

treeDist	9
treescape	10
treescapeServer	11
treeVec	12
woodmiceTrees	13

Index	14
--------------	-----------

findGroves	<i>Identify clusters of similar trees</i>
------------	---

Description

This function uses hierarchical clustering on principal components output by [treescape](#) to identify groups of similar trees. Clustering relies on [hclust](#), using Ward's method by default.

Usage

```
findGroves(x, method = "treeVec", nf = NULL, clustering = "ward.D2",
           nclust = NULL, ...)
```

Arguments

x	an object of the class multiPhylo or the output of the function treescape
method	(ignored if x is from treescape) this specifies a function which outputs the summary of a tree in the form of a vector. Defaults to treeVec .
nf	(ignored if x is from treescape) the number of principal components to retain
clustering	a character string indicating the clustering method to be used; defaults to Ward's method; see argument <code>method</code> in <code>?hclust</code> for more details.
nclust	an integer indicating the number of clusters to find; if not provided, an interactive process based on cutoff threshold selection is used.
...	further arguments to be passed to treescape

Value

A list containing:

- groups: a factor defining groups of trees
- treescape: the output of [treescape](#)

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

See Also

[plotGroves](#) to display results

Examples

```
if(require("adegenet") && require("adegraphics")){
## load data
data(woodmiceTrees)

## run findGroves: treescape+clustering
res <- findGroves(woodmiceTrees, nf=5, nclust=6)

## plot results on first 2 axes
PCs <- res$treescape$pc$li
s.class(PCs, fac=res$groups, col=funky(6))

## using plotGroves
plotGroves(res)
}
```

linearMrca

Linear MRCA function

Description

Function to make the most recent common ancestor (MRCA) matrix of a tree, where entry (i,j) gives the MRCA of tips i and j. The function is linear, exploiting the fact that the tree is rooted.

Usage

```
linearMrca(tree, k = 0)
```

Arguments

tree an object of the class phylo which should be rooted.
k (optional) number of tips in tree, for faster computation

Author(s)

Michelle Kendall <michelle.louise.kendall@gmail.com>

Examples

```
## generate a random tree
x <- rtree(6)

## create matrix of MRCAs: entry (i,j) is the node number of the MRCA of tips i and j
linearMrca(x,6)
```

 medTree

Geometric median tree function

Description

Finds the geometric median of a set of trees according to the Kendall Colijn metric.

Usage

```
medTree(x, groups = NULL, lambda = 0, weights = NULL,
        emphasise.tips = NULL, emphasise.weight = 2,
        return.lambda.function = FALSE, save.memory = FALSE)
```

Arguments

x	An object of the class multiPhylo, containing the trees for which the median tree will be computed, or a matrix of tree vectors as given by treescape\$ <i>vectors</i> .
groups	an optional factor defining groups of trees; if provided, one median tree will be found for each group.
lambda	a number in [0,1] which specifies the extent to which topology (default, with lambda=0) or branch lengths (lambda=1) are emphasised. This argument is ignored if return.lambda.function=TRUE or if the vectors are already supplied as the object x.
weights	A vector of weights for the trees. Defaults to a vector of 1's so that all trees are equally weighted, but can be used to encode likelihood, posterior probabilities or other characteristics.
emphasise.tips	an optional list of tips whose entries in the tree vectors should be emphasised. Defaults to NULL.
emphasise.weight	applicable only if a list is supplied to emphasise.tips, this value (default 2) is the number by which vector entries corresponding to those tips are emphasised.
return.lambda.function	If true, a function that can be invoked with different lambda values is returned. This function returns the vector of metric values for the given lambda. Ignored if the tree vectors are already supplied as the object x.
save.memory	A flag that saves a lot of memory but increases the execution time (not compatible with return.lambda.function=TRUE). Ignored if the tree vectors are already supplied as the object x.

Value

A list of five objects: `$centre` is the "central vector", that is, the (weighted) mean of the tree vectors (which typically does not correspond to a tree itself); `$distances` gives the distance of each tree from the central vector; `$mindist` is the minimum of these distances; `$treenumbers` gives the numbers (and, if supplied, names) of the "median tree(s)": the tree(s) which achieve this minimum distance to the centre; `$trees` if trees were supplied then this returns the median trees as a `multiPhylo` object. If groups are provided, then one list is returned for each group. If `return.lambda.function=TRUE` a function that produces this list for a given value of `lambda` is returned.

