(Not run)
}
```

`orthogram` *Orthonormal decomposition of variance*

Description

This function performs the orthonormal decomposition of variance of a quantitative variable on an orthonormal basis. It also returns the results of five non parametric tests associated to the variance decomposition. It thus provides tools (graphical displays and test) for analysing phylogenetic, pattern in one quantitative trait. This implementation replace the (deprecated) version from the `ade4` package.

Usage

```
orthogram(
  x,
  tre = NULL,
  orthobas = NULL,
  prox = NULL,
  nrepet = 999,
  posinega = 0,
  tol = 1e-07,
  cdot = 1.5,
  cfont.main = 1.5,
  lwd = 2,
  nclass,
  high.scores = 0,
  alter = c("greater", "less", "two-sided")
)
```

Arguments

`x`	a numeric vector corresponding to the quantitative variable
`tre`	a tree of class `phylo`, phylo4 or phylo4d.
`orthobas`	an object of class `'orthobasis'`
`prox`	a matrix of phylogenetic proximities as returned by `proxTips`.
`nrepet`	an integer giving the number of permutations
`posinega`	a parameter for the ratio test. If posinega > 0, the function computes the ratio test.
`tol`	a tolerance threshold for orthonormality condition
`cdot`	a character size for points on the cumulative decomposition display
`cfont.main`	a character size for titles
`lwd`	a character size for dash lines
`nclass`	a single number giving the number of cells for the histogram

`high.scores`	a single number giving the number of vectors to return. If > 0, the function returns labels of vectors that explains the larger part of variance.
`alter`	a character string specifying the alternative hypothesis, must be one of "greater" (default), "less" or "two-sided"

Details

Several orthonormal bases can be used. By default, basis is constructed from a partition of tips according to tree topology (as returned by `treePart`); for this, the argument `tre` must be provided. Alternatively, one can provide an orthonormal basis as returned by `orthobasis.phylo`/`me.phylo` (argument `orthobas`), or provide a proximity matrix from which an orthobasis based on Moran's eigenvectors will be constructed (argument `prox`).

The function computes the variance decomposition of a quantitative vector x on an orthonormal basis B. The variable is normalized given the uniform weight to eliminate problem of scales. It plots the squared correlations R^2 between x and vectors of B (variance decomposition) and the cumulated squared correlations SR^2 (cumulative decomposition). The function also provides five non parametric tests to test the existence of autocorrelation. The tests derive from the five following statistics :

- R2Max=$\max(R^2)$. It takes high value when a high part of the variability is explained by one score.
- SkR2k=$\sum_{i=1}^{n-1}(iR_i^2)$. It compares the part of variance explained by internal nodes to the one explained by end nodes.
- Dmax=$\max_{m=1,\dots,n-1}(\sum_{j=1}^{m} R_j^2 - \frac{m}{n-1})$. It examines the accumulation of variance for a sequence of scores.
- SCE=$\sum_{m=1}^{n-1}(\sum_{j=1}^{m} R_j^2 - \frac{m}{n-1})^2$. It examines also the accumulation of variance for a sequence of scores.
- ratio: depends of the parameter posinega. If posinega > 0, the statistic ratio exists and equals $\sum_{i=1}^{posinega} R_i^2$. It compares the part of variance explained by internal nodes to the one explained by end nodes when we can define how many vectors correspond to internal nodes.

Value

If (high.scores = 0), returns an object of class `'krandtest'` (randomization tests) corresponding to the five non parametric tests.

If (high.scores > 0), returns a list containg :

`w`	: an object of class `'krandtest'` (randomization tests)
`scores.order`	: a vector which terms give labels of vectors that explain the larger part of variance

Note

This function replaces the former version from the ade4 package, which is deprecated. Note that if ade4 is not loaded BEFORE adephylo, then the version from ade4 will erase that of adephylo, which will still be available from adephylo::orthogram. In practice, though, this should never happen, since ade4 is loaded as a dependence by adephylo.

Author(s)

Original code: Sebastien Ollier and Daniel Chessel.

Current maintainer: Stephane Dray <stephane.dray@univ-lyon1.fr>

References

Ollier, S., Chessel, D. and Couteron, P. (2005) Orthonormal Transform to Decompose the Variance of a Life-History Trait across a Phylogenetic Tree. *Biometrics*, **62**, 471–477.

See Also

`orthobasis.phylo`

Examples

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

## a phylogenetic example
data(ungulates)
tre <- read.tree(text=ungulates$tre)
plot(tre)

## look at two traits
afbw <- log(ungulates$tab[,1])
neonatw <- log((ungulates$tab[,2]+ungulates$tab[,3])/2)
names(afbw) <- tre$tip.label
names(neonatw) <- tre$tip.label
plot(afbw, neonatw) # relationship between traits
lm1 <- lm(neonatw~afbw)
resid <- residuals(lm1)
abline(lm1)

## plot the two traits and the residuals of lm1
x <- phylo4d(tre, cbind.data.frame(afbw, neonatw, residuals=resid))
table.phylo4d(x) # residuals are surely not independant

## default orthogram for residuals of lm1
orthogram(resid, tre)

## using another orthonormal basis (derived from Abouheif's proximity)
myOrthoBasis <- orthobasis.phylo(tre, method="oriAbouheif") # Abouheif's proximities
orthogram(resid, ortho=myOrthoBasis) # significant phylog. signal

## Abouheif's test
W <- proxTips(tre, method="oriAbouheif") # proximity matrix
abouheif.moran(resid, W)
}
```

```
## End(Not run)
```

