

Package: recluster (via r-universe)

November 9, 2024

Type Package

Title Ordination Methods for the Analysis of Beta-Diversity Indices

Version 3.3

Date 2024-10-12

Depends alphahull, ape, sf, sp, vegan,

Imports grDevices, graphics, stats, utils, phytools, phangorn,
picante, cluster, plotrix

Author Leonardo Dapporto, Matteo Ramazzotti, Simone Fattorini, Roger
Vila, Gerard Talavera, Roger H.L. Dennis

Maintainer Leonardo Dapporto <leondap@gmail.com>

Description The analysis of different aspects of biodiversity requires specific algorithms. For example, in regionalisation analyses, the high frequency of ties and zero values in dissimilarity matrices produced by Beta-diversity turnover produces hierarchical cluster dendrograms whose topology and bootstrap supports are affected by the order of rows in the original matrix. Moreover, visualisation of biogeographical regionalisation can be facilitated by a combination of hierarchical clustering and multi-dimensional scaling. The recluster package provides robust techniques to visualise and analyse pattern of biodiversity and to improve occurrence data for cryptic taxa.

License GPL (>= 2.0)

LazyLoad yes

URL <https://github.com/leondap/recluster>

NeedsCompilation no

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libglpk-dev
libxml2-dev libssl-dev libproj-dev libsqlite3-dev
libudunits2-dev

Repository <https://leondap.r-universe.dev>

RemoteUrl <https://github.com/leondap/recluster>

RemoteRef HEAD

RemoteSha 041ebb505bd99f75ff0cb709cb912f58bb1f0faa

Contents

biodecrypt	3
biodecrypt.cross	5
biodecrypt.optimise	8
biodecrypt.plot	9
biodecrypt.view	10
biodecrypt.wrap	10
dataisl	13
datamod	13
multiboot	14
recluster	14
recluster.boot	17
recluster.col	18
recluster.cons	19
recluster.dist	21
recluster.expl	22
recluster.expl.diss	23
recluster.fst	24
recluster.fst.pair	25
recluster.group.col	27
recluster.hist	28
recluster.identify.nodes	28
recluster.line	30
recluster.multi	31
recluster.node.strength	32
recluster.plot	33
recluster.plot.col	34
recluster.plot.matrix	36
recluster.plot.pie	36
recluster.plot.sites.col	38
recluster.procrustes	40
recluster.region	41
recluster.rotate	43
recluster.test.dist	44
treebut	46
treemod	47

Index

48

biodecrypt	<i>An algorithm to attribute unidentified occurrence data based on a subset of identified records</i>
------------	---

Description

biodecrypt uses the function ahull from package alphahull to construct concave hulls with custom concavity (alpha) for each taxon. This function can also remove sea or ground areas from the analysis based on a SpatialPolygonsDataFrame representing the area of interest. The main input is represented by: i) a matrix of longitude and latitude (decimal degrees of longitude and latitude, WGS84) for all occurrence records, ii) a vector indicating species membership of each record in the same order of the matrix (1,2,..n for known species and 0 for cases to be attributed). By using spatial coordinates, the list of identified records and alpha values, biodecrypt computes a concave hull for each species based on known records. Then, the function attempts to attribute unknown cases to their most likely species based on the comparison of hull localtion, geometry and the location of occurrence data (see details).

Usage

```
biodecrypt(mat, id, alpha = NULL, ratio = 2.5, buffer = 90, polygon=NULL, checkdist = T, minimum = 7, plot = TRUE, xlim = NULL, ylim = NULL, main = NULL)
```

Arguments

mat	A matrix for longitude and latitude (in decimal degrees) for all records.
id	A vector indicating species membership of each record (in the same order of mat). Identified records are indicated with 1,2,..n, unidentified records with 0.
alpha	A vector indicating an initial alpha value for each species. If NULL, the default value of 8 for all species is used.
ratio	The minimum ratio between the distance from the second distant hull compared to the closest hull to allow attribution. Default 2.5 times.
buffer	A distance buffer from hulls (metres).
polygon	A SpatialPolygonsDataFrame with area of interests (ground or sea). Typically obtained from Natural Earth (https://www.naturalearthdata.com/). If NULL no removal is applied
checkdist	Logical, if TRUE cases attributed to a given species based on relative distance from hulls but closer to an identified record of another species are not attributed.
minimum	The minimum number of specimens required to build alpha hulls. If the number of identified specimens is lower, convex hulls are calculated to improve procedure stability.
map	A map to be plotted during the procedure to show the separation progress.
plot	Flag to FALSE is plotting the result is not required
xlim	Longitude boudaries for the map.
ylim	latitude boudaries for the map.
main	The name to be plotted on the graph

Details

Once the hulls for species are drawn according to the distribution of known records, each unidentified record could be either: i) inside more than one hull, ii) inside a single hull, or iii) outside all hulls. - Cases inside more than one hull In this case, the function cannot attribute the unidentified records to a species and only the a priori identified records belonging to intersection areas are passed to the final vector as identified. - Cases inside a single hull The unidentified records falling inside a single hull are attributed to that species if their distance to any other hull is higher than a buffer value provided by the user. Unidentified records inside the buffer of another hull are not attributed. - Cases outside all hulls The unidentified records which do not fall inside any hull are attributed to the closest hull if: i) the distance from the second nearest hull is higher than the buffer and if ii) the ratio between the minimum distance to the second closest hull and to the closest hull is more than a value indicated by the user (ratio). - Check for distances from the nearest identified record As described above, the attribution of unknown records is strictly determined by the distance from the hulls. The biodecrypt function also contains an option (checkdist=T) to check if cases attributed to a given species based on relative distance from hulls are closer to an identified record of another species, which may occasionally occur. If this option is selected (as in the default settings) these cases are not attributed to any species. In general, alpha values <5 return small and fragmented hulls with low predictive power; moreover, depending on the topology of occurrence data, some low alpha values could return no hulls. In this cases, biodecrypt tries to increase alpha up the the lowest value for which a hull is obtained. The alpha values used is stored in the alphaised vector.

Value

type	"sep" an argument to be passed to biodecrypt.plot.
areas	The areas of hulls for all the species (in km squares).
intersections	The areas of intersections among hulls for each pair of species.
sympatry	The fraction of the overlap area compared to the total area of the two hulls.
NUR	The percentage of Non-attributed Unidentified Records (NUR).
table	The result table with Longitude and Latitude for each occurrence datum, its id after the biodecrypt procedure (id2, the result of the procedure) and its initial id (id).
hulls	The hulls in sf format.
hullspl	The hulls in alphahull format.
alphaised	The values of alpha used fopr each species (see details).

Author(s)

Leonardo Dapporto

References

Platania L. et al. Assigning occurrence data to cryptic taxa improves climatic niche assessments: biodecrypt, a new tool tested on European butterflies. *Glocal Ecology and Biogeography* (2020)

Examples

```

n1<-7
n2<-7
mat<-rbind(cbind(rnorm(n = n1, mean = 1, sd = 2),rnorm(n = n1, mean = 40, sd = 2)),cbind(rnorm(n = n2, mean = 7, sd = 2),rnorm(n = n2, mean = 40, sd = 2)))

id<-c(rep(1,n1),rep(2,n2))
id[sample(c(1:(n1+n2)))[1:round((n1+n2)/4,0)]]<-0

# Make the separation with custom parameters
attribution<-biodecrypt(mat,id, alpha=c(10,10))

#plot the results
plot(mat,type="n")
biodecrypt.plot(attribution)

#Group plots into pies
biodecrypt.plot(attribution, square=2, minsize=0.5)

# Make the separation with custom parameters
# With a lower fraction values the first hull (alpha equal to 1) can become more
#concave. Excluded dots works as a punctiform sub-hull in the attribution.
attribution<-biodecrypt(mat,id, alpha=c(1,5), buffer=20, ratio=2, minimum=5)

#plot the results
plot(mat,type="n")
biodecrypt.plot(attribution)

{record attribution}

```

biodecrypt.cross

Perform a cross validation analysis to test the attribution of biodecrypt on attributed records

Description

The function `biodecrypt.cross` wraps the `biodecrypt` function to carry out cross-validation of known cases thus verifying the robustness of the attribution of unknown cases. This function requires the same input of `biodecrypt` (coordinates and vector with attribution together with values of distance ratio, buffer and alpha). Moreover this function requires a "runs" value defining the number of different runs, thus the fraction of test records included in each run. In each run, randomly selected group of test records (actually identified to a given species) are regarded as unidentified (0 value) and the `biodecrypt` function is carried out to attribute them. The analysis is repeated as often as defined in runs (a runs value of 10 will perform a ten-fold cross-validation based on the initial selection of ten randomly distributed subsets).

