Package ‘daee’

July 16, 2018

Type Package

Title Data Analysis for Ecology and Evolution

Version 0.1.6

Date 2018-07-16

Author Vanderlei Julio Debastiani

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

Depends PCPS (>= 1.0.4), ape

Imports bbmle, stats, graphics, utils, phytools, phylobase,

Description Set of functions for analysis of ecological and evolutionary data. These functions calculate Phylogenetic Eigenvector Regression (PVR) with eigenvector selection, extract residuals from Mantel test, generate Linear Models (LM) with all possible combinations of variables included in the full model, show information about a label in a phylogenetic tree, makes node labels, add species in a phylogenetic tree, organize a list in a single matrix, compact a phylogenetic tree, define graphical parameters to plot phylogenetic tree, plot phylogenetic tree with nodes collapsed and extract details of information criterion. The functions are available as an R package only to facilitate installation.

License GPL-2

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

daee-package *Data Analysis for Ecology and Evolution*

Description

Set of functions for analyzing ecological and evolutionary data. The functions are available as an R package only to facilitate installation.

Author(s)

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

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

add.taxa.phylo *Add species in a phylogenetic tree*

Description

Function add species in a phylogenetic tree. Three method are available, see details. This function is based in the functions bind.tree and bind.tip.

Usage

```
add.taxa.phylo(tree, taxa, m = 0, prefix = "NeWNodEPhylO")

add.taxa.phylo.phylomatic(tree, taxa, renove.artificial.tip = TRUE, m = 0,
  prefix = "NeWNodEPhylomatiC")

add.taxa.phylo.random(tree, taxa, random.order = TRUE, m = 0,
  prefix = "NeWNodERandoM", method = "length")
```

Arguments

tree	as an object of class "phylo".
taxa	Taxa structure to add species. See details.
m	One number to starting sequence (default m = 0).
prefix	The prefix to new nodes labels (default "NeWNodEPhylO", "NeWNodERandoM" or "NeWNodEPhylomatiC").
renove.artificial.tip	Logical argument (TRUE or FALSE) to specify if artificial tips are removed or not, only to add.taxa.phylo.phylomatic function (default renove.artificial.tip = TRUE).
random.order	Logical argument (TRUE or FALSE) to specify if sequence order in input is keeped (FALSE) or random (TRUE) (default random.order = TRUE).
method	The method to add the new tip, partial match to "equal" or "length". See details (Default method = "length").

Details

The function add.taxa.phylo.random add new tips at random. Taxa object is matrix with two columns. The first column is the node label to new tip be anchored. The tip will be anchored random inside this clade. The most recent common ancestors known can be used to define the point of anchorage. The second column is the label to new tip. Two methods are available, the method "length" add the tip with probability equivalent to edge length, long edge have more probability than short edge. The method "equal" each edge has equal probability to receive the new tip.

The function add.taxa.phylo allows to add the new tips according to more related species or node provided. The taxa object is matrix with three columns. The first column is the node label or tip label to new tip be anchored. The second column is the label to new tip. The third column is the terminal edge length for the added tips, can be NA. If anchored label is a node the new tip is inserted as polytomy and edge length is not used, the length is computed to keep the tree ultrametric. If anchored is a tip label and edge length was provided the new tip is inserted with length provide, and if edge length was NA the length is computed simply by split current edge length by two.

The function add.taxa.phylo.phylomatic allows to add the new tips according structure provided, this method is similar to software phylomatic. The taxa object is matrix with two columns. The first column is the node label or tip label to new tip be anchored. The second column is the structure to new tip. This structure is composed by labels separated by common slash (/), for example family/genus/sp. The anchored label is removed and the edge length is divided by the number of levels (number of slash) within each clade. Artificial tips (named TReMoVtIP) are inserted to mark the steps of splits, they can be removed.

Value

A list with:

`call` The call arguments.

`m.start` The start m value.

`m.current` The new m value, the difference between m.start and m.current is the number of node names created.

`prefix` Prefix used.

`new.tips` The new tips.

`tree` The tree, class "phylo".