`palm` *Phylogenetic and quantitative traits of amazonian palm trees*

Description

This data set describes the phylogeny of 66 amazonian palm trees. It also gives 7 traits corresponding to these 66 species.

Format

`palm` is a list containing the 2 following objects:

tre is a character string giving the phylogenetic tree in Newick format.

traits is a data frame with 66 species (rows) and 7 traits (columns).

Details

Variables of `palm$traits` are the following ones:
- rord: specific richness with five ordered levels
- h: height in meter (squared transform)
- dqual: diameter at breast height in centimeter with five levels `sout : subterranean`, `d1(0, 5 cm)`, `d2(5, 15 cm)`, `d3(15, 30 cm)` and `d4(30, 100 cm)`
- vfruit: fruit volume in mm^3 (logged transform)
- vgrain: seed volume in mm^3 (logged transform)
- aire: spatial distribution area (km^2)
- alti: maximum altitude in meter (logged transform)

Note

This dataset replaces the former version in ade4.

Source

This data set was obtained by Clementine Gimaret-Carpentier.

Examples

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

## load data, make a tree and a phylo4d object
data(palm)
tre <- read.tree(text=palm$tre)
rord <- as.integer(palm$traits$rord) # just use this for plotting purpose
```

```
traits <- data.frame(rord, palm$traits[,-1])
x <- phylo4d(tre, traits)

## plot data
par(mar=rep(.1,4))
table.phylo4d(x, cex.lab=.6)

## test phylogenetic autocorrelation
if(require(ade4)){
prox <- proxTips(x, method="sumDD")
phylAutoTests <- gearymoran(prox, traits[,-3], nrep=499)
plot(phylAutoTests)
}
}

## End(Not run)
```

`ppca` *Phylogenetic principal component analysis*

Description

These functions are designed to perform a phylogenetic principal component analysis (pPCA, Jombart et al. 2010) and to display the results.

Usage

```
ppca(
  x,
  prox = NULL,
  method = c("patristic", "nNodes", "oriAbouheif", "Abouheif", "sumDD"),
  f = function(x) {
     1/x
 },
  center = TRUE,
  scale = TRUE,
  scannf = TRUE,
  nfposi = 1,
  nfnega = 0
)

## S3 method for class 'ppca'
scatter(x, axes = 1:ncol(x$li), useLag = FALSE, ...)

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

```
## S3 method for class 'ppca'
summary(object, ..., printres = TRUE)

## S3 method for class 'ppca'
screeplot(x, ..., main = NULL)

## S3 method for class 'ppca'
plot(x, axes = 1:ncol(x$li), useLag = FALSE, ...)
```

Arguments

`x`	a phylo4d object (for `ppca`) or a ppca object (for other methods).
`prox`	a marix of phylogenetic proximities as returned by `proxTips`. If not provided, this matrix will be constructed using the arguments `method` and `a`.
`method`	a character string (full or abbreviated without ambiguity) specifying the method used to compute proximities; possible values are: - `patristic`: (inversed sum of) branch lengths - `nNodes`: (inversed) number of nodes on the path between the nodes - `oriAbouheif`: original Abouheif's proximity, with diagonal (see details in `proxTips`) - `Abouheif`: Abouheif's proximity (see details in `proxTips`) - `sumDD`: (inversed) sum of direct descendants of all nodes on the path (see details in `proxTips`).
`f`	a function to change a distance into a proximity.
`center`	a logical indicating whether traits should be centred to mean zero (TRUE, default) or not (FALSE).
`scale`	a logical indicating whether traits should be scaled to unit variance (TRUE, default) or not (FALSE).
`scannf`	a logical stating whether eigenvalues should be chosen interactively (TRUE, default) or not (FALSE).
`nfposi`	an integer giving the number of positive eigenvalues retained ('global structures').
`nfnega`	an integer giving the number of negative eigenvalues retained ('local structures').
`axes`	the index of the principal components to be represented.
`useLag`	a logical stating whether the lagged components (`x\$ls`) should be used instead of the components (`x\$li`).
`...`	further arguments passed to other methods. Can be used to provide arguments to `table.phylo4d` in `plot` method.
`object`	a `ppca` object.
`printres`	a logical stating whether results should be printed on the screen (TRUE, default) or not (FALSE).
`main`	a title for the screeplot; if NULL, a default one is used.