Usage

```
biodecrypt.cross(mat,id,alpha=NULL,ratio=2.5,buffer=90,fraction=0.95, partCount = 10,
checkdist=T, clipToCoast="terrestrial", proj = "+proj=longlat +datum=WGS84",minimum=7,
map=NULL,xlim=NULL,ylim=NULL,main=NULL,runs=10,test=T)
```

Arguments

mat	A matrix for longitude and latitude (in decimal degrees) for all records.
id	A vector indicating species membership of each record (in the same order of mat). Identified records are indicated with 1,2..n, unidentified records with 0.
alpha	A vector indicating an initial alpha value for each species. If NULL, the default value of 8 for all species is used.
ratio	The minimum ratio between the distance from the second distant hull compared to the closest hull to allow attribution. Default 2.5 times.
buffer	A distance buffer from hulls(in km).
fraction	The minimum fraction of occurrences that must be included in polygon.
partCount	The maximum number of disjunct polygons that are allowed..
clipToCoast	Either "no" (no clipping), "terrestrial" (only terrestrial part of range is kept) or "aquatic" (only non-terrestrial part is clipped).
checkdist	Logical, if TRUE cases attributed to a given species based on relative distance from hulls but closer to an identified record of another species are not attributed.
proj	the projection information for mat. In this version, the default is the only supported option.
minimum	The minimum number of specimens required to build alpha hulls. If the number of identified specimens is lower, convex hulls are calculated to improve procedure stability.
map	A map to be plotted during the procedure to show the separation progress.
xlim	Longitude boudaries for the map.
ylim	latitude boudaries for the map.
main	The name to be plotted on the graph
runs	The number of runs among which the cases are randomly assigned as non-attributed records
test	A logical, if TRUE, a biodecrypt analysis is also carried out to compute NUR.

Details

The procedure attributes the subsets of identified records to the test group (unknown cases) as evenly as possible among runs both in terms of total number of test records and records belonging to the same original species. If the number of runs equates the number of records, then each identified record is individually attributed in a jackknife procedure. Subsequently, the attribution vector obtained is provided and compared with the original membership and two values are provided: the percentages of identified cases attributed to a wrong species (Mis-Identified Records, MIR) and the percentage of known cases not attributed to any species (Non-attributed Identified Records, NIR).

The function also has an option to calculate the percentage of Non-attributed Unidentified Records (NUR) representing the fraction of unknown records that could not be attributed to a species after a typical biodecrypt analysis using the parameters provided by the user and the complete set of records.

Value

type	"cross" an argument to be passed to biodecrypt.plot.
NUR	The percentage of Non-attributed Unidentified Records.
areas	The hull areas for all the species (in km squares).
intersections	The areas of intersections among hulls for each pair of species.
sympatry	The fraction of the overlap area compared to the total area of the two hulls.
table	The result table of the test (if test=TRUE) with Longitude and Latitude for each occurrence datum, its id after the biodecrypt procedure (id2) and its initial id (id).
cross	The result table with the original attribution (original), the attribution obtained after cross validation (predicted) and the classification as MIR or NIR. Longitude and Latitude are also provided.
MIR	The percentage of Mis-Identified Records.
NIR	The percentage of Non-Identified Records.

Author(s)

Leonardo Dapporto

References

Platania L. et al. Assigning occurrence data to cryptic taxa improves climatic niche assessments: biodecrypt, a new tool tested on European butterflies. *Glocal Ecology and Biogeography* (2020).

Examples

```
## Not Run
## Create an example for a dataset

#mat<-rbind(cbind(rnorm(n = 20, mean = 1, sd = 4),rnorm(n = 20, mean = 40, sd = 3)),
#cbind(rnorm(n = 20, mean = 7, sd = 5),rnorm(n = 20, mean = 45, sd = 2))

#id<-c(rep(1,20),rep(2,20))
#id[sample(c(1:40))[1:10]]<-0

#cross<-biodecrypt.cross(mat,id)
#plot(mat,type="n")
#biodecrypt.plot(cross)
```

biodecrypt.optimise *Comparing the values obtained by biodecrypt.wrap, it optimises the combination of alpha, buffer and ratio values to be used with biodecrypt function.*

Description

The function biodecrypt.optimise analyses the output of biodecrypt.wrap. By default, a combination of $MIR^2+NIR+NUR$ is used as a penalty value for the different combinations of the parameters (providing a higher importance to MIR). The exponents can be changed by the user. Since the method showing the lowest penalty in cross-validation might not necessarily be the optimal value for the final analysis, all the combinations showing a penalty value not higher than a certain threshold compared with the analysis showing the lowest penalty should be considered as similarly good. We provided a value of 10 as a default, representing a variation of about 3 for each addendum of the penalty. The optimal parameters can then be calculated as mean values of distance ratio, alpha and buffer among those used in these cross-validation analyses, weighted by $1/penalty$ in order to provide an increasing contribution to the solutions showing the lowest penalty values.

Usage

```
biodecrypt.optimise(tab,coef=c(2,1,1), penalty=10)
```

Arguments

tab	A matrix obtained with biodecrypt.wrap.
coef	The three exponents to be applied to MIR, NIR and NUR, respectively, to calculate the penalties.
penalty	The penalty threshold for inclusion in the calculation.

Value

ratio	The optimized ratio value.
buffer	The optimized buffer value.
alpha	The optimized alpha value.
MIR	The weighted average MIR among selected combinations.
NIR	The weighted average NIR among selected combinations.
NUR	The weighted average NUR among selected combinations.

Author(s)

Leonardo Dapporto

References

Platania L. et al. Assigning occurrence data to cryptic taxa improves climatic niche assessments: biodecrypt, a new tool tested on European butterflies. *Glocal Ecology and Biogeography* (2020).

Examples

```
#See the example provided in biodecrypt.wrap
```

```
biodecrypt.plot          Plotting biodecrypt and biodecrypt.cross results.
```

Description

The function plots the results of `biodecrypt` and `biodecrypt.cross` analyses. It provides plot with circles with different colours to identify different kinds of records. Records known a priori can be distinguished in the plot from records attributed by `biodecrypt` as likely belonging to a given species or as NUR (or MIR and NIR in `biodecrypt.cross`).

Usage

```
biodecrypt.plot(x,minsize=0.3,pchid=1,cexid=0.1,square=0.001,col=c("red","darkgreen",
"blue","purple"), attributed=c("fade","points"), NUR="black", fading=50, ... )
```

Arguments

<code>x</code>	An object obtained by <code>biodecrypt</code> or <code>biodecrypt.cross</code>
<code>minsize</code>	The size of the dots to be plotted.
<code>pchid</code>	The pch of the points marking known cases in case when <code>attributed="points"</code> .
<code>cexid</code>	The size of the points marking known cases in case when <code>attributed="points"</code> .
<code>square</code>	The size of square grid to which occurrence are collapsed and organized in pies. If the value is lower than data resolution then records are not grouped in pies.
<code>col</code>	The colours to be attributed to species: 1...n.
<code>attributed</code>	The method to plot known records. Using <code>attributed="fade"</code> will make the attributed dots paler than known cases based on fading (see below). Using <code>attributed="points"</code> will plot a black dot to distinguish known cases.
<code>NUR</code>	The colour for NUR records after <code>biodecrypt</code> .
<code>fading</code>	The degree of fading for the colours of records attributed by <code>biodecrypt</code> if <code>attributed="fading"</code> (100 makes the points white).
<code>...</code>	other parameters of the default plot

Details

The function adds dots to a previous plot (usually a map). The records with a priori known attribution (1...n in id) are marked with a point inside the dots (`attributed="points"`) or by fading the colour of the dots for the records that have been attributed by `biodecrypt` (`attributed="fading"`). In the results of `biodecrypt.cross`, MIR are represented as black dots and NIR as white dots. For `biodecrypt` black default colour for NUR can be changed.

Value

a plot

Author(s)

Leonardo Dapporto

References

Platania L. et al. Assigning occurrence data to cryptic taxa improves climatic niche assessments: biodecrypt, a new tool tested on European butterflies. *Glocal Ecology and Biogeography* (2020).

Examples

#See examples in biodecrypt and biodecrypt.cross

biodecrypt.view	<i>A function to view the hulls and manually adjust alpha values before applying biodecrypt</i>
-----------------	---

Description

biodecrypt.view is no longer supported. Please use biodecrypt followed by biodecrypt.plot to achieve the same result.

biodecrypt.wrap	<i>Wraps the biodecrypt.cross analysis to compare the performance of biodecrypt among different parameters.</i>
-----------------	---

Description

The function biodecrypt.wrap wraps the biodecrypt.cross analysis by using all possible combinations of a series of distance ratio, alpha and buffer values to compare their resulting MIR, NIR and NUR.

Usage

```
biodecrypt.wrap(mat,id,alpha=c(1,5,10,15),alphamat=NULL,ratio=c(2,3,4,5),
buffer=c(0,40,80,120,160),fraction=0.95, partCount=10, checkdist=T,
clipToCoast="terrestrial", proj="+proj=longlat +datum=WGS84", minimum=7,
map=NULL,xlim=NULL,ylim=NULL,main=NULL,save=T,name="res_cross.txt",runs=10)
```

Arguments

mat	A matrix for longitude and latitude (in decimal degrees) for all records.
id	A vector indicating species membership of each record (in the same order of mat). Identified records are indicated with 1,2..n, unidentified records with 0.
alpha	A vector indicating the initial alpha values. It will be the same for all species
alphamat	A matrix indicating different alpha values for different species (optional).
ratio	The values of ratio.
buffer	The values of buffer.
fraction	The minimum fraction of occurrences that must be included in polygon.
partCount	The maximum number of disjunct polygons that are allowed..
checkdist	Logical, if TRUE cases attributed to a given species based on relative distance from hulls but closer to an identified record of another species are not attributed.
clipToCoast	Either "no" (no clipping), "terrestrial" (only terrestrial part of range is kept) or "aquatic" (only non-terrestrial part is clipped).
proj	the projection information for mat. In this version, the default is the only supported option.
minimum	The minimum number of specimens required to build alpha hulls. If the number of identified specimens is lower, convex hulls are calculated to improve procedure stability.
map	A map to be plotted during the procedure to show the separation progress.
xlim	Longitude boudaries for the map.
ylim	latitude boudaries for the map.
main	The name to be plotted on the graph
save	Logical, if TRUE a result table is saved after each biodecrypt.cross run
name	The name of the saved file
runs	The number of runs among which the cases are randomly assigned as non-attributed records

Details

The resulting table can be passed to biodecrypt.optimise to compute the best combination of alpha, buffer and ratio.