Author(s)

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

See Also

bind.tree, bind.tip, add.random, add.tip.random

Examples

```
tree <- read.tree(text="(C:32,(B:16,A:16)N2:16)N1;")
tree <- compute.brlen(tree)
plot(tree, show.node=TRUE)

# add.taxa.phylo without edge length
taxa <- matrix(c("A", "C", "N2", "N2", "A1","C1", "in_N2.1", "in_N2.2",rep(NA,4)),4,3)
taxa
plot(add.taxa.phylo(tree, taxa)$tree, show.node.label = TRUE)
```

```
# add.taxa.phylo taxa with edge length
taxa <- matrix(c("A", "C", "A1","C1", 0.01, 0.9), 2,3)
taxa
plot(add.taxa.phylo(tree, taxa)$tree, show.node.label = TRUE)

# add.taxa.phylo.random
taxa <- matrix(c("N1","N2","Anywhere","Only_in_N2"), 2, 2)
taxa
plot(add.taxa.phylo.random(tree, taxa)$tree, show.node.label = TRUE)
plot(add.taxa.phylo.random(tree, taxa)$tree, show.node.label = TRUE)

# add.taxa.phylo.phylomatic
taxa <- matrix(c("B","B","D/D1/DD1","D/D2/DD2/DDD1"),2,2)
taxa
plot(add.taxa.phylo.phylomatic(tree, taxa, renove.artificial.tip = TRUE)$tree,
 show.node.label = TRUE)
plot(add.taxa.phylo.phylomatic(tree, taxa, renove.artificial.tip = FALSE)$tree,
 show.node.label = TRUE)
```

add.tip.random *Add tips at random to the tree*

Description

Function to add new tip at random to a tree with branch length. This function is a small change in the function add.random, see details.

Usage

```
add.tip.random(tree, tip, method = "length")
```

Arguments

tree	phylogeny as an object of class "phylo".
tip	tip name for the added.
method	The method to add the new tip, partial match to "equal" or "length". See details (Default method = "length").

Details

This function is a small change in the function add.random allowing add only one tip at random only in ultrametric phylogenetic tree. Two methods are available. The method "length" add the tip with probability equivalent to edge length, long edge have more probability than short edge. The method "equal" each edge has equal probability to receive the new tip.

Value

The phylogeny as an object of class "phylo" with new tip.

Author(s)

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

See Also

add.random add.taxa.phylo

Examples

```
set.seed(10)
tree <- rtree(5)
tree <- compute.brlen(tree)
tree <- makeNodeLabel(tree)
plot.phylo(tree, show.node.label = TRUE)
res <- add.tip.random(tree, "NEW", method = "length")
res
plot.phylo(res, show.node.label = TRUE)
```

---

allmodels *Linear Models with all possible combinations of all variables*

Description

The function generates Linear Models (LM) with all possible combinations of all variables included in the full model. Each model is a GLM of family gaussian with the response variable modeled by one or more independent variables (Y= a+X1b1+e, Y= a+X1b1+X2b2+e, and so on).

Usage

```
allmodels(response, variables, subset, type = "AICc",
  only_intercept = FALSE, importance = TRUE, ...)

## S3 method for class 'allmodels'
print(x, n = 10, ...)

## S3 method for class 'allmodels'
summary(object, ...)
```

Arguments

`response`	A data frame containing the response variable.
`variables`	A data frame containing all the independent variables for the models.
`subset`	Maximum number of independent variables to be considered in each model.
`type`	Information criterion to be used "AIC","BIC","AICc","qAIC" and "qAICc" (Default type = "AICc").
`only_intercept`	Logical argument (TRUE or FALSE) to specify if a model containing only the intercept should be included (Default only_intercept = FALSE).
`importance`	Logical argument (TRUE or FALSE) to specify if the relative importance of variables should be calculated. (Default importance = TRUE).
`...`	Other parameters for the respective functions. In allmodels function the parameters are for the glm function.
`x`	An object of class allmodels.
`n`	Number of model for print.
`object`	An object of class allmodels.

Value

Envir_class	The class of each variable.
N_models	The number of models.
IC	A data frame containing the information criterion, the number of parameters, difference in IC from minimum - IC model and the weights IC. The same that function ICtab.
Models	A list with all models. Each model is the class glm.
ImpValues	Relative importance of each variable.

Note

If the model with only the intercept is include, this will be the last model.

Author(s)

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

See Also

glm ICtab

Examples

```
#require(vegan)
#data(mite)
#data(mite.pcnm)
#response<-rowSums(mite)
#Res<-allmodels(response,mite.pcnm,subset=3)
#Res
#summary(Res)
#Res$Models$Model_1157
```

compact.tree *Compact a phylogenetic tree*

Description

Funtion to compact a phylogenetic tree. See details.