Author(s)

Jacob Almagro-Garcia <nativecoder@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

Thibaut Jombart <thibautjombart@gmail.com>

Examples

```
## EXAMPLE WITH WOODMICE DATA
data(woodmiceTrees)

## LOOKING FOR A SINGLE MEDIAN
## get median tree(s)
res <- medTree(woodmiceTrees)
res

## plot first tree
med.tree <- res$trees[[1]]
plot(med.tree)

## LOOKING FOR MEDIANS IN SEVERAL CLUSTERS
## identify 6 clusters
groves <- findGroves(woodmiceTrees, nf=3, nclust=6)

## find median trees
res.with.grp <- medTree(woodmiceTrees, groves$groups)

## there is one output per cluster
names(res.with.grp)

## get the first median of each
med.trees <- lapply(res.with.grp, function(e) ladderize(e$trees[[1]]))

## plot trees
par(mfrow=c(2,3))
for(i in 1:length(med.trees)) plot(med.trees[[i]], main=paste("cluster",i))
```

`multiDist`*Metric function for multiPhylo input*

Description

Comparison of a list of trees using the Kendall Colijn metric. Output is given as a pairwise distance matrix. This is equivalent to the `$D` output from `treescap` but may be preferable for large datasets, and when principal co-ordinate analysis is not required. It includes an option to save memory at the expense of computation time.

Usage

```
multiDist(trees, lambda = 0, return.lambda.function = FALSE,  
          save.memory = FALSE, emphasise.tips = NULL, emphasise.weight = 2)
```

Arguments

<code>trees</code>	an object of the class <code>multiPhylo</code> containing the trees to be compared
<code>lambda</code>	a number in $[0,1]$ which specifies the extent to which topology (default, with <code>lambda=0</code>) or branch lengths (<code>lambda=1</code>) are emphasised. This argument is ignored if <code>return.lambda.function=TRUE</code> .
<code>return.lambda.function</code>	If true, a function that can be invoked with different <code>lambda</code> values is returned. This function returns the matrix of metric values for the given <code>lambda</code> .
<code>save.memory</code>	A flag that saves a lot of memory but increases the execution time (not compatible with <code>return.lambda.function=TRUE</code>).
<code>emphasise.tips</code>	an optional list of tips whose entries in the tree vectors should be emphasised. Defaults to <code>NULL</code> .
<code>emphasise.weight</code>	applicable only if a list is supplied to <code>emphasise.tips</code> , this value (default 2) is the number by which vector entries corresponding to those tips are emphasised.

Value

The distance matrix or a function that produces the distance matrix given a value for `lambda`.

Author(s)

Jacob Almagro-Garcia <nativecoder@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

Examples

```
## generate 10 random trees, each with 6 tips
trees <- rmtree(10,6)

## pairwise distance matrix when lambda=0
multiDist(trees)

## pairwise distance matrix as a function of lambda:
m <- multiDist(trees, return.lambda.function=TRUE)

## evaluate at lambda=0. Equivalent to multiDist(trees).
m0 <- m(0)

## save memory by recomputing each tree vector for each pairwise tree comparison (for fixed lambda):
m0.5 <- multiDist(trees,0.5,save.memory=TRUE)
```

plotGroves

Scatterplot of groups of trees

Description

This function displays the scatterplot of the Multidimensional Scaling (MDS) output by treescape, superimposing group information (derived by [findGroves](#)) using colors.

Usage

```
plotGroves(x, groups = NULL, xax = 1, yax = 2, type = c("chull",
  "ellipse"), col.pal = funky, bg = "white", lab.show = FALSE,
  lab.col = "black", lab.cex = 1, lab.optim = TRUE, point.cex = 1,
  scree.pal = NULL, scree.size = 0.2, scree.posi = c(0.02, 0.02), ...)
```

Arguments

x	a list returned by findGroves or a MDS with class dudi
groups	a factor defining groups of trees
xax	a number indicating which principal component to be used as 'x' axis
yax	a number indicating which principal component to be used as 'y' axis
type	a character string indicating which type of graph to use
col.pal	a color palette to be used for the groups
bg	the background color
lab.show	a logical indicating whether labels should be displayed
lab.col	a color for the labels

lab.cex	the size of the labels
lab.optim	a logical indicating whether label positions should be optimized to avoid overlap; better display but time-consuming for large datasets
point.cex	the size of the points
scree.pal	a color palette for the screeplot
scree.size	a size factor for the screeplot, between 0 and 1
scree.posi	either a character string or xy coordinates indicating the position of the screeplot.
...	further arguments passed to <code>s.class</code>

Details

This function relies on `s.class` from the `adegraphics` package.