Value

table	The result table indicating for each cross validation test the MIR, NIR and NUR values together with the used ratio, buffer and alpha values.
-------	---

Author(s)

Leonardo Dapporto

References

Platania L. et al. Assigning occurrence data to cryptic taxa improves climatic niche assessments: biodecrypt, a new tool tested on European butterflies. *Glocal Ecology and Biogeography* (2020).

Examples

```
# Create an example for a dataset
mat<-rbind(cbind(rnorm(n = 20, mean = 1, sd = 4),rnorm(n = 20, mean = 40, sd = 3)),
  cbind(rnorm(n = 20, mean = 7, sd = 5),rnorm(n = 20, mean = 45, sd = 2)))

id<-c(rep(1,20),rep(2,20))
id[sample(c(1:40))[1:10]]<-0

## Not run: wrap_data_fast<-biodecrypt.wrap(mat,id, alpha=c(1,4), ratio=2,
  buffer=20, runs=2)
## End(Not run)
## Not run: parameters<-biodecrypt.optimise(wrap_data_fast$table,penalty=10)

#Make the example with default 10 runs and more values
## Not run: wrap_data<-biodecrypt.wrap(mat,id, alpha=c(1,4), ratio=c(2,4),
  buffer=c(20,50))
## End(Not run)
## Not run: parameters<-biodecrypt.optimise(wrap_data$table)

#inspect the optimised parameters
## Not run: parameters

#Use different alpha for the two species
#alpha for first

## Not run: alpha1<-c(1,3)

#alpha for second
## Not run: alpha2<-c(1,5)

## Not run: alphamat<-cbind(alpha1,alpha2)

## Not run: wrap_data<-biodecrypt.wrap(mat,id, alphamat=alphamat, ratio=c(2,4),
  buffer=c(20,50))
## End(Not run)

## Not run: parameters<-biodecrypt.optimise(wrap_data$table, penalty=20)

#inspect the optimised parameters
## Not run: parameters
```

dataisl	<i>West Mediterranean island butterflies provided with the package recluster</i>
---------	--

Description

This dataset represents occurrence data of butterfly species in 30 West-Mediterranean islands

Usage

```
data(dataisl)
```

Details

A data frame with 30 observations (islands) on 123 binary variables (species).

Author(s)

Leonardo Dapporto and Roger Vila

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

datamod	<i>Virtual island faunas provided with the package recluster</i>
---------	--

Description

This dataset represents a series of virtual faunas in different sites

Usage

```
data(datamod)
```

Details

A data frame with 9 observations (sites) on 31 binary variables (species).

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

multiboot

A multiboot result obtained with the dataisl dataset.

Description

This dataset represents an output for a multiscale bootstrap composed of 30 scales (x1-x30).

Usage

```
data(dataisl)
```

Details

A data frame with 29 nodes (rows) and 30 different scales of bootstrap(columns). NAs values represent collapsed nodes

Author(s)

Leonardo Dapporto

Source

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

recluster

Ordination methods for biodiversity patterns.

Description

The analysis of different aspects of biodiversity requires specific algorithms. For example, in regionalisation analyses, the high frequency of ties and zero values in dissimilarity matrices produced by Beta-diversity turnover produces hierarchical cluster dendrograms whose topology and bootstrap supports are affected by the order of rows in the original matrix. Moreover, visualisation of biogeographical regionalisation can be facilitated by a combination of hierarchical clustering and multi-dimensional scaling. The recluster package provides robust techniques to visualise and analyse pattern of biodiversity and to improve occurrence data for cryptic taxa.

Details

Package: recluster
Type: Package
Version: 3.0
Date: 2020-05-09
License: GPL (>= 2.0)

Author(s)

Leonardo Dapporto, Matteo Ramazzotti, Simone Fattorini, Roger Vila, Gerard Talavera, Roger H.L. Dennis Maintainer: Leonardo Dapporto <leondap@gmail.com>

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

Dapporto, L., Fattorini, S., Voda, R., Dinca, V., Vila, R. "Biogeography of western Mediterranean butterflies: combining turnover and nestedness components of faunal dissimilarity." *J Biogeogr* (2014), 41: 1639-1650.

Dapporto L., Voda R., Dinca V., Vila R. "Comparing population patterns for genetic and morphological markers with uneven sample sizes. An example for the butterfly *Maniola jurtina*" *Methods Ecol Evol* (2014), 5, 834-843.

Platania L., Menchetti M. Dinca V., Corbella C., Kay-Lavelle I., Vila R., Wiemers M., Schweiger O., Dapporto L. "Assigning occurrence data to cryptic taxa improves climatic niche assessments: biodecrypt, a new tool tested on European butterflies". *Glocal Ecology and Biogeography* (2020).

<https://github.com/leondap/recluster>

Examples

```
#load model data provided with the package
## Not run:
data(datamod)

#explore zero and tied values in the data set
simpdiss<- recluster.dist(datamod)
recluster.hist(simpdiss)

#create and view unbiased consensus tree (100
constree_full<-recluster.cons(datamod, tr=10, p=1)
plot(constree_full$cons,direction="downwards")

#compute and view node strength
recluster.node.strength(datamod, tr=10)

#create and view unbiased consensus tree (50
constree_half<-recluster.cons(datamod, tr=10, p=0.5)
plot(constree_half$cons, direction="downwards")

#the latter is the correct tree
tree<-constree_half$cons

#perform and view bootstrap on nodes
```

```

boot<-recluster.boot(tree, datamod, tr=10, p=0.5, boot=50)
recluster.plot(tree,boot)

#perform and view multiscale bootstrap on nodes
multiboot<- recluster.multi(tree, datamod, tr=10, boot=50, levels=2, step=1)
recluster.plot(tree,multiboot,low=1,high=2, direction="downwards")

#project and plot a bi-dimensional plot in the RGB colour space
sordiss<- recluster.dist(datamod,dist="sorensen")
points<-cmdscales(sordiss)
col<-recluster.col(points)
recluster.plot.col(col)

#inspect explained diversity for different cuts of a tree
tree<-recluster.cons(datamod, dist="sorensen",tr=10, p=0.5)
expl_div<-recluster.expl.diss(tree$cons,sordiss)
expl_div

#Select cut #4 and group data in RGB space
ncol<-recluster.group.col(col,expl_div$matrix[,4])

#Plot mean values for clusters
recluster.plot.col(ncol$aggr)

#Plot mean colours for sites in the geographic space
lat<-c(2,2,2,1,3,1,1,3,3)
long<-c(1,5,3,3,3,1,5,1,5)
recluster.plot.sites.col(long, lat, ncol$all,text=TRUE)

#Use recluster.region procedure on island butterflies
data(dataisl)
simpson<-recluster.dist(dataisl)
turn_cl<-recluster.region(simpson,tr=10,rettree=TRUE)
turn_cl

#Select solution with three cluster and plot the tree.
plot(turn_cl$tree[[2]])
turn_cl$grouping

#Perform a procrustes with uneven sample size
#Create and plot a target matrix
ex1 <-rbind(c(1,5),c(5,5),c(3,4),c(3,6))
plot(ex1,col=c(1:4),pch=19,xlim=c(0,6),ylim=c(0,6),cex=2)
#Create and plot a matrix to be rotated. Only the points 1-4 are shared
ex2<-rbind(c(3,1),c(3,3),c(2.5,2),c(3.5,2),c(3,4))
plot(ex2,col=c(1:5),pch=19,xlim=c(0,6),ylim=c(0,6),cex=2)

#Perform the procrustes on points 1-4
#Apply the transformation to point 5 of ex2 and plot the matrices
procr1<-recluster.procrustes(ex1,ex2,num=4)
plot(procr1$X,col=c(1:4),pch=19,xlim=c(-4,4),ylim=c(-4,4),cex=2)
plot(procr1$Yrot,col=c(1:5),pch=19,xlim=c(-4,4),ylim=c(-4,4),cex=2)

```



```

# Create an example for biodecrypt
mat<-rbind(cbind(rnorm(n = 20, mean = 1, sd = 4),rnorm(n = 20, mean = 40, sd = 3)),
cbind(rnorm(n = 20, mean = 7, sd = 5),rnorm(n = 20, mean = 45, sd = 2)))

id<-c(rep(1,20),rep(2,20))
id[sample(c(1:40))[1:10]]<-0

# Perform biodecrypt with default parameters
# alpha gets high to include 95
attribution<-biodecrypt(mat,id, clipToCoast="no")
#plot the results
plot(mat,type="n")
biodecrypt.plot(attribution)

## End(Not run)

```

recluster.boot

Bootstrap nodes of consensus trees

Description

Given an initial tree and a data matrix, this function computes bootstrap for nodes. Each tree used for bootstrap is constructed by re-sampling the row order several times and by applying a consensus rule as done by `recluster.cons`. The number of sampled columns (species) can be varied.