Usage

```
compact.tree(tree, nodes)
```

Arguments

tree	phylogeny as an object of class "phylo".
nodes	a vector with node label to compact the edges. Only two tips are keeps, most short a most long within each nodes

Details

This function compacts the node of a phylogenetic tree. Inside each node the function removes tips using the function `drop.tip`. Only two tips are keeped: the longhest and shortest edge length within each node. This can be useful only for the function `plotcollapse.phylo`.

Value

The phylogeny as an object of class "phylo" with nodes compacted.

Author(s)

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

See Also

`plotcollapse.phylo` `graphical.node.patterns`

Examples

```
set.seed(10)
tree <- rtree(15)
tree <- makeNodeLabel(tree)
plot.phylo(tree, show.node.label = TRUE)
nodes <- c("Node2", "Node9")
res <- compact.tree(tree, nodes)
res
plot.phylo(res, show.node.label = TRUE)
```

---

graphical.node.patterns

Define graphical parameters to plot phylogenetic tree.

Description

Function to define graphical parameters to each edge when draw a phylogenetic tree. See details.

Usage

```
graphical.node.patterns(tree, nodes, basicpattern, nodespatterns,
  include.node = TRUE, force.order = TRUE)
```

Arguments

`tree`	phylogeny as an object of class "phylo".
`nodes`	a vector with node label to search the edges to change the basic graphical parameter.
`basicpattern`	the basic pattern for graphical parameter. This is apply for all edges.
`nodespatterns`	a vector with new graphical parameter for each node label. This change the basic graphical parameter in each node.
`include.node`	logical argument (TRUE or FALSE) to specify if edge of each node is include in change (default include.node = TRUE).
`force.order`	logical argument (TRUE or FALSE) to specify if force the search as according to edges (default force.order = TRUE).

Details

This function can be used to especify diferent graphical parameters (e.g. color, width and line types) for specific nodes when draw a phylogenetic tree. First, the basicpattern argument is defined for all edges of phylogenetic tree and when the basic pattern is changed in all edge within of each node, following the nodespatterns specify. The argument force.order specify if changes following order in nodes arguments step by step (this case, some changes may have no effect) or change are done from root to tips.

Value

A vector with the new graphical parameters for each edge in phylogenetic tree.

Author(s)

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

See Also

`plot.phylo` `plotcollapse.phylo`

Examples

```
set.seed(10)
tree <- rtree(15)
tree <- makeNodeLabel(tree)
plot.phylo(tree, show.node.label = TRUE)

edge.col <- graphical.node.patterns(tree, nodes = c("Node2", "Node8"),
   basicpattern = "black", nodespatterns = c("red","blue"))
edge.col # Color vector for each edge
plot.phylo(tree, show.node.label = TRUE, edge.color = edge.col)

edge.width <- graphical.node.patterns(tree, nodes = c("Node11","Node3"),
   basicpattern = 1, nodespatterns = 5, include.node = FALSE)
edge.width # width vector for each edge
plot.phylo(tree, show.node.label = TRUE, edge.width = edge.width)

tree <- rtree(250)
tree <- makeNodeLabel(tree)
plot(tree, show.tip.label = FALSE)
edge.col <- graphical.node.patterns(tree, nodes = tree$node.label,
   basicpattern = "black", nodespatterns = rainbow(length(tree$node.label)))
edge.col
plot.phylo(tree, edge.color = edge.col, show.tip.label = FALSE)
```

ICdetails *Extract details of information criterion*

Description

The function extract details of information criterion (IC, e.g. AIC, AICc) allowing calculate difference in IC from minimum - IC model (delta), the Akaike weights and the sum of Akaike weights by each explanatory variable (relative importance of each variable).

Usage

```
ICdetails(x, variables = NULL, order = TRUE)
```

Arguments

`x` A data frame or matrix containing the information criterion for each model. The information criterion in the first column and the models in rows. See examples.

`variables` A data frame or matrix containing the indication of all independent variables used in each model. The models in row and variables in columns. This must be binary, where 1 indicate presence of variable in respective model and 0 the ausence of the variable in respective model. See examples (default variables = NULL).

`order` Logical argument (TRUE or FALSE) to specify if informtion criterion is sorted (Default order = TRUE).

Value

A list with:

`call` The arguments used.

`IC` A data frame containing the information criterion, difference in IC from minimum - IC model (delta) and the weights IC.

`IValue` Relative importance of each variable.

Author(s)

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

References

Anderson, D.R. 2008. Model Based Inference in the Life Sciences: A Primer on Evidence. Springer-Verlag New York.