Details

`ppca` performs the phylogenetic component analysis. Other functions are:

- `print.ppca`: prints the ppca content

- `summary.ppca`: provides useful information about a ppca object, including the decomposition of eigenvalues of all axes

- `scatter.ppca`: plot principal components using `table.phylo4d`

- `screeplot.ppca`: graphical display of the decomposition of pPCA eigenvalues

- `plot.ppca`: several graphics describing a ppca object

The phylogenetic Principal Component Analysis (pPCA, Jombart et al., 2010) is derived from the spatial Principal Component Analysis (spca, Jombart et al. 2008), implemented in the adegenet package (see `spca`).

pPCA is designed to investigate phylogenetic patterns a set of quantitative traits. The analysis returns principal components maximizing the product of variance of the scores and their phylogenetic autocorrelation (Moran's I), therefore reflecting life histories that are phylogenetically structured. Large positive and large negative eigenvalues correspond to global and local structures.

Value

The class `ppca` are given to lists with the following components:

`eig`	a numeric vector of eigenvalues.
`nfposi`	an integer giving the number of global structures retained.
`nfnega`	an integer giving the number of local structures retained.
`c1`	a data.frame of loadings of traits for each axis.
`li`	a data.frame of coordinates of taxa onto the ppca axes (i.e., principal components).
`ls`	a data.frame of lagged prinpal components; useful to represent of global scores.
`as`	a data.frame giving the coordinates of the axes of an 'ordinary' PCA onto the ppca axes.
`call`	the matched call.
`tre`	a phylogenetic tre with class phylo4.
`prox`	a matrix of phylogenetic proximities.

Other functions have different outputs:

- `scatter.ppca` returns the matched call.

Author(s)

Thibaut Jombart <`tjombart@imperial.ac.uk`>

References

Jombart, T.; Pavoine, S.; Dufour, A. & Pontier, D. (2010, in press) Exploring phylogeny as a source of ecological variation: a methodological approach. doi:10.1016/j.jtbi.2010.03.038

Jombart, T., Devillard, S., Dufour, A.-B. and Pontier, D. (2008) Revealing cryptic phylogenetic patterns in genetic variability by a new multivariate method. *Heredity*, **101**, 92–103.

See Also

The implementation of `spca` in the adegenet package (`adegenet`)