Usage

```
recluster.boot(tree, mat, phylo = NULL, tr = 100, p = 0.5,
dist = "simpson", method = "average", boot = 1000, level = 1)
```

Arguments

tree	A reference phylo tree for sites presumably constructed with <code>recluster.cons</code> function.
mat	The matrix used to construct the tree.
phylo	An ultrametric and rooted tree for species phylogeny having the same labels of the mat columns. Only required for phylogenetic beta-diversity indices.
tr	The number of trees to be included in the consensus.
p	A numeric value between 0.5 and 1 giving the proportion for a clade to be represented in the consensus tree.
dist	A beta-diversity index (the Simpson index by default) included in <code>recluster.dist</code> or any custom binary dissimilarity to be specified according to the syntax of <code>designdist</code> function of the <code>vegan</code> package.
method	Any clustering method allowed by <code>hclust</code> .
boot	The number of trees used for bootstrap computation.
level	The ratio between the number of species to be included in the analysis and the original number of species in the mat matrix.

Details

Computation can be time consuming due to the high number of trees required for analysis. It is suggested to assess the degree of row bias by `recluster.hist` and `recluster.node.strength` to optimize the number of required consensus trees before starting the analysis.

Value

A vector indicating the percentage of bootstrap trees replicating each original node.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

Examples

```
data(datamod)
tree<-recluster.cons(datamod,tr=10)
boot<-recluster.boot(tree$cons,tr=5,boot=50,datamod)
recluster.plot(tree$cons,boot,direction="downwards")

data(treemod)
tree<-recluster.cons(datamod,treemod, dist="phylosort", tr=10)
boot<-recluster.boot(tree$cons, datamod, treemod,tr=5,boot=50)
recluster.plot(tree$cons,boot,direction="downwards")
```

`recluster.col`*Projecting a two dimensional plot in RGB space*

Description

This function projects a two dimensional matrix into a RGB space with red, green, yellow and blue at its four corners. RGB combination for each case corresponding to its position in this space is provided together with new coordinates.

Usage

```
recluster.col(mat,st=TRUE,rot=TRUE)
```

Arguments

mat	A matrix containing two dimensional coordinates for cases.
st	Logical, if TRUE then values in axes are standardized between 0 and 1, if FALSE then original values are maintained.
rot	Logical, if TRUE then the axis with highest variance is oriented on the x-axis.

Value

A matrix with the first two columns representing the coordinates and the third, fourth and fifth representing the red, green and blue components, respectively.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Kreft H., Jetz, W. "A framework for delineating biogeographic regions based on species distributions" *J Biogeogr* (2010),37: 2029-2053.

Dapporto, L., Fattorini, S., Voda, R., Dinca, V., Vila, R. "Biogeography of western Mediterranean butterflies: combining turnover and nestedness components of faunal dissimilarity." *J Biogeogr* (2014), 41: 1639-1650.

Examples

```
data(datamod)
sordiss<- recluster.dist(datamod,dist="sorensen")
points<-cmdscale(sordiss)
col<-recluster.col(points)
col
```

recluster.cons

Consensus tree among re-sampled trees

Description

This function creates a series of trees by resampling the order of sites in the original dissimilarity matrix. Then, it computes a consensus among them. The resulting tree is independent of the original row order.

Usage

```
recluster.cons(mat, phylo = NULL, tr = 100, p = 0.5,
dist = "simpson", method = "average", blenghts=TRUE, select=FALSE)
```

Arguments

mat	A matrix containing sites (rows) and species (columns) or any dissimilarity matrix.
phylo	An ultrametric and rooted tree for species phylogeny having the same labels as in mat columns. Only required to compute phylogenetic beta-diversity indexes.
tr	The number of trees to be used for the consensus.
p	A numeric value between 0.5 and 1 giving the proportion for a clade to be represented in the consensus tree.
dist	A beta-diversity index (the Simpson index by default) included in recluster.dist or any custom binary dissimilarity to be specified according to the syntax of designdist function of the vegan package.
method	Any clustering method allowed by hclust.
blenghts	A logical indicating if non-negative least squares branch lengths should be computed.
select	A logical indicating if only trees having a fit higher than the median value in the least squares regression should be included in the consensus analysis.

Details

According to the primitive "consensus" function from the "ape" package, p must range between 0.5 and 1. Select = TRUE can allow lowering polytomies by removing trees with topology showing particularly low correlation with the distance matrix. Row names are required.

Value

cons	The consensus tree, an object of class phylo.
trees	The trees used to construct the final consensus tree.
RSS	The Residual Sum of Squares for the trees resulting if select=TRUE.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

Examples

```
#Faunistic beta diversity
data(datamod,treemod)
tree<-recluster.cons(datamod,tr=10)
plot(tree$cons,direction="downwards")

#Phylogenetic beta diversity
tree_p<-recluster.cons(datamod,treemod,dist="phylosort",tr=10)
plot(tree_p$cons, direction="downwards")
```

recluster.dist	<i>Compute a dissimilarity matrix using a battery of beta-diversity indices</i>
----------------	---

Description

This function computes dissimilarity matrices based on the two most popular partitions of faunistic and phylogenetic beta-diversity. In particular Jaccard = beta3 + richness (Carvalho et al. 2012), Jaccard = Jturnover + Jnestedness (Baselga, 2012) and Sorensen = Simpson + nestedness (Baselga 2010) for faunistic indexes and Unifrac = Unifrac_turn + Unifrac_PD and PhyloSor = PhyloSor_turn + Phylosor_PD (Leprieur et al. 2012). Any other binary index can be included in brackets by using the syntax of designdist function of the vegan package.

Usage

```
recluster.dist(mat, phylo=NULL, dist="simpson")
```

Arguments

mat	A matrix containing sites (rows) and species (columns).
phylo	An ultrametric and rooted phylogenetic tree for species having the same labels as in mat columns. Only required for phylogenetic beta-diversity indexes.
dist	One among the 14 beta-diversity indexes "simpson" "sorensen" "nestedness" "beta3" "richness" "jaccard" "jturnover" "jnestedness" "phylosor" "phylosort" "phylosorpd" "unifrac" "unifracturn" "unifracpd". Any custom binary dissimilarity can also be specified according to the syntax of designdist function of the vegan package.

Details

Syntax for binary indices in vegdist: J, number of common species; A and B, number of species exclusive of the first and of the second site.

Value

An object of class dist (see vegan:designdist for further details)

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Baselga A. "Partitioning the turnover and nestedness components of beta diversity." Global Ecol Biogeogr (2010), 19: 134-143.

Carvalho J. C., Cardoso P., Gomes P. "Determining the relative roles of species replacement and species richness differences in generating beta-diversity patterns." *Global Ecol Biogeogr* (2012), 21: 760-771.

Leprieur F., Albouy C., De Bortoli J., Cowman P.F., Bellwood D.R., Mouillot D. "Quantifying Phylogenetic Beta Diversity: Distinguishing between 'True' Turnover of Lineages and Phylogenetic Diversity Gradients." *Plos One* (2012), 7

recluster.expl *Computes the dissimilarity contained in a distance matrix which is explained by a clustering solution.*

Description

This function computes the fraction of the distances contained in a dissimilarity matrix which is explained by a clustering solution of the elements. The value is obtained by computing the sum of all the dissimilarity values among elements belonging to different clusters and divided by the sum of all the cells of the original dissimilarity matrix.

Usage

```
recluster.expl(mat, clust)
```

Arguments

mat	A dissimilarity matrix
clust	A clustering solution for the cases contained in the dissimilarity matrix.

Value

A number ranging between 0 and 1 indicating the fraction of explained dissimilarity.

Author(s)

Leonardo Dapporto

References

Holt, B.G. et al "An Update of Wallace's Zoogeographic Regions of the World." *Science*, 339:74-78.

Examples

```
data(datamod)
sor_tree<- recluster.cons(datamod, dist="sorensen")
sor_diss <- recluster.dist (datamod, dist="sorensen")
expl_diss <- recluster.expl.diss (sor_tree$cons,sor_diss)
expl_diss
```

recluster.expl.diss *Cuts a phylogenetic tree and provides cluster membership of areas for custom of all possible clustering solutions and their explained dissimilarity.*

Description

This function cuts a phylogenetic tree at all its nodes, and provides membership for each element in the series of resulting clusters and computes the fraction of dissimilarity explained by each solution.

Usage

```
recluster.expl.diss(tree, dist, maxcl=NULL, mincl=NULL, maxnode=NULL, expld=TRUE)
```

Arguments

tree	A phylo tree
dist	A dissimilarity matrix.
maxcl	A custom number indicating the solution with the minimum number of clusters. If NULL the minimum number of clusters is returned.
mincl	A custom number indicated the solution with the maximum number of clusters. If NULL the maximum number of clusters is returned
maxnode	A custom number indicated the most external node for the cut. If NULL all the nodes will be cut
expld	A logical. If TRUE then the matrix for explained dissimilarity is computed.

Details

When polytomic nodes are involved in a cut the number of clusters at that cut could increase more than one unit. It is also possible that at the first cut more than two cluster are identified, it is thus possible to obatin a first solution showing a higher number of clusters then the miminum number included in mincl. Holt at al. (2013) identified levels of explained dissimilarity to be used as a reliable threshold to assess a tree cut. When cases are highly numerous maxnode can be set in order to avoid a very long computation keeping in mind that a cut at node 6 can produce solutions with >6 clusters

Value

matrix	A matrix indicating cluster membership of each site in each cut of the tree.
expl.div	A vector indicating the explained dissimilarity for each cut.
nclust	A vector indicating the number of clusters resulting from each cut.