See Also

`ICtab`

Examples

```
my.aic <- matrix(c(4,2,3,5),4,1)
colnames(my.aic) <- ”AIC”
rownames(my.aic) <- rownames(my.aic, do.NULL = FALSE, prefix = ”M.”)
my.aic
ICdetails(my.aic)

my.models.by.variables<-matrix(c(1,0,0,0,0,1,1,0,0,1,0,1,0,0,0,1),4,4)
colnames(my.models.by.variables) <- colnames(my.models.by.variables,
    do.NULL = FALSE, prefix = ”Var.”)
rownames(my.models.by.variables) <- rownames(my.aic)
my.models.by.variables
ICdetails(my.aic, variables = my.models.by.variables)
```

`mantel.residuals` *Residuals from Mantel test*

Description

Function to extract residuals from Mantel test.

Usage

```
mantel.residuals(y, x)
```

Arguments

`y`	Dissimilarity matrices of dist class.
`x`	Dissimilarity matrices of dist class.

Value

`statistic`	The Mantel statistic.
`residuals`	Residuals extracted from the Mantel test (Class dist).

Author(s)

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

Examples

```
#require(vegan)
#A<-vegdist(matrix(sample(1:10, 20, replace  =TRUE), 5, 5))
#B<-vegdist(matrix(sample(1:10, 30, replace = TRUE), 5, 6))
#mantel(A, B)
#mantel.residuals(A, B)
```

`node.tree` *Makes node labels*

Description

This function makes node labels in a tree keeping already existing node names. This function is based in the function `makeNodeLabel`.

Usage

```
node.tree(tree, m = 0, prefix = "NodE")
```

Arguments

`tree`	phylogeny as an object of class "phylo".
`m`	One number to starting sequence (default m = 0).
`prefix`	The prefix (default prefix = "NodE")

Value

A list with:

`call`	The call arguments.
`m.start`	The start m value.
`m.current`	The new m value, the difference between m.start and m.current is the number of node names created.
`prefix`	Prefix used.
`tree`	The tree, class "phylo".

Author(s)

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

See Also

`makeNodeLabel`

Examples

```
tree <- read.tree(text = "(C:32,(B:16,A:16):16);")
plot(tree, show.node = TRUE)
tree <- node.tree(tree, m = 0, prefix = "N")$tree
plot(tree, show.node = TRUE)
```

`organize.list` *Organize a list with data.frame/matrix in a single matrix*

Description

Function organize a list of data.frame or matrix in a single matrix. The function combine the data.frame/matrix by columns, thus all objects in the list must be equal number of columns

Usage

```
organize.list(x, force.names = FALSE)
```

Arguments

`x`	A list with data.frame or matrix
`force.names`	Logical argument (TRUE or FALSE) to specify if new row names are created, using list name as prefix.

Value

A matrix combined.

Author(s)

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

Examples

```
A<-matrix(1,10,6)
rownames(A) <- rownames(A, do.NULL = FALSE, prefix = "A.")
B<-matrix(2,3,6)
rownames(B) <- rownames(B, do.NULL = FALSE, prefix = "B.")
C<-matrix(3,15,6)
rownames(C) <- rownames(C, do.NULL = FALSE, prefix = "C.")
RES<-list(as.data.frame(A), as.data.frame(B), as.data.frame(C))
organize.list(RES)
organize.list(RES, force.names = TRUE)
```

plotcollapse.phylo *Plot phylogenetic tree with nodes collapsed.*

Description

This function plot phylogenetic tree with nodes collapsed. All arguments used are of the function plot.phylo. See details.

Usage

```
plotcollapse.phylo(tree, nodes, show.tip.label = TRUE,
  show.node.label = FALSE, edge.color = "black", edge.width = 1,
  edge.lty = 1, polygon.color = "gray", density = NULL, border = NULL,
  nhchar = 2, text.nodes.color = NULL, ...)
```

Arguments

tree	phylogeny as an object of class "phylo".
nodes	a vector with node label to collapse the edges.
show.tip.label	a logical indicating whether to show the tip labels on the phylogeny. The same used in the function plot.phylo (default show.tip.label = TRUE).
show.node.label	a logical indicating whether to show the node labels on the phylogeny. The same used in the function plot.phylo (default show.node.label = FALSE).
edge.color	a vector of colours giving the colours used to draw the branches of the plotted phylogeny. The same used in the function plot.phylo (default edge.color = "black").
edge.width	a numeric vector giving the width of the branches of the plotted phylogeny. The same used in the function plot.phylo (default edge.width = 1).
edge.lty	a numeric vector giving the line types of the branches of the plotted phylogeny. The same used in the function plot.phylo (default edge.lty = 1).