Examples

```
data(lizards)

if(require(ape) && require(phylobase)){

#### ORIGINAL EXAMPLE FROM JOMBART ET AL 2010 ####


## BUILD A TREE AND A PHYLO4D OBJECT
liz.tre <- read.tree(tex=lizards$hprA)
liz.4d <- phylo4d(liz.tre, lizards$traits)
par(mar=rep(.1,4))
table.phylo4d(liz.4d,var.lab=c(names(lizards$traits),
   "ACP 1\n(\"size effect\")"),show.node=FALSE, cex.lab=1.2)


## REMOVE DUPLICATED POPULATIONS
liz.4d <- prune(liz.4d, c(7,14))
table.phylo4d(liz.4d)


## CORRECT LABELS
lab <- c("Pa", "Ph", "Ll", "Lmca", "Lmcy", "Phha", "Pha",
   "Pb", "Pm", "Ae", "Tt", "Ts", "Lviv", "La", "Ls", "Lvir")
tipLabels(liz.4d) <- lab


## REMOVE SIZE EFFECT
dat <- tdata(liz.4d, type="tip")
dat <- log(dat)
newdat <- data.frame(lapply(dat, function(v) residuals(lm(v~dat$mean.L))))
rownames(newdat) <- rownames(dat)
tdata(liz.4d, type="tip") <- newdat[,-1] # replace data in the phylo4d object
```

```
## pPCA
liz.ppca <- ppca(liz.4d,scale=FALSE,scannf=FALSE,nfposi=1,nfnega=1, method="Abouheif")
liz.ppca
tempcol <- rep("grey",7)
tempcol[c(1,7)] <- "black"
barplot(liz.ppca$eig,main='pPCA eigenvalues',cex.main=1.8,col=tempcol)

par(mar=rep(.1,4))
plot(liz.ppca,ratio.tree=.7)


## CONTRIBUTIONS TO PC (LOADINGS) (viewed as dotcharts)
dotchart(liz.ppca$c1[,1],lab=rownames(liz.ppca$c1),main="Global principal
component 1")
abline(v=0,lty=2)

dotchart(liz.ppca$c1[,2],lab=rownames(liz.ppca$c1),main="Local principal
component 1")
abline(v=0,lty=2)


## REPRODUCE FIGURES FROM THE PAPER
obj.ppca <- liz.4d
tdata(obj.ppca, type="tip") <- liz.ppca$li
myLab <- paste(" ",rownames(liz.ppca$li), sep="")

## FIGURE 1
par(mar=c(.1,2.4,2.1,1))
table.phylo4d(obj.ppca, ratio=.7, var.lab=c("1st global PC", "1st local
   PC"), tip.label=myLab,box=FALSE,cex.lab=1.4, cex.sym=1.2, show.node.label=TRUE)
add.scatter.eig(liz.ppca$eig,1,1,1,csub=1.2, posi="topleft", ratio=.23)


## FIGURE 2
s.arrow(liz.ppca$c1,xlim=c(-1,1),clab=1.3,cgrid=1.3)



#### ANOTHER EXAMPLE - INCLUDING NA REPLACEMENT ####
## LOAD THE DATA
data(maples)
tre <- read.tree(text=maples$tre)
x <- phylo4d(tre, maples$tab)
omar <- par("mar")
par(mar=rep(.1,4))
table.phylo4d(x, cex.lab=.5, cex.sym=.6, ratio=.1) # note NAs in last trait ('x')

## FUNCTION TO REPLACE NAS
f1 <- function(vec){
if(any(is.na(vec))){
m <- mean(vec, na.rm=TRUE)
vec[is.na(vec)] <- m
```

```
}
return(vec)
}


## PERFORM THE PPCA
dat <- apply(maples$tab,2,f1) # replace NAs
x.noNA <- phylo4d(tre, as.data.frame(dat))
map.ppca <- ppca(x.noNA, scannf=FALSE, method="Abouheif")
map.ppca


## SOME GRAPHICS
screeplot(map.ppca)
scatter(map.ppca, useLag=TRUE)
plot(map.ppca, useLag=TRUE)


## MOST STRUCTURED TRAITS
a <- map.ppca$c1[,1] # loadings on PC 1
names(a) <- row.names(map.ppca$c1)
highContrib <- a[a< quantile(a,0.1) | a>quantile(a,0.9)]
datSel <- cbind.data.frame(dat[, names(highContrib)], map.ppca$li)
temp <- phylo4d(tre, datSel)
table.phylo4d(temp) # plot of most structured traits


## PHYLOGENETIC AUTOCORRELATION TESTS FOR THESE TRAITS
prox <- proxTips(tre, method="Abouheif")
abouheif.moran(dat[, names(highContrib)], prox)

}
```

`procella` *Phylogeny and quantitative traits of birds*

Description

This data set describes the phylogeny of 19 birds as reported by Bried et al. (2002). It also gives 6 traits corresponding to these 19 species.

Format

`procella` is a list containing the 2 following objects:

tre is a character string giving the phylogenetic tree in Newick format.

traits is a data frame with 19 species and 6 traits

Details

Variables of `procella$traits` are the following ones:
- site.fid: a numeric vector that describes the percentage of site fidelity
- mate.fid: a numeric vector that describes the percentage of mate fidelity
- mass: an integer vector that describes the adult body weight (g)
- ALE: a numeric vector that describes the adult life expectancy (years)
- BF: a numeric vector that describes the breeding frequencies
- col.size: an integer vector that describes the colony size (no nests monitored)

Note

This dataset replaces the former version in ade4.

References

Bried, J., Pontier, D. and Jouventin, P. (2002) Mate fidelity in monogamus birds: a re-examination of the Procellariiformes. *Animal Behaviour*, **65**, 235–246.

See a data description at `http://pbil.univ-lyon1.fr/R/pdf/pps037.pdf` (in French).