Author(s)

Leonardo Dapporto

References

Dapporto L., Ciolli G., Dennis R.L.H., Fox R., Shreeve, T.G. "A new procedure for extrapolating turnover regionalization at mid?small spatial scales, tested on B ritish butterflies." *Methods in Ecology and Evolution* (2015), 6:1287-1297.

Examples

```
data(datamod)
sor_tree<- recluster.cons(datamod, dist="sorensen")
sor_diss <- recluster.dist (datamod, dist="sorensen")
expl_diss <- recluster.expl.diss (sor_tree$cons,sor_diss)
expl_diss
```

<code>recluster.fst</code>	<i>Compute some indexes of genetic differentiation</i>
----------------------------	--

Description

This function computes some indexes of genetic differentiation based on a distance matrix and on a vector for populations.

Usage

```
recluster.fst(dist,vect,setzero=F,setnazero=F)
```

Arguments

<code>dist</code>	A distance matrix.
<code>vect</code>	A vector indicating population membership. Cases must be in the some order of the distance matrix.
<code>setzero</code>	A logical indicating if negative values should be set to zero
<code>setnazero</code>	A logical indicating if NA values should be set to zero

Details

There has been a large dabate around FST like indexes. Two main indexes are culcalated by this function: the absolute differentiation (Dst) and the standardized differentiation (Gst) (Nei, 1987). Dst is calculated as: $Dst = Ht - Hs$ where Ht represents the average distances among all the specimens in the sample, and Hs is the average of the intra-area (or intra-sub-area) distances. Thus, Dst represents the average genetic differentiation among areas in p-distance units. Gst is a standardized index defined as: $Gst = Dst/Ht$ representing the fraction of the total genetic differentiation encompassed by the differentiation among areas (Nei, 1987). This index ranges from negative values to 1 (complete differentiation). Negative values in Gst and Dst (intra-area differentiation higher than inter-area differentiation) can have different subtle meanings, but are most often generated as bias due to relatively small sample sizes; usually they are set to zero (Meirmans & Hedrick, 2011) and we applied this solution. In the species showing no mutations in the sample, Gst returns a NA

value (while Dst equals to zero). These cases can be also set to zero. The use of Dst and Gst has been debated as a measure of population diversification for extremely variable markers (as microsatellites) as it tends to underestimate differentiation among populations and to strongly depend on intra-population variability (Jost, 2008; Whitlock, 2011). D and G-st indices are less affected by high values of Hs

Value

Ht	The average distances among all the specimens in the sample.
lengthHt	The number of distances among all the specimens in the sample.
Hs	The average distances among the specimens of the same populations.
lengthHs	The number of distances among the specimens of the same populations.
Dst	The Dst value.
Gst	The Gst value.
D	The D value.
G1st	The G'st value.

Author(s)

Leonardo Dapporto

References

- Jost L. "GST and its relatives do not measure differentiation." *Mol Ecol* (2008), 17:4015-4026.
- Meirmans P. G., Hedrick P. W. "Assessing population structure: FST and related measures: Invited Technical Review." *Mol Ecol Res* (2011), 11: 5-18.
- Nei M. *Molecular evolutionary genetics* (1987), Columbia University Press.
- Whitlock M.C. "G'ST and D do not replace FST." *Mol Ecol* (2011), 20: 1083-1091.

Examples

```
datavirtual<-data.frame(replicate(10,sample(0:1,30,rep=TRUE)))
dist<-recluster.dist(datavirtual)
population<-c(rep(1,20),rep(2,20),rep(3,20))
recluster.fst(dist,population)
```

recluster.fst.pair	<i>Compute pairwise indexes of genetic differentiation among populations</i>
--------------------	--

Description

This function computes pairwise indexes of genetic differentiation among populations based on a distance matrix and on a vector for populations.

Usage

```
recluster.fst.pair(dist, vect, setzero=F, setnzero=F)
```

Arguments

dist	A distance matrix.
vect	A vector indicating population membership. Cases must be in the some order of the distance matrix.
setzero	A logical indicating if negative values should be set to zero
setnzero	A logical indicating if NA values should be set to zero

Details

The formulas used for pairwise calculations between i and j populations are $D_{stij} = H_{tij} - H_{sij}$
 $G_{stij} = D_{stij}/H_t$ $D_{ij} = (D_{stij}/(1-H_{sij}))*2$ $G'_{stij} = G_{stij}/((1-H_{sij})/(1+H_{sij}))$ see also recluster.fst for a discussion of indexes

Value

Dstm	The Dst distance matrix.
Gstm	The Gst distance matrix.
Dm	The D distance matrix.
G1stm	The G'st distance matrix.

Author(s)

Leonardo Dapporto

References

Jost L. "GST and its relatives do not measure differentiation." *Mol Ecol* (2008), 17:4015-4026.
 Meirmans P. G., Hedrick P. W. "Assessing population structure: FST and related measures: Invited Technical Review." *Mol Ecol Res* (2011), 11: 5-18.
 Nei M. *Molecular evolutionary genetics* (1987), Columbia University Press.
 Whitlock M.C. "G'ST and D do not replace FST." *Mol Ecol* (2011), 20: 1083-1091.

Examples

```
datavirtual<-data.frame(replicate(20,sample(0:1,60,rep=TRUE)))
dist<-recluster.dist(datavirtual)
population<-c(rep(1,20),rep(2,20),rep(3,20))
recluster.fst.pair(dist,population)
```

recluster.group.col *Computes mean coordinate values and RGB colours.*

Description

This function computes barycenters and their RGB colours for cases belonging to the same group from an original RGB colour matrix obtained by recluster.col.

Usage

```
recluster.group.col(mat, member)
```

Arguments

mat	An inherited matrix from recluster.col containing the original RGB colour space.
member	A vector indicating group membership for each case.

Value

aggr	A matrix in the recluster.col format with mean values for coordinates and RGB colours for groups.
all	A matrix in the recluster.col format reporting mean RGB colours of the group of each original case.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Kreft H., Jetz, W. "A framework for delineating biogeographic regions based on species distributions" *J Biogeogr* (2010),37: 2029-2053.

Dapporto, L., Fattorini, S., Voda, R., Dinca, V., Vila, R. "Biogeography of western Mediterranean butterflies: combining turnover and nestedness components of faunal dissimilarity." *J Biogeogr* (2014), 41: 1639-1650.

Examples

```
data(datamod)
sordiss<- recluster.dist(datamod,dist="sorensen")
points<-cmdscale(sordiss)
col<-recluster.col(points)
group<-c(1,2,3,3,3,1,2,1,2)
ncol<-recluster.group.col(col,group)
recluster.plot.col(ncol$aggr)
```

recluster.hist	<i>Histogram of dissimilarity with tied and zero values</i>
----------------	---

Description

This function creates a histogram with the values of a dissimilarity matrix where the number of cells with zero value are explicitly showed in the first bar. Moreover, it provides the percentage of cells having equal values in the matrix.

Usage

```
recluster.hist(x)
```

Arguments

x A dissimilarity matrix.

Value

An histogram with supplementary information. The first bar only shows the zero values.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

Examples

```
data(datamod)
simpdiss<- recluster.dist(datamod)
recluster.hist(simpdiss)
```

recluster.identify.nodes	<i>Evaluating solutions in multiscale bootstrap</i>
--------------------------	---

Description

This function helps to understand different behaviours of node supports in multiscale bootstrap by i) plotting trends of support values in different bootstrap scales, ii) identifying the bootstrap scale with highest diversification between two groups of nodes and iii) identifying nodes into two classes according to the best bootstrap level identified in (ii) and plotting their mean support values.

Usage

```
recluster.identify.nodes(mat, low=TRUE)
```

Arguments

mat	A matrix containing nodes (rows) and bootstrap levels (columns) as obtained by recluster.multi.
low	A logical value indicating if lower scales should be favoured in the selection.

Details

This function recognizes nodes showing different trends of support in multiscale bootstrap. In the analysis of turnover in biogeography some nodes may show a substantial increase in support in a multiscale bootstrap. Areas connected by these nodes may host a few species responsible for turnover, but the biogeographic pattern with respect is clear. Other nodes may show a slow (or no) increase in support. In this case, the links among areas can be considered as uncertain. Partitioning Around Medoids is used to identify two classes of nodes at each level, then the bootstrap scale showing the best diversification in two classes is identified by silhouette scores weighted by differences in mean values between classes. If "low" is set to TRUE the function favours low scales.

Value

A plot with bootstrap supports and their means (diamonds) for the best combination of two groups of nodes (black and red).

scale	The best bootstrap scale to identify two groups of nodes.
nodes	A vector containing classification for nodes in the best bootstrap scale.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

Examples

```
data(multiboot)
recluster.identify.nodes(multiboot)
```

recluster.line *Identifies a line in a configuration and computes its intercept and angular coefficient*

Description

This function identifies a line in a configuration based on different criteria and produces its slope and intercept values. It can be used together with `recluster.rotate` to rotate a configuration based on a custom line.

Usage

```
recluster.line(mat, type="maxd", X1=NULL, X2=NULL)
```

Arguments

mat	The bidimensional configuration.
type	The type of line to be computed: "maxd" is the line connecting the most distant points, "regression" is the regression line between X and Y values, "points" is the line connecting two custom points of the configuration (X1 and X2).
X1	The row number in mat of the first custom point.
X2	The row number in mat of the second custom point.