`polygon.color` a vector of colours for each node collapsed, this is used to filling each polygon (default polygon.color = "gray").

`density` a numeric vector with density of shading lines (lines per inch) for each node collapsed, this is used to filling each polygon (default density = NULL).

`border` a vector of colours for each node collapsed, this is used to draw the border in each polygon (default border = NULL).

`nhchar` number of hidden characters in tip.labels. This adjusts the margin of tip or nodes names are not shown completely (default nhchar = 2).

`text.nodes.color` a vector of colours to text label in each nodes collapsed. If NULL color of text is same that tip.label (default text.nodes.color = NULL).

`...` other arguments to function plot.phylo.

Details

The function use the function `plot.phylo` to plot the phylonenies. This function only hidden all edge, node labels and tips labels of each collapsed node and draw a polygon that represent a collapsed node. Type "phylogram" and "cladogram" work properly, other type of phyloneny will be implemented in future (Tip labels are not draw in correct position). The function `compact.tree` can be used to compact tips of phylogenetic tree before plot. The function `graphical.node.patterns` can be used to define graphical parameters (e.g. color, width and line types) for each edge in phylogenetic tree.

This function require node labels. The function `makeNodeLabel` or `node.tree` can be used to make node labels before plot, anyway the the function `node.tree` is apply automatically. Make sure that that there no conflict with node labels.

Value

The plot of phylogenetic trees with nodes collapsed and returns invisibly a list with components which values are those used for the plot. See details in `plot.phylo`.

Author(s)

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

See Also

`plot.phylo` `compact.tree` `graphical.node.patterns`

Examples

```
set.seed(10)
tree<-rtree(15)
tree <- makeNodeLabel(tree)
plot.phylo(tree, show.node.label = TRUE)
nodes <- c("Node9","Node2")

plot.phylo(tree, show.node.label = TRUE)
plotcollapse.phylo(tree, nodes)
plotcollapse.phylo(tree, nodes, nhchar = 5) # nhchar ajust the margin
plotcollapse.phylo(tree, nodes, show.tip.label = FALSE)
plotcollapse.phylo(tree, nodes, direction = "rightwards", polygon.color = c("red","blue"))
plotcollapse.phylo(tree, nodes, direction = "rightwards", polygon.color = c("red","blue"),
   text.nodes.color = c("red", "blue"))
plotcollapse.phylo(tree, nodes, direction = "upwards", polygon.color = c("red","blue"))
plotcollapse.phylo(tree, nodes, direction = "downwards", polygon.color = c("red","blue"))
plotcollapse.phylo(tree, nodes, direction = "leftwards", polygon.color = c("red","blue"))
plotcollapse.phylo(tree, nodes, direction = "leftwards", polygon.color = c("red","blue"),
   nhchar = 6)

set.seed(20)
tree <- rtree(15)
tree <- makeNodeLabel(tree)
plot.phylo(tree, show.node.label = TRUE)
tree.compacted <- compact.tree(tree, "Node6")
plot.phylo(tree.compacted, show.node.label = TRUE) # plot tree compacted
nodes <- c("Node5","Node6")
plotcollapse.phylo(tree, nodes, nhchar = 4)
plotcollapse.phylo(tree.compacted, nodes, nhchar = 4)

e.wid <- graphical.node.patterns(tree, nodes, basicpattern = 1,
    nodespatterns = 3, include.node = TRUE)
e.wid
plotcollapse.phylo(tree, nodes, show.node.label = TRUE, edge.width = e.wid)
e.col <- graphical.node.patterns(tree, nodes, basicpattern = "black",
    nodespatterns = c("red","blue"), include.node = TRUE)
e.col
plotcollapse.phylo(tree, nodes, show.node.label = TRUE, edge.width = e.wid, edge.color = e.col)

tree <- rtree(50)
tree <- compute.brlen(tree)
tree <- makeNodeLabel(tree)
plot.phylo(tree, type ="fan", show.tip.label = FALSE)
nodes <- "Node30"
plotcollapse.phylo(tree, nodes, type ="fan", show.tip.label = FALSE)
```

PVR *Phylogenetic Eigenvector Regression*

Description

Phylogenetic Eigenvector Regression (PVR) and eigenvector selection.

Usage

```
PVR(traits, dist, cumulative = 0.99, VMoran = 0.025, pMoran = 0.05,
  check = TRUE)

## S3 method for class 'pvr'
plot(x, trait = 1, ...)
```

Arguments

traits	Species described by continuous traits, with traits as columns and species as rows.