Examples

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

## load data, make tree and phylo4d object
data(procella)
tre <- read.tree(text=procella$tre)
x <- phylo4d(tre, procella$traits)
par(mar=rep(.1,4))
table.phylo4d(x,cex.lab=.7)
}

## End(Not run)
```

`proxTips` *Compute some phylogenetic proximities between tips*

Description

The function `proxTips` computes a given proximity between a set of tips of a phylogeny. A vector of tips is supplied: proximities between all possible pairs of these tips are computed. The proximities are computed from the shortest path between the tips.

Usage

```
proxTips(
  x,
  tips = "all",
  method = c("patristic", "nNodes", "oriAbouheif", "Abouheif", "sumDD"),
  f = function(x) {
     1/x
 },
  normalize = c("row", "col", "none"),
  symmetric = TRUE,
  useC = TRUE
)
```

Arguments

`x`	a tree of class `phylo`, phylo4 or phylo4d.
`tips`	A vector of integers identifying tips by their numbers, or a vector of characters identifying tips by their names. Distances will be computed between all possible pairs of tips.
`method`	a character string (full or abbreviated without ambiguity) specifying the method used to compute proximities; possible values are: - `patristic`: (inversed sum of) branch length - `nNodes`: (inversed) number of nodes on the path between the nodes - `oriAbouheif`: original Abouheif's proximity, with diagonal (see details) - `Abouheif`: Abouheif's proximity without diagonal (see details) - `sumDD`: (inversed) sum of direct descendants of all nodes on the path (see details)
`f`	a function to change a distance into a proximity.
`normalize`	a character string specifying whether the matrix must be normalized by row (`row`), column (`col`), or not (`none`). Normalization amounts to dividing each row (or column) so that the marginal sum is 1. Hence, default is matrix with each row summing to 1.
`symmetric`	a logical stating whether M must be coerced to be symmetric (TRUE, default) or not. This is achieved by taking (denoting N the matrix of proximities before re-symmetrization): $$M = 0.5 * (N + N^T)$$ Note that $x^T N y = x^T M y$, but the latter has the advantage of using a bilinear symmetric form (more appropriate for optimization purposes).
`useC`	a logical indicating whether computations of distances (before transformation into proximities) should be performed using compiled C code (TRUE, default), or using a pure R version (FALSE). C version is several orders of magnitude faster, and R version is kept for backward compatibility.

Details

Proximities are computed as the inverse (to the power `a`) of a phylogenetic distance (computed by `distTips`. Denoting $D = [d_{ij}]$ a matrix of phylogenetic distances, the proximity matrix $M = [m_{ij}]$ is computed as:

$$m_{ij} = \frac{1}{d_{ij}^a} \forall i \neq j$$

and

$$m_{ii} = 0$$

Several distances can be used, defaulting to the sum of branch lengths (see argument `method`). Proximities are not true similarity measures, since the proximity of a tip with itself is always set to zero.

The obtained matrix of phylogenetic proximities (M) defines a bilinear symmetric form when M is symmetric (default):

$$f(x, y) = x^T M y$$

In general, M is not a metric because it is not positive-definite. Such a matrice can be used to measure phylogenetic autocorrelation (using Moran's index):

$$I(x) = \frac{x^T M x}{var(x)}$$

or to compute lag vectors (Mx) used in autoregressive models, like:

$$x = Mx + ... + e$$

where '...' is the non-autoregressive part of the model, and 'e' are residuals.

`Abouheif` proximity refers to the phylogenetic proximity underlying the test of Abouheif (see references). Let P be the set of all the nodes in the path going from `node1` to `node2`. Let DDP be the number of direct descendants from each node in P. Then, the so-called 'Abouheif' distance is the inverse of the product of all terms in DDP. `oriAbouheif` returns a matrix with non-null diagonal elements, as formulated in Pavoine *et al.* (2008). This matrix is bistochastic (all marginal sums equal 1), but this bilinear symmetric form does not give rise to a Moran's index, since it requires a null diagonal. `Abouheif` contains Abouheif's proximities but has a null diagonal, giving rise to a Moran's index.

`sumDD` refers to a phylogenetic proximity quite similar to that of Abouheif. We consider the same sets P and DDP. But instead of taking the inverse of the product of all terms in DDP, this proximity computes the inverse of the sum of all terms in DDP. This matrix was denoted 'M' in Pavoine *et al.* (2008), who reported that it is related to May's index (May, 1990).

Value

A matrix of phylogenetic proximities.

Author(s)

Thibaut Jombart `<tjombart@imperial.ac.uk>`

References

== About Moran's index with various proximities ==
Pavoine, S.; Ollier, S.; Pontier, D.; Chessel, D. (2008) Testing for phylogenetic signal in life history variable: Abouheif's test revisited. *Theoretical Population Biology*: **73**, 79-91.

== About regression on phylogenetic lag vector ==
Cheverud, J. M.; Dow, M. M.; Leutenegger, W. (1985) The quantitative assessment of phylogentic constaints in comparative analyses: sexual dimorphism in body weights among primates. *Evolution* **39**, 1335-1351.

Cheverud, J. M.; Dow, M. M. (1985) An autocorrelation analysis of genetic variation due to lineal fission in social groups of Rhesus macaques. *American Journal of Phyisical Anthropology* **67**, 113-121.

== Abouheif's original paper ==
Abouheif, E. (1999) A method for testing the assumption of phylogenetic independence in comparative data. *Evolutionary Ecology Research*, **1**, 895-909.

== May's index ==
May, R.M. (1990) Taxonomy as destiny. *Nature* **347**, 129-130.

See Also

`distTips` which computes several phylogenetic distances between tips.

Examples