Value

m	The slope of the line.
q	The intercept of the line.

Author(s)

Leonardo Dapporto

References

Dapporto L., Voda R., Dinca V., Vila R. "Comparing population patterns for genetic and morphological markers with uneven sample sizes. An example for the butterfly *Maniola jurtina*" *Methods Ecol Evol* (2014), 5, 834-843.

Examples

```
data(dataisl)
#Compute bidimensional representation for islands
pcoa<-cmdscale(recluster.dist(dataisl))
#Compute the line
lin<-recluster.line(pcoa)
```

recluster.multi *Multiscale bootstrap based on a consensus tree*

Description

Given an initial tree and a data matrix, this function computes bootstrap for nodes as done by recluster.boot. Different levels of bootstrap can be computed by varying the proportions of species sampled from the original matrix.

Usage

```
recluster.multi(tree, mat, phylo = NULL, tr = 100, p = 0.5,
  dist = "simpson", method = "average", boot = 1000, levels = 2, step = 1)
```

Arguments

tree	A reference phylo tree for sites presumably constructed with recluster.cons function.
mat	The matrix used to construct the tree.
phylo	An ultrametric and rooted phylo tree for species having the same labels as in mat columns. Only required for phylogenetic beta-diversity indexes.
tr	The number of trees to be included in the consensus.
p	A numeric value between 0.5 and 1 giving the proportion for a clade to be represented in the consensus tree.
dist	One among the twelve beta-diversity indexes "simpson" "sorensen" "nestedness" "beta3" "richness" "jaccard" "phylosor" "phylosort" "phylosorpd" "unifrac" "unifract" "unifractpd". Any custom binary dissimilarity can also be specified according to the syntax of designdist function of the vegan package.
method	Any clustering method allowed by hclust.
boot	The number of trees used for bootstrap computation.
levels	The number of levels to be used in multiscale bootstrap.
step	The increase in ratio between the first level (x1) and the next ones.

Details

Computation can be time consuming. It is suggested to assess the degree of row bias by recluster.hist and recluster.node.strength to optimize the number of consensus trees before starting the analysis.

Value

A matrix indicating the percentage of bootstrap trees replicating each node for each level.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

Examples

```
data(datamod)
tree<-recluster.cons(datamod,tr=10)
multiboot<-recluster.multi(tree$cons,tr=10,boot=50,datamod,levels=2,step=1)
recluster.plot(tree$cons,multiboot,1,2,direction="downwards")
```

recluster.node.strength

Evaluating order row bias in a cluster

Description

This function helps to understand the magnitude of row bias by computing a first tree with the original order of areas. Then it creates a default series of six trees by `recluster.cons` with increasing consensus rule from 50

Usage

```
recluster.node.strength(mat, phylo = NULL, dist = "simpson",
  nodelab.cex=0.8, tr = 100, levels=6, method = "average", ...)
```

Arguments

<code>mat</code>	A matrix containing sites (rows) and species (columns).
<code>phylo</code>	An ultrametric and rooted phylogenetic tree for species having the same labels as in <code>mat</code> columns. Only required for phylogenetic beta-diversity indexes.
<code>tr</code>	The number of trees to be used for the consensus.
<code>dist</code>	A beta-diversity index (the Simpson index by default) included in <code>recluster.dist</code> or any custom binary dissimilarity to be specified according to the syntax of <code>designdist</code> function of the <code>vegan</code> package.
<code>nodelab.cex</code>	the <code>cex()</code> parameter for controlling the size of the labels on the nodes (see <code>?nodelabels</code>).
<code>levels</code>	The number of levels of different consensus threshold to be used.
<code>method</code>	Any clustering method allowed by <code>hclust</code> .
<code>...</code>	Arguments to be passed to <code>plot.phylo</code> methods, see the <code>ape</code> package manual and <code>?plot.phylo</code> .

Details

It has to be noted that values obtained by this function are not bootstrap supports for nodes but a crude indication of the magnitude of the row bias. Nodes with low value in this analysis can have strong bootstrap support and vice versa. This preliminary analysis can avoid that the use of a strict consensus (100

Value

A cluster with percentages of recurrence over different consensus runs for each node.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

Examples

```
data(datamod)
recluster.node.strength(datamod, tr=10)
```

recluster.plot	<i>A plotter for recluster bootstrapped objects</i>
----------------	---

Description

This function produces plots for recluster trees and assigns single or pairs of support values belonging to single or multiscale analyses.

Usage

```
recluster.plot(tree, data, low = 1, high = 0, id=NULL,
  nodelab.cex=0.8, direction="downwards",...)
```

Arguments

tree	A phylo tree presumably constructed with recluster.cons function.
data	A matrix belonging to recluster.multi.
id	A vector used to mark node supports (low and high) with different colours. Such classification is presumably made by recluster.identify.nodes.
low	The low scale level for which bootstrap values should be indicated in the tree.
high	The high scale level for which bootstrap values should be indicated in the tree.

nodelab.cex	the cex() parameter for controlling the size of the labels on the nodes (see ?nodelabels).
direction	the direction parameter for controlling the orientation of the plot, see the ape package manual and ?plot.phylo. This parameters also controls the display of the labels on nodes.
...	Arguments to be passed to plot.phylo methods, see the ape package manual and ?plot.phylo.

Details

This function allows to print on a tree, one or two labels for bootstrap values and optimize their layout. This is done with the nodelabels ape function, by specifying the adj parameters in the appropriate way.

Value

A plot representing the tree with pairs of bootstrap values, below (usually x1 BP above) and high, above.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

Examples

```
data(datamod)
tree<-recluster.cons(datamod, tr=10)
boot<-recluster.boot(tree$cons,datamod, tr=10, boot=50)
recluster.plot(tree$cons,boot,direction="downwards")
```

recluster.plot.col *Plotting data in RGB space*

Description

This function plots a matrix obtained by recluster.col in the RGB space.

Usage

```
recluster.plot.col(mat,cext=0.3,cex=1,cex.axis=0.7,cex.lab=0.8,pch=16,text=TRUE,
add=F,xlim=NULL,ylim=NULL,ylab="Axis 2",xlab="Axis 1",...)
```

Arguments

mat	A matrix inherited by recluster.col.
cext	Dimension for labels of row names.
cex	Dimension of dots.
cex.axis	Dimension of axis labels.
cex.lab	Dimension of labels.
text	A logical indicating if row names should be plotted.
pch	The shape of the dots (See par()).
add	A logical indicating if the plot should be added to a precedent graph.
xlim	The limit values for x-axis, if NULL the values in the original matrix is used.
ylim	The limit values for y-axis, if NULL the values in the original matrix is used.
ylab	The label of the y-axis
xlab	The label of the x-axis
...	See par() for other graphical parameters

Value

A colour plot.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Kreft H., Jetz, W. 2010. "A framework for delineating biogeographic regions based on species distributions" *J Biogeogr* (2010),37: 2029-2053.

Dapporto, L., Fattorini, S., Voda, R., Dinca, V., Vila, R. "Biogeography of western Mediterranean butterflies: combining turnover and nestedness components of faunal dissimilarity." *J Biogeogr* (2014), 41: 1639-1650.

Examples

```
data(datamod)
sordiss<- recluster.dist(datamod,dist="sorensen")
points<-cmdscale(sordiss)
col<-recluster.col(points)
recluster.plot.col(col)
```

`recluster.plot.matrix` *Plot the values of the cells of a matrix in grey scale*

Description

This function plots the values of the cells of a matrix in grey scale.

Usage

```
recluster.plot.matrix(mat)
```

Arguments

`mat` A dissimilarity matrix.

Value

A plot of cell values.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Dapporto, L., Fattorini, S., Voda, R., Dinca, V., Vila, R. "Biogeography of western Mediterranean butterflies: combining turnover and nestedness components of faunal dissimilarity." *J Biogeogr* (2014), 41: 1639-1650.

Examples

```
data(datamod)
simpdiss<- recluster.dist(datamod)
recluster.plot.matrix(simpdiss)
```

`recluster.plot.pie` *Plotting pies with RGB colours on a custom coordinate space*

Description

This function groups cases based on a space grid in a user defined set of coordinates (usually longitude and latitude) and plot them in pies using RGB colours. The function can either use an output from `recluster.col` function or compute colours based on any distance matrix where the cases are in the same order as in the latitude and longitude data.

Usage

```
recluster.plot.pie(long, lat, mat=NULL, distance=NULL, loc=NULL, areas=NULL, square=2,
                  map=NULL, add=FALSE, minsize=NULL, proportional=T, xlim=NULL, ylim=NULL,
                  main=NULL, xlab=NULL, ylab=NULL, ...)
```

Arguments

long	A vector indicating longitude for cases.
lat	A vector indicating latitude for cases.
mat	A matrix inherited by recluster.col.
distance	A dissimilarity matrix for cases.
loc	A list of localities to group cases, if available.
square	The grid to be used to divide cases into groups (2 degrees latitude and longitude by default).
areas	An additional vector to divide groups (e.g. islands versus continents).
map	A map to be plotted.
add	A logical. If TRUE then the points are added to an existing graph.
minsize	Dimension for the dimension of a single-case pie.
proportional	A logical. If TRUE then the point area is proportional to the number of cases.
xlim	Limits of the plot in the x-axis.
ylim	Limits of the plot in the y-axis.
main	The title of the graph.
xlab	The label of x-axis
ylab	The label of y-axis
...	See par() for other graphical parameters

Value

A colour plot.