`dist` Phylogenetic distance matrix.

`cumulative` Percentage of variation in the phylogenetic distances considered in the analysis. Cumulative percentage must be higher than the cumulative percentage of the first two eigenvalues, and less than 1.

`VMoran` Stopping rule based on Moran I value (Absolute value, smaller than the specified value).

`pMoran` Stopping rule based on the p-value of Moran I (Greater than the specified). check Logical argument (TRUE or FALSE) that checks whether traits and phylogeny taxa labels match. The sequence of species in the trait data must be the same as that in the phylogenetic distance matrix.

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

`x` An object of class pvr.

`trait` Trait for plot.

`...` Other parameters for the respective functions.

Details

This function is based on a non-sequential approach, that uses the combination of eigenvectors that minimizes the residual phylogenetic autocorrelation, measured by Moran I. The method can be used to measure the level of phylogenetic signal in ecological data and to study correlated evolution (Diniz-Filho et al 2011).

The phylogenetic distance matrix is double-centered and submitted to principal coordinates analysis (PCoA). This method generates orthogonal eigenvectors that summarize the phylogenetic structure (Diniz-Filho et al 2008).

The sets of eigenvectors is selected with multiple regression model:

Y = a+Xb+e

where Y is a vector describing trait variation in the set of species, X contains a set of k eigenvectors, a the intercept, b is the vector with regression coefficients estimated and e its residuals, the part of variation in Y that is not explained by the X (Diniz-Filho et al 1998).

The function use an iterative search for the eigenvector that reduces the autocorrelation in the residuals. Primarily, the regression for all eigenvectors is calculated, obtaining the residuas. Then, Moran I for each eigenvector is calculated for the residuals. The function select the eingenvector with the lowest Moran I, and then, as new eigenvectors are added to the model, residuals are updated and autocorrelation is reestimated. The search stops when residual autocorrelation is reduced below threshold Moran I specified and when the statistical significance is reached (Diniz-Filho et al 2011).

Value

`values` Eigenvalues, relative eigenvalues and cumulative eigenvalues for the PCoA of distance matrix.

`vectors` The principal coordinates with positive eigenvalues.

`inf.cumulative` Percentage of the variation in the phylogenetic distances considered in the analysis (The result should be approximately the specified cumulative value).

`n.axis.considered` Number of axes considered.

`moran.less.than`
: Morans I value considered in the stopping rules (Absolute value).

`p.moran.greater.than`
: Stopping rule for the p-value.

`PSR.curve.axis.x`
: Values for the PSR curve (Abscissa).

`PSR.curve.axis.y`
: Values for the PSR curve (Ordinate, for each traits).

`minimun.moran` Parameters, number of parameters, observed Moran I, p-value for Moran I, R Squared and p-value for regression model that minimize autocorrelation coefficients in the residuals for each trait.

Note

The parameter in minimun.moran is shown as follows y ~ x[,3] + x[,5], in other words Trait~Axis_3+Axis_5 and so on.

Author(s)

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

References

Diniz-Filho, J. A. F., Santana, C. E. R., Bini, L. M. (1998). An eigenvector method for estimating phylogenetic inertia. Evolution, 52(5), 1247-1262.

Diniz-Filho, J.A.F., Bini, L. M., Rangel, T. F., Morales-Castilla, I., Olalla-Tarraga, M. A., Rodriguez, M. A. & Hawkins, B. A. (2011). On the selection of phylogenetic eigenvectors for ecological analyses. Ecography, 35(3), 239-249.

Examples

```
data(flona)
Res<-PVR(flona$traits[,1:4], flona$phylo, VMoran = 0.01)
Res
plot(Res, trait = 1)
```

`tree.label.info` *Show information about a label in a phylogenetic tree*

Description

Function for discover the edge position, edge length, edge height, max height of tree and type of the label in a phylogenetic tree. This function is based in the function `nodeHeights`.

Usage

```
tree.label.info(tree, label)
```

Arguments

`tree`	phylogeny as an object of class "phylo".
`label`	One tip label or one node label.

Value

A data.frame with type of label, edge position, edge length, edge height and max height of tree.

Author(s)

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

See Also

`nodeHeights`

Examples

```
tree <- read.tree(text = "(C:32,(B:16,A:16)N2:16)N1;")
plot(tree, show.node = TRUE)
tree.label.info(tree, label = "A")
```

Index