```
if(require(ape) & require(phylobase)){
## make a tree
x <- as(rtree(10),"phylo4")
plot(x, show.node=TRUE)
axisPhylo()
## compute different distances
proxTips(x, 1:5)
proxTips(x, 1:5, "nNodes")
proxTips(x, 1:5, "Abouheif")
proxTips(x, , "sumDD")

## see what one proximity looks like
M <- proxTips(x)
obj <- phylo4d(x,as.data.frame(M))
table.phylo4d(obj,symbol="sq")
}
```

`sp.tips` *Find the shortest path between tips of a tree*

Description

The function `sp.tips` finds the shortest path between tips of a tree, identified as `tip1` and `tip2`. This function applies to trees with the class `phylo`, phylo4 or phylo4d. Several tips can be provided at a time.

Usage

```
sp.tips(x, tip1, tip2, useTipNames = FALSE, quiet = FALSE, include.mrca = TRUE)
```

Arguments

`x`	A tree of class `phylo`, phylo4 or phylo4d.
`tip1`	A vector of integers identifying tips by their numbers, or a vector of characters identifying tips by their names. Recycled if needed.
`tip2`	A vector of integers identifying tips by their numbers, or a vector of characters identifying tips by their names. Recycled if needed.
`useTipNames`	a logical stating whether the output must be named using tip names in all cases (TRUE), or not (FALSE). If not, names of `tip1` and `tip2` will be used.
`quiet`	a logical stating whether a warning must be issued when tip1==tip2, or not (see details).
`include.mrca`	a logical stating whether the most recent common ancestor shall be included in the returned path (TRUE, default) or not (FALSE).

Details

The function checks if there are cases where tip1 and tip2 are the same. These cases are deleted when detected, issuing a warning (unless `quiet` is set to TRUE).

Value

A list whose components are vectors of named nodes forming the shortest path between a couple of tips.

Author(s)

Thibaut Jombart `<tjombart@imperial.ac.uk>`

See Also

`shortestPath` which does the same thing as `sp.tips`, for any node (internal or tip), but much more slowly.

Examples

```
## Not run:
if(require(ape) & require(phylobase)){
## make a tree
x <- as(rtree(20),"phylo4")
plot(x,show.node=TRUE)
## get shortest path between tip 1 and all other tips.
sp.tips(x, "t1", "t2")
sp.tips(x, 1, 2:20, TRUE)
}

## End(Not run)
```

`table.phylo4d` *Graphical display of phylogeny and traits*

Description

This function represents traits onto the tips of a phylogeny. Plotted objects must be valid phylo4d objects (implemented by the `phylobase` package). Current version allows plotting of a tree and one or more quantitative traits (possibly containing missing data, represented by an 'x').

Usage

```
table.phylo4d(
  x,
  treetype = c("phylogram", "cladogram"),
  symbol = c("circles", "squares", "colors"),
  repVar = 1:ncol(tdata(x, type = "tip")),
  center = TRUE,
  scale = TRUE,
  legend = TRUE,
  grid = TRUE,
  box = TRUE,
  show.tip.label = TRUE,
  show.node.label = TRUE,
  show.var.label = TRUE,
  ratio.tree = 1/3,
  font = 3,
  tip.label = tipLabels(x),
  var.label = colnames(tdata(x, type = "tip")),
  cex.symbol = 1,
  cex.label = 1,
  cex.legend = 1,
```