Author(s)

Leonardo Dapporto

References

Hernandez Roldan J.L., Dapporto L., Dinca V, Vicente J.C., Hornett E.A., Sichova J., Lukhtanov V.L., Talavera G. & Vila, R. Integrative analyses unveil speciation linked to host plant shift in *Spialia* butterflies. *Molecular Ecology* (2016) 25: 4267-4284.

Examples

```

# create a virtual dataset and a corresponding distance matrix
lat<-runif(50,min=20,max=40)
long<-runif(50,min=20,max=40)
datavirtual<-data.frame(replicate(20,sample(0:1,50,rep=TRUE)))
dist<-recluster.dist(datavirtual)

# Make a plot using a custom distance
recluster.plot.pie(long,lat,distance=dist,xlab="Longitude",ylab="Latitude")

# Make a plot using a recluster.col matrix
colours<-recluster.col(cmdscale(dist))
recluster.plot.pie(long,lat,mat=colours,xlab="Longitude",ylab="Latitude")

# Make points of equal size
recluster.plot.pie(long,lat,mat=colours,xlab="Longitude", proportional=FALSE,
ylab="Latitude")

# Reduce the grid
recluster.plot.pie(long,lat,distance=dist,square=1, xlab="Longitude",ylab="Latitude")

# Reduce the size of the plots
recluster.plot.pie(long,lat,distance=dist,xlab="Longitude",ylab="Latitude", minsize=0.5)
# Use a custom colour matrix
pcoa<-cmdscale(dist)
colour<-recluster.col(pcoa)
recluster.plot.col(colour)
recluster.plot.pie(long,lat,mat=colour,xlab="Longitude",ylab="Latitude")

# Include an additional factor for separating dots in groups(e.g. two continents)
continent<-rep(1,50)
continent[which(long>25)]<-2
recluster.plot.pie(long,lat,distance=dist,xlab="Longitude",ylab="Latitude",
areas=continent)

```

```
recluster.plot.sites.col
```

Plotting RGB dots on a custom coordinate space

Description

This function plots the RGB dots belonging to a matrix obtained by `recluster.col` on a user defined set of coordinates (usually longitude and latitude) for original sites.

Usage

```
recluster.plot.sites.col (long, lat, mat, cext = 0.3, cex = 1, cex.axis = 0.7,
cex.lab = 0.8, text = FALSE, pch=21, add = FALSE,...)
```

Arguments

<code>long</code>	A vector indicating longitude for cases.
<code>lat</code>	A vector indicating latitude for cases.
<code>mat</code>	A matrix inherited by <code>recluster.col</code> .
<code>text</code>	A logical indicating if row names should be plotted.
<code>cext</code>	Dimension for row names.
<code>cex</code>	Dimension of dots.
<code>cex.axis</code>	Dimension of axis labels.
<code>cex.lab</code>	Dimension of labels.
<code>add</code>	A logical. If TRUE then the points are added to an existing graph.
<code>pch</code>	The symbol to use when plotting points
<code>...</code>	See <code>par()</code> for other graphical parameters

Value

A colour plot.

Author(s)

Leonardo Dapporto and Matteo Ramazzotti

References

Dapporto, L., Fattorini, S., Vod?, R., Dinc?, V., Vila, R. "Biogeography of western Mediterranean butterflies: combining turnover and nestedness components of faunal dissimilarity." *J Biogeogr* (2014), 41: 1639-1650.

Examples

```
data(datamod)
sordiss<- recluster.dist(datamod, dist="sorensen")
lat<-c(2,2,2,1,3,1,1,3,3)
long<-c(1,5,3,3,3,1,5,1,5)
points<-cmdscale(sordiss)
col<-recluster.col(points)
recluster.plot.sites.col(long, lat, col,text=TRUE)
```

recluster.procrustes *Computes a procrustes analysis between two matrices even if only a subset of cases are shared.*

Description

This function computes a procrustes analysis (as done by the `vegan` `procrustes` function) but it also allows including a subset of cases shared between the two matrices and some unshared cases. The shared cases must be listed first and in the same order in the two matrices. Moreover, the number of shared cases must be indicated. The function applies a procrustes analysis by scaling, mirroring and rotating the second matrix to minimizing its dissimilarity from the first on the basis of shared cases. Then, the same transformation is applied to the unshared cases of the second matrix. Finally, it allows including the matrices of coordinates for variables as obtained, for example, by PCA.

Usage

```
recluster.procrustes(X, Y, Yv=FALSE, num=nrow(X), scale = TRUE, ...)
```

Arguments

X	Target matrix.
Y	Matrix to be rotated.
Yv	Matrix of variables for the matrix to be rotated.
num	number of shared cases between the target matrix and the matrix to be rotated (by default all).
scale	number of shared cases between the target matrix and the matrix to be rotated (by default all).
...	See <code>procrustes()</code> for other parameters

Details

`recluster.procrustes` uses the `vegan` function `procrustes` to rotate a configuration (Y) to maximum similarity with another target matrix configuration (X) on the basis of a series of shared objects (rows). These objects must be in the same order in the two X and Y matrices. In case of additional cases (rows) in both the X and Y matrices, the same transformation is applied to the case of the Y matrices which are not shared with X. Moreover, the same transformation can be applied to an additional Yv matrix likely representing the coordinates of variables as obtained for example by PCA or other ordination methods. The function returns an object of the class "procrustes" as implemented in `vegan`.

Value

Yrot	Rotated matrix Y.
X	Target matrix.
Yvrot	Rotated matrix of variables Yv.

ss	Sum of squared differences between X and Yrot on the basis of shared objects.
rotation	Orthogonal rotation matrix on the basis of shared objects.
translation	Translation of the origin on the basis of shared objects.
scale	Scaling factor on the basis of shared objects.
xmean	The centroid of the target on the basis of shared objects.

Author(s)

Leonardo Dapporto

References

Dapporto L., Voda R., Dinca V., Vila R. "Comparing population patterns for genetic and morphological markers with uneven sample sizes. An example for the butterfly *Maniola jurtina*" *Methods Ecol Evol* (2014), 5, 834-843.

Examples

```
#Create and plot a target matrix
ex1 <-rbind(c(1,5),c(5,5),c(3,4),c(3,6))
plot(ex1,col=c(1:4),pch=19,xlim=c(0,6),ylim=c(0,6),cex=2)
#Create and plot a matrix to be rotated. Only the points 1-4 are shared
ex2<-rbind(c(3,1),c(3,3),c(2.5,2),c(3.5,2),c(3,4))
plot(ex2,col=c(1:5),pch=19,xlim=c(0,6),ylim=c(0,6),cex=2)

#Perform the procrustes and plot the matrices
procr1<-recluster.procrustes(ex1,ex2,num=4)
plot(procr1$X,col=c(1:4),pch=19,xlim=c(-4,4),ylim=c(-4,4),cex=2)
plot(procr1$Yrot,col=c(1:5),pch=19,xlim=c(-4,4),ylim=c(-4,4),cex=2)
```

recluster.region	<i>A clustering method based on continuous consensus among clustering solutions after resampling row order.</i>
------------------	---

Description

This function is specifically designed to facilitate regionalization analysis in cases where zero and tied values are particularly frequent. This often occurs when using turnover indices at small or intermediate spatial scales where large barriers are absent. The function requires a matrix as input, with areas in rows and species occurrence (1,0) in columns. It also allows for the inclusion of a phylogenetic tree to compute phylogenetic beta-diversity.

The indices used are those supported by `recluster.dist`, but custom indices can also be introduced (see `recluster.dist`). Alternatively, a dissimilarity matrix generated by any function can be provided. The function requires input for a custom number of trees (default $n=50$) and a range of `mincl-maxcl` values (default 2-3), indicating the number of regions to be identified. Clustering methods implemented in `hclust` are supported, as well as Partition Around Medoids (PAM) and DIANA.

The default method, ward.2D, typically offers the best performance, but ward.D, complete linkage clustering, PAM, and DIANA may also perform well.

The function generates n trees by randomly reordering the original row order. These trees are then cut at different nodes (from the mincl -1th to the maxcl -1th node), resulting in an increasing number of clusters. The function compares clustering solutions at the same cut levels across different resampled trees, producing a dissimilarity matrix between areas based on how often each pair of areas appears in different clusters across the different tree solutions at the same cut level. This dissimilarity is standardized by the number of resampled trees, yielding values from 0 (for pairs of areas always in the same cluster) to 1 (for pairs never in the same cluster).

A final hierarchical clustering is applied to generate an interval of maxcl - mincl . Since the user-defined number of clusters may not exactly match the mean number of clusters obtained from the tree cuts, the clustering solution for each k value is selected from the dissimilarity matrix closest to the mean number of clustering solutions.

Usage

```
recluster.region (mat, tr=50, dist="simpson", method="ward.D2", members=NULL, phylo=NULL, mincl=2, maxcl=
rettree=FALSE, retmat=FALSE, retmemb=FALSE)
```

Arguments

mat	A binary presence-absence community matrix or any dissimilarity matrix.
tr	The number of trees to be included in the consensus.
dist	One among the beta-diversity indexes allowed by <code>recluster.dist</code> or a custom binary dissimilarity specified according to the syntax of <code>designdist</code> function of the <code>vegan</code> package. Not required when the input is a dissimilarity matrix.
method	Any clustering method allowed by <code>hclust</code> but also "pam" and "diana".
members	For <code>hclust</code> methods, a vector.
phylo	An ultrametric and rooted phylogenetic tree for species having the same labels as in <code>mat</code> columns. Only required for phylogenetic beta-diversity indices.
mincl	The minimum number of regions requested
maxcl	The maximum number of regions requested
rettree	Logical, if TRUE the final trees are returned.
retmat	Logical, if TRUE the new dissimilarity matrices are returned.
retmemb	Logical, if TRUE the memberships for areas in different random trees is returned.