Value

An `adegraphics` object (class: `ADEgS`)

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

See Also

[s.class](#)

Examples

```
## Not run:
if(require("adegenet") && require("adegraphics")){
  ## load data
  data(woodmiceTrees)

  ## run findGroves: treescape+clustering
  res <- findGroves(woodmiceTrees, nf=5, nclust=6)

  ## basic plot
  plotGroves(res)

  ## adding labels
  plotGroves(res, lab.show=TRUE)

  ## customizing
  plotGroves(res, lab.show=TRUE,
             bg="black", lab.col="white", scree.size=.35)

  ## customizing
  plotGroves(res, type="ellipse", lab.show=TRUE,
             lab.optim=FALSE, scree.size=.35)
```



```
## example with no group information
plotGroves(res$treescape$pco)

## adding labels
plotGroves(res$treescape$pco, lab.show=TRUE, lab.cex=2)

}

## End(Not run)
```

treeDist	<i>Metric function</i>
----------	------------------------

Description

Comparison of two trees using the Kendall Colijn metric

Usage

```
treeDist(tree.a, tree.b, lambda = 0, return.lambda.function = FALSE,
          emphasise.tips = NULL, emphasise.weight = 2)
```

Arguments

tree.a	an object of the class phylo
tree.b	an object of the class phylo (with the same tip labels as tree.a)
lambda	a number in [0,1] which specifies the extent to which topology (default, with lambda=0) or branch lengths (lambda=1) are emphasised. This argument is ignored if type="function".
return.lambda.function	If true, a function that can be invoked with different lambda values is returned. This function returns the vector of metric values for the given lambda.
emphasise.tips	an optional list of tips whose entries in the tree vectors should be emphasised. Defaults to NULL.
emphasise.weight	applicable only if a list is supplied to emphasise.tips, this value (default 2) is the number by which vector entries corresponding to those tips are emphasised.

Value

The vector with the metric values or a function that produces the vector given a value of lambda.

Author(s)

Jacob Almagro-Garcia <nativecoder@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

Examples

```
## generate random trees
tree.a <- rtree(6)
tree.b <- rtree(6)
treeDist(tree.a,tree.b) # lambda=0
treeDist(tree.a,tree.b,1) # lambda=1
dist.func <- treeDist(tree.a,tree.b,return.lambda.function=TRUE) # distance as a function of lambda
dist.func(0) # evaluate at lambda=0. Equivalent to treeDist(tree.a,tree.b).
## We can see how the distance changes when moving from focusing on topology to length:
plot(sapply(seq(0,1,length.out=100), function(x) dist.func(x)), type="l",ylab="",xlab="")

## The distance may also change if we emphasise the position of certain tips:
plot(sapply(tree.a$tip.label, function(x) treeDist(tree.a,tree.b,emphasise.tips=x)),
      xlab="Tip number",ylab="Distance when vector entries corresponding to tip are doubled")
```

treescape

Phylogenetic tree exploration

Description

Compares phylogenetic trees and maps them into a small number of dimensions for easy visualisation and identification of clusters.

Usage

```
treescape(x, method = "treeVec", nf = NULL, return.tree.vectors = FALSE,
  ...)
```

Arguments

x	an object of the class multiPhylo
method	the method for summarising the tree as a vector. Choose from: treeVec (default) the Kendall Colijn metric vector. The others are inherited from distTips in adephylo: patristic: for each pair of tips, the sum of branch lengths on the path between them nNodes: for each pair of tips, the number of nodes on the path between them Abouheif: performs Abouheif's test. See Pavoine et al. (2008) and adephylo. sumDD: sum of direct descendants of all nodes on the path, related to Abouheif's test. See adephylo.
nf	the number of principal components to retain
return.tree.vectors	option to also return the tree vectors. Note that this can use a lot of memory so defaults to FALSE.
...	further arguments to be passed to method.