```
    pch = 20,
    col = heat.colors(100),
    coord.legend = NULL,
    ...
)
```

Arguments

`x`	a phylo4d object
`treetype`	the type of tree to be plotted ("phylogram" or "cladogram")
`symbol`	the type of symbol used to represent data ("circles", "squares", or "colors")
`repVar`	the numerical index of variables to be plotted
`center`	a logical stating whether variables should be centred (TRUE, default) or not (FALSE)
`scale`	a logical stating whether variables should be scaled (TRUE, default) or not (FALSE)
`legend`	a logical stating whether a legend should be added to the plot (TRUE) or not (FALSE, default)
`grid`	a logical stating whether a grid should be added to the plot (TRUE, default) or not (FALSE)
`box`	a logical stating whether a box should be added around the plot (TRUE, default) or not (FALSE)
`show.tip.label`	a logical stating whether tip labels should be printed (TRUE, default) or not (FALSE)
`show.node.label`	a logical stating whether node labels should be printed (TRUE, default) or not (FALSE)
`show.var.label`	a logical stating whether labels of variables should be printed (TRUE, default) or not (FALSE)
`ratio.tree`	the proportion of width of the figure occupied by the tree
`font`	an integer specifying the type of font for the labels: 1 (plain text), 2 (bold), 3 (italic, default), or 4 (bold italic).
`tip.label`	a character vector giving the tip labels
`var.label`	a character vector giving the labels of variables
`cex.symbol`	a numeric giving the factor scaling the symbols
`cex.label`	a numeric giving the factor scaling all labels
`cex.legend`	a numeric giving the factor scaling the legend
`pch`	is `symbol` is set to 'colors', a number indicating the type of point to be plotted (see ?points)
`col`	is `symbol` is set to 'colors', a vector of colors to be used to represent the data
`coord.legend`	an optional list with two components 'x' and 'y' indicating the lower-left position of the legend. Can be set to `locator(1) to position the legend interactively.`
`...`	further arguments to be passed to plot methods from `ape`. See `plot.phylo`.

Details

The plot of phylogenies is performed by a call to `plot.phylo` from the ape package. Hence, many of the arguments of `plot.phylo` can be passed to `table.phylo4d`, through the ... argument, but their names must be complete.

For large trees, consider using `bullseye`.

The function `table.phylo4d` is based on former plot method for phylo4d objects from the `phylobase` package. It replaces the deprecated `ade4` functions `symbols.phylog` and `table.phylog`.

Author(s)

Thibaut Jombart `<tjombart@imperial.ac.uk>`

See Also

The phylo4d class for storing `phylogeny+data`.

For large trees, consider using `bullseye`.

`plot.phylo` from the ape package.

An alternative (deprecated) representation is available from `dotchart.phylog`.

Examples

```
if(require(ape) & require(phylobase) & require(ade4)){

## simulated data
tr <- rtree(20)
dat <- data.frame(a = rnorm(20), b = scale(1:20), c=runif(20,-2,2) )
dat[3:6, 2] <- NA # introduce some NAs
obj <- phylo4d(tr, dat) # build a phylo4d object
table.phylo4d(obj) # default scatterplot
table.phylo4d(obj,cex.leg=.6, use.edge.length=FALSE) # customized
table.phylo4d(obj,treetype="clad", show.node=FALSE, cex.leg=.6,
use.edge.length=FALSE, edge.color="blue", edge.width=3) # more customized


## teleost fishes data
data(mjrochet)
temp <- read.tree(text=mjrochet$tre) # make a tree
mjr <- phylo4d(x=temp,tip.data=mjrochet$tab) # male a phylo4d object
table.phylo4d(mjr,cex.lab=.5,show.node=FALSE,symb="square")


## lizards data
data(lizards)
liz.tr <- read.tree(tex=lizards$hprA) # make a tree
liz <- phylo4d(liz.tr, lizards$traits) # make a phylo4d object
table.phylo4d(liz)
```

```
## plotting principal components
liz.pca1 <- dudi.pca(lizards$traits, scannf=FALSE, nf=2) # PCA of traits
myPC <- phylo4d(liz.tr, liz.pca1$li) # store PC in a phylo4d object
varlab <- paste("Principal \ncomponent", 1:2) # make labels for PCs
table.phylo4d(myPC, ratio=.8, var.lab=varlab) # plot the PCs
add.scatter.eig(liz.pca1$eig,2,1,2,posi="topleft", inset=c(0,.15))
title("Phylogeny and the principal components")