Details

Like other evaluators for goodness of clustering solutions, the function provides silhouette values and the explained dissimilarity. The explained dissimilarity (sensu Holt et al. 2013) is represented by the ratio between sums of mean dissimilarities among members of different clusters and the sum of all dissimilarities of the matrix. This value clearly tends to 1 when all areas are considered as independent groups. Silhouette width measures the strength of any partition of objects from a dissimilarity matrix by comparing the minimum distance between each cell and the most similar

cell belonging to any other cluster and the mean distance between that cell and the others belonging to the same cluster (see silhouette function in the cluster package). Silhouette values range between -1 and +1, with a negative value suggesting that most cells are probably located in an incorrect cluster.

Value

memb	An array with different matrices indicating for each area (rows) the membership in each random tree (columns) in each cut (matrix).
matrices	The new dissimilarity matrices. Up-right cells provided as NAs.
nclust	Mean number of clusters among random trees obtained by different cuts.
solutions	A matrix providing number of clusters for each solution (k), the associated mean number of clusters obtained by cuts (clust), the silhouette (silh) value and the explained dissimilarity (ex.diss).
grouping	A matrix indicating cluster membership of each site in each solution for different numbers of clusters.

Author(s)

Leonardo Dapporto

References

Dapporto L. et al. A new procedure for extrapolating turnover regionalization at mid-small spatial scales, tested on British butterflies. *Methods Ecol Evol* (2015), 6, 1287-1297

Examples

```
data(dataisl)
simpson<-recluster.dist(dataisl)
turn_cl<-recluster.region(simpson,tr=10,rettree=TRUE)
#plot the three for three clusters
plot(turn_cl$tree[[2]])
#inspect cluster membership
turn_cl$grouping
```

recluster.rotate *Rotates a bidimensional configuration according to a line*

Description

This function rotates the points of a configuration to a new configuration where a line identified by its intercept and its angular coefficient is rotated to become horizontal. The function can also flip or centre a configuration

Usage

```
recluster.rotate(table,m=FALSE,q=FALSE,flip="none",centre=TRUE)
```

Arguments

table	The bidimensional configuration.
m	The line slope.
q	The line intercept
flip	The kind of flip, no flip, "none"; "hor", flip horizontally; "ver", flip vertically; "both", flip vertically and horizontally.
centre	A logical. If TRUE the configuration, after transformation is centered to the mean X and Y values.

Value

table2	The transformed bidimensional configuration.
--------	--

Author(s)

Leonardo Dapporto

References

Dapporto L., Voda R., Dinca V., Vila R. "Comparing population patterns for genetic and morphological markers with uneven sample sizes. An example for the butterfly *Maniola jurtina*" *Methods Ecol Evol* (2014), 5, 834-843.

Examples

```
data(dataisl)
#Compute bidimensional representation for islands
pcoa<-cmdscale(recluster.dist(dataisl))
plot(pcoa)
#Compute the line
lin<-recluster.line(pcoa)
transf<-recluster.rotate(pcoa,m=lin$m,q=lin$q)
plot(transf)
```

recluster.test.dist	<i>Test variation lost by a bidimensional configuration when the coordinates of the elements are reduced to the configuration of the barycentres of a given series of groups.</i>
---------------------	---

Description

This function evaluates the amount of variation maintained by a bidimensional configuration after the elements are reduced to the barycentres according to a grouping variable. If elements of different groups are randomly scattered in the configuration, almost all barycentres are expected to attain a rather central position with respect to the original elements, which would result in a small mean distance between barycentres. Conversely, if the elements of different groups are strictly clustered in the representation, the distances among barycentres are expected to be similar to the distances among original elements.

Usage

```
recluster.test.dist(mat1,mat2,member,perm=1000,elev=2)
```

Arguments

mat1	The bidimensional configuration before computing barycentres for groups.
mat2	The bidimensional configuration after computing barycentres for groups.
member	A vector indicating group membership for each element.
perm	The number of permutations.
elev	The power of distances (by default 2:squared distances).

Details

The function produces a ratio between the mean squared pairwise distance for all elements and the mean squared pairwise distance for barycentres. This ratio is calculated for the overall configuration and for the two axes separately. The function also provides a test for the significance of the variation preserved by barycentres by creating a custom number of matrices (1000 by default) by randomly sampling the original vector defining groups. Then it computes the frequency of mean squared distance ratios in random configurations higher than the observed ratio.

Value

ratio	The ratio between mean distances among original elements and barycentres over the overall configuration.
ratioX	The ratio between mean distances among original elements and barycentres on the X axis.
ratioY	The ratio between mean distances among original elements and barycentres on the Y axis.
test	The permutation test for variation maintained over the overall configuration.
testX	The permutation test for variation maintained along the X axis.
testY	The permutation test for variation maintained along the Y axis.

Author(s)

Leonardo Dapporto

References

Dapporto L., Voda R., Dinca V., Vila R. "Comparing population patterns for genetic and morphological markers with uneven sample sizes. An example for the butterfly *Maniola jurtina*" *Methods Ecol Evol* (2014), 5, 834-843.

Examples

```
data(dataisl)
#Define groups of islands
memb<-c(2,3,5,7,5,3,1,1,2,5,1,3,1,1,5,2,2,1,2,4,1,3,1,5,2,1,7,6,1,1,1)
#Compute bidimensional representation for elements
pcoa<-cmdscale(recluster.dist(dataisl))
bar<-aggregate(pcoa~memb,FUN="mean")[,2:3]
# test if the variation has been significantly lost
recluster.test.dist(pcoa,bar,memb,perm=100)
```

treebut

Phylogenetic tree for the butterfly species included in dataisl dataset

Description

This phylogenetic tree has been created based on known phylogeny of butterflies at family and sub-family level and on COI sequences at genus and species level. Branch lengths have been calculated by Graphen method

Usage

```
data(treemod)
```

Details

A phylogenetic tree of butterfly species occurring on Western Mediterranean islands.

Author(s)

Gerard Talavera and Roger Vila

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

treemod	<i>Hypothetical phylogenetic tree for the virtual island faunas provided with the package recluster</i>
---------	---

Description

This phylogenetic tree has been created from the datamod dataset representing a series of virtual faunas in different sites

Usage

```
data(treemod)
```

Details

A phylogenetic tree of 31 species taken from 9 sites.

Author(s)

Gerard Talavera

References

Dapporto L., Ramazzotti M., Fattorini S., Talavera G., Vila R., Dennis R. "recluster: an unbiased clustering procedure for beta-diversity turnover" *Ecography* (2013), 36:1070-1075.

Index

- * **array**
 - recluster.dist, 21
- * **bootstrap**
 - recluster.identify.nodes, 28
- * **cluster**
 - recluster.boot, 17
 - recluster.cons, 19
 - recluster.expl, 22
 - recluster.expl.diss, 23
 - recluster.group.col, 27
 - recluster.multi, 31
 - recluster.node.strength, 32
 - recluster.procrustes, 40
 - recluster.region, 41
 - recluster.test.dist, 44
- * **configurations**
 - recluster.line, 30
 - recluster.rotate, 43
- * **datasets**
 - dataisl, 13
 - datamod, 13
 - multiboot, 14
 - treebut, 46
 - treemod, 47
- * **differentiation**
 - recluster.fst, 24
 - recluster.fst.pair, 25
- * **dissimilarity**
 - recluster.hist, 28
 - recluster.plot.matrix, 36
- * **hplot**
 - recluster.plot, 33
- * **package**
 - recluster, 14
- * **plot**
 - recluster.col, 18
 - recluster.plot.col, 34
 - recluster.plot.pie, 36
 - recluster.plot.sites.col, 38
- * **record attribution**
 - biodecrypt.cross, 5
 - biodecrypt.optimise, 8
 - biodecrypt.plot, 9
 - biodecrypt.wrap, 10
- biodecrypt, 3
- biodecrypt.cross, 5
- biodecrypt.optimise, 8
- biodecrypt.plot, 9
- biodecrypt.view, 10
- biodecrypt.wrap, 10
- dataisl, 13
- datamod, 13
- multiboot, 14
- reCluster (recluster), 14
- recluster, 14
- recluster.boot, 17
- recluster.col, 18
- recluster.cons, 19
- recluster.dist, 21
- recluster.expl, 22
- recluster.expl.diss, 23
- recluster.fst, 24
- recluster.fst.pair, 25
- recluster.group.col, 27
- recluster.hist, 28
- recluster.identify.nodes, 28
- recluster.line, 30
- recluster.multi, 31
- recluster.node.strength, 32
- recluster.plot, 33
- recluster.plot.col, 34
- recluster.plot.matrix, 36
- recluster.plot.pie, 36
- recluster.plot.sites.col, 38
- recluster.procrustes, 40

recluster.region, [41](#)
recluster.rotate, [43](#)
recluster.test.dist, [44](#)

treebut, [46](#)
treemod, [47](#)