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

Examples

```
## generate list of trees
x <- rmtree(10, 20)
names(x) <- paste("tree", 1:10, sep = "")

## use treescape
res <- treescape(x, nf=3)
table.paint(as.matrix(res$D))
scatter(res$pco)

data(woodmiceTrees)
woodmiceDists <- treescape(woodmiceTrees, nf=3)
plot(woodmiceDists$pco$li[,1], woodmiceDists$pco$li[,2])
woodmicedf <- woodmiceDists$pco$li
if(require(ggplot2)){
  woodmiceplot <- ggplot(woodmicedf, aes(x=A1, y=A2)) # create plot
  woodmiceplot + geom_density2d(colour="gray80") + # contour lines
  geom_point(size=6, shape=1, colour="gray50") + # grey edges
  geom_point(size=6, alpha=0.2, colour="navy") + # transparent blue points
  xlab("") + ylab("") + theme_bw(base_family="") # remove axis labels and grey background
}
## Not run:
if(require(rgl)){
  plot3d(woodmicedf[,1], woodmicedf[,2], woodmicedf[,3], type="s", size=1.5,
  col="navy", alpha=0.5, xlab="", ylab="", zlab="")
}

## End(Not run)
```

treescapeServer

Web-based tree explorer

Description

This function opens up an application in a web browser for an interactive exploration of the diversity in a set of trees.

Usage

```
treescapeServer()
```

Author(s)

Thibaut Jombart <thibautjombart@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

treeVec

Tree vector function

Description

Function which takes an object of class phylo as input and outputs the vector for the Kendall Colijn metric. The elements of the vector are numeric if `return.lambda.function=FALSE` (default), and otherwise they are functions of lambda.

Usage

```
treeVec(tree, lambda = 0, return.lambda.function = FALSE,
        emphasise.tips = NULL, emphasise.weight = 2)
```

Arguments

tree	an object of the class phylo
lambda	a number in [0,1] which specifies the extent to which topology (default, with lambda=0) or branch lengths (lambda=1) are emphasised. This argument is ignored if <code>return.lambda.function=TRUE</code> .
return.lambda.function	If true, a function that can be invoked with different lambda values is returned. This function returns the vector of metric values for the given lambda.
emphasise.tips	an optional list of tips whose entries in the tree vector should be emphasised. Defaults to NULL.
emphasise.weight	applicable only if a list is supplied to <code>emphasise.tips</code> , this value (default 2) is the number by which vector entries corresponding to those tips are emphasised.

Value

The vector with the metric values or a function that produces the vector given a value of lambda.

Author(s)

Jacob Almagro-Garcia <nativecoder@gmail.com>

Michelle Kendall <michelle.louise.kendall@gmail.com>

Examples

```
## generate a random tree
tree <- rtree(6)
## topological vector of mrca distances from root:
treeVec(tree)
## vector of mrca distances from root when lambda=0.5:
treeVec(tree,0.5)
## vector of mrca distances as a function of lambda:
vecAsFunction <- treeVec(tree,return.lambda.function=TRUE)
## evaluate the vector at lambda=0.5:
vecAsFunction(0.5)
```

woodmiceTrees

Bootstrap trees from woodmouse dataset

Description

These trees were created using the neighbour-joining and bootstrapping example from the ape documentation.

Usage

```
woodmiceTrees
```

Format

A multiPhylo object containing 201 trees, each with 15 tips

Source

A set of 15 sequences of the mitochondrial gene cytochrome b of the woodmouse (*Apodemus sylvaticus*) which is a subset of the data analysed by Michaux et al. (2003). The full data set is available through GenBank (accession numbers AJ511877 to AJ511987)

References

Michaux, J. R., Magnanou, E., Paradis, E., Nieberding, C. and Libois, R. (2003) Mitochondrial phylogeography of the Woodmouse (*Apodemus sylvaticus*) in the Western Palearctic region. *Molecular Ecology*, 12, 685-697

Index

*Topic **datasets**

woodmiceTrees, [13](#)

findGroves, [2](#), [7](#)

hclust, [2](#)

linearMrca, [3](#)

medTree, [4](#)

multiDist, [6](#)

plotGroves, [3](#), [7](#)

s.class, [8](#)

treeDist, [9](#)

treescape, [2](#), [10](#)

treescapeServer, [11](#)

treeVec, [12](#)

woodmiceTrees, [13](#)