}
```

`tithonia` *Phylogeny and quantitative traits of flowers*

Description

This data set describes the phylogeny of 11 flowers as reported by Morales (2000). It also gives morphologic and demographic traits corresponding to these 11 species.

Format

`tithonia` is a list containing the 2 following objects :

tre is a character string giving the phylogenetic tree in Newick format.

tab is a data frame with 11 species and 14 traits (6 morphologic traits and 8 demographic).

Details

Variables of `tithonia$tab` are the following ones :
morho1: is a numeric vector that describes the seed size (mm)
morho2: is a numeric vector that describes the flower size (mm)
morho3: is a numeric vector that describes the female leaf size (cm)
morho4: is a numeric vector that describes the head size (mm)
morho5: is a integer vector that describes the number of flowers per head
morho6: is a integer vector that describes the number of seeds per head
demo7: is a numeric vector that describes the seedling height (cm)
demo8: is a numeric vector that describes the growth rate (cm/day)
demo9: is a numeric vector that describes the germination time
demo10: is a numeric vector that describes the establishment (per cent)
demo11: is a numeric vector that describes the viability (per cent)
demo12: is a numeric vector that describes the germination (per cent)
demo13: is a integer vector that describes the resource allocation
demo14: is a numeric vector that describes the adult height (m)

Note

This dataset replaces the former version in ade4.

Source

Data were obtained from Morales, E. (2000) Estimating phylogenetic inertia in Tithonia (Asteraceae) : a comparative approach. *Evolution*, **54**, 2, 475–484.

Examples

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

data(tithonia)
tre <- read.tree(text=tithonia$tre)
traits <- log(tithonia$tab + 1)
rownames(traits) <- gsub("_", ".", rownames(traits))

## build a phylo4d object
x <- phylo4d(tre, traits)
par(mar=rep(.1,4))
table.phylo4d(x)

}

## End(Not run)
```

`treePart` *Define partitions of tips according from a tree*

Description

The function `treePart` defines partitions of tips reflecting the topology of a tree. There are two possible outputs (handled by the argument `result`):
- `basis` mode: each node but the root is translated into a dummy vector having one value for each tip: this value is '1' if the tip descends from this node, and '0' otherwise.
- `orthobasis`: in this mode, an orthonormal basis is derived from the basis previously mentionned. This orthobasis was proposed in the orthogram (Ollier *et al.* 2006).

Usage

```
treePart(x, result = c("dummy", "orthobasis"))
```

Arguments

`x`	a tree of class `phylo`, phylo4 or phylo4d.
`result`	a character string specifying the type of result: either a basis of dummy vectors (`dummy`), or an orthobasis derived from these dummy vectors (`orthobasis`).

Details

Orthobasis produced by this function are identical to those stored in the Bscores component of deprecated phylog objects, from the ade4 package.

Value

A matrix of numeric vectors (in columns) having one value for each tip (rows).

Author(s)

Thibaut Jombart <`tjombart@imperial.ac.uk`>

References

Ollier, S., Chessel, D. and Couteron, P. (2005) Orthonormal Transform to Decompose the Variance of a Life-History Trait across a Phylogenetic Tree. *Biometrics*, **62**, 471–477.

See Also

- `listDD` which is called by `treePart`.
- `orthogram`, which uses by default the orthobasis produced by `treePart`.

Examples

```
## Not run:

if(require(ape) & require(phylobase)){
## make a tree
x <- as(rtree(10),"phylo4")
partition <- treePart(x)
partition

## plot the dummy vectors with the tree
temp <- phylo4d(x, partition)
table.phylo4d(temp, cent=FALSE, scale=FALSE)
}

## End(Not run)
```

`ungulates` *Phylogeny and quantitative traits of ungulates.*

Description

This data set describes the phylogeny of 18 ungulates as reported by Pelabon et al. (1995). It also gives 4 traits corresponding to these 18 species.

Format

`fission` is a list containing the 2 following objects :

tre is a character string giving the phylogenetic tree in Newick format.

tab is a data frame with 18 species and 4 traits

Details

Variables of `ungulates$tab` are the following ones :

- afbw: is a numeric vector that describes the adult female body weight (g)
- mnw: is a numeric vector that describes the male neonatal weight (g)
- fnw: is a numeric vector that describes the female neonatal weight (g)
- ls: is a numeric vector that describes the litter size

Note

This dataset replaces the former version in ade4.

Source

Data were obtained from Pelabon, C., Gaillard, J.M., Loison, A. and Portier, A. (1995) Is sex-biased maternal care limited by total maternal expenditure in polygynous ungulates? *Behavioral Ecology and Sociobiology*, **37**, 311–319.

Examples

```
## Not run:
if(require(ape) && require(phylobase)){
## load data
data(ungulates)
tre <- read.tree(text=ungulates$tre)
plot(tre)

## look at two traits
afbw <- log(ungulates$tab[,1])
neonatw <- log((ungulates$tab[,2]+ungulates$tab[,3])/2)
names(afbw) <- tre$tip.label
names(neonatw) <- tre$tip.label
plot(afbw, neonatw) # relationship between traits
lm1 <- lm(neonatw~afbw)
abline(lm1)
x <- phylo4d(tre, cbind.data.frame(afbw, neonatw)) # traits on the phylogeny

## test phylogenetic inertia in residuals
orthogram(residuals(lm1), x)
}
```

```
## End(Not run)
```

Index