Package: sidier (via r-universe)

September 1, 2024

Type Package

Title Substitution and Indel Distances to Infer Evolutionary Relationships

Version 4.1.0

Date 2021-06-25

Author A. Jesus Muñoz Pajares

Maintainer A.J. Muñoz-Pajares <ajesusmp@ugr.es>

Depends R (>= 3.5.0)

Imports ape, network, igraph, gridBase, grid, ggmap, ggplot2

Encoding UTF-8

Description Evolutionary reconstruction based on substitutions and insertion-deletion (indels) analyses in a distance-based framework.

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2021-06-25 11:50:02 UTC

Contents

`sidier-package` *SIDIER: Substitution and Indel Distances to Infer Evolutionary Relationships*

Description

Package for evolutionary reconstruction and bar code analyses based on substitutions and insertion-deletion (indels) in a distance-based framework.

Details

The DESCRIPTION file:

Package:	sidier
Type:	Package
Title:	Substitution and Indel Distances to Infer Evolutionary Relationships
Version:	4.1.0
Date:	2021-06-25
Author:	A. Jesus Muñoz Pajares
Maintainer:	A.J. Muñoz-Pajares <ajesusmp@ugr.es>
Depends:	R (>= 3.5.0)
Imports:	ape, network, igraph, gridBase, grid, ggmap, ggplot2
Encoding:	UTF-8
Description:	Evolutionary reconstruction based on substitutions and insertion-deletion (indels) analyses in a distance-based
License:	GPL-2

Index of help topics:

```
BARRIEL                 Indel distances following Barriel method
Example_Spatial.plot_Alignment
                        example alignment #1 (fasta format)
FIFTH                   Indel distances following the fifth state
                        rationale
FilterHaplo             Filter haplotypes by occurrence
FindHaplo               Find equal haplotypes
GetHaplo                Get sequences of unique haplotypes
HapPerPop               Returns the number of haplotypes per
                        population.
MCIC                    Modified Complex Indel Coding as distance
                        matrix
NINA.thr                No Isolated Nodes Allowed network
SIC                     Indel distances following the Simple Index
                        Coding method
alignExample            example alignment #1 ('DNAbin' class)
assign.whole.taxo       Get taxonomy described in sequence names
barcode.gap             Barcode gap identification
barcode.quality         Estimates of barcode quality
barcode.summary         Summary of the inter- and intraspecific
                        distances
colour.scheme           internal function for node colour scheme
compare.dist            Threshold to discriminate species comparing
                        intra- and interspecific distance distributions
distance.comb           Distance matrices combination
double.plot             Haplotype and population networks including
                        mutations and haplotype frequencies.
ex_BLAST                example BLAST output
ex_Coords               example coordinates
```

ex_alignment1	example alignment #1
filter.whole.taxo	Get consensus taxonomy
genbank.sp.names	Species names from genbank accessions
get.majority.taxo	Get majority taxonomy for a sequence
inter.intra.plot	Histogram of the intra- and interspecific distances
mergeNodes	Merges nodes showing distance values equal to zero
mutation.network	Haplotype network depiction including mutations
mutationSummary	Summary of observed mutations
nt.gap.comb	substitution and indel distance combinations
perc.thr	Percolation threshold network
pie.network	Population network depiction including haplotype frequencies
pop.dist	Distances among populations
rule	Threshold to discriminate species.
sidier-package	SIDIER: Substitution and Indel Distances to Infer Evolutionary Relationships
simplify.network	Network showing modules as nodes
simuEvolution	Simulate sequences evolution
single.network	Plot a network given a threshold
single.network.module	Get modules and network given a threshold
spatial.plot	spatial plot of populations
zero.thr	Zero distance networks

Functions can be classified according to the following groups:

- *Barcode analysis*: barcode.gap; barcode.quality; barcode.summary; compare.dist; genbank.sp.names; inter.intra.plot; rule.

- *Example files*: alignExample; Example_Spatial.plot_Alignment; ex_alignment1; ex_Coords.

- *Indel coding methods*: BARRIEL; FIFTH; MCIC; SIC.

- *Matrix/network manipulation*: distance.comb; mergeNodes; nt.gap.comb; simplify.network.

- *Network-from-distance methods*: NINA.thr; perc.thr; zero.thr; single.network; single.network.module.

- *Network visualization*: mutation.network; pie.network; double.plot; colour.scheme; spatial.plot.

- *Sequence/haplotype analysis*: FilterHaplo; FindHaplo; GetHaplo; HapPerPop; mutationSummary; pop.dist; simuEvolution.

Author(s)

A. Jesus Muñoz Pajares

Maintainer: A.J. Muñoz-Pajares <ajesusmp@ugr.es>

References

Muñoz-Pajares, A. J. (2013). SIDIER: substitution and indel distances to infer evolutionary relationships. Methods in Ecology and Evolution 4, 1195-1200. doi: 10.1111/2041-210X.12118

`alignExample` *example alignment #1 ('DNAbin' class)*

Description

object of class 'DNAbin' to test some function within this package

Usage

```
data(ex_alignment1)
```

Details

Because fasta file examples are not automatically loaded into R environment, 'ex_alignment1' function generates a fasta file (named Example_Spatial.plot_Alignment) that is stored as a 'DNAbin' object named alignExample

Author(s)

A. J. Muñoz-Pajares

See Also

`ex_alignment1`, `Example_Spatial.plot_Alignment`

Examples

```
# data(ex_alignment1) # this will read a fasta file with the name 'alignExample'
# alignExample
```

`assign.whole.taxo` *Get taxonomy described in sequence names*

Description

Assign taxonomy to every line in a BLAST output using the information provided in the name of the subject sequences (stitle)

Usage

```
assign.whole.taxo(BLAST)
```

Arguments

`BLAST`	data.frame containing the output of a BLAST analysis. The first column must be the name of the sequences matching the queries and must contain information on the taxonomy of the subject sequences. See details.

Details

The expected input data.frame must contain information about taxonomy in the first column. Additional information is accepted if separated by "|", but taxonomy must be the last bit of information. Taxonomical information must be provided for kingdom, phylum, class, order, family, genus, and species, each separated by ";" and identified by a letter as follows:

optionalTEXT|optionalTEXT|k__kingdomName;p__phylumName;c__className; o__orderName;f__familyName;g__gen

This is the typical format of sequence names in several databases. Thus a BLAST output using any of these databases will automatically produce the desired format.

Value

a data.frame contining all the information provided in the input data.frame and seven additional columns containing the name of kingdom, phylum, class, order, family, genus, and species for this sequence

Author(s)

A. J. Muñoz-Pajares

See Also

filter.whole.taxo, and get.majority.taxo

Examples

```
# data(ex_BLAST)
# TAXO <- assign.whole.taxo(ex_BLAST)
```

barcode.gap *Barcode gap identification*

Description

Identifies barcode gaps based on representing intra- and interspecific distances. Species above the 1:1 line are considered to show a barcode gap.

Usage

```
barcode.gap(summary=NULL,stat.intra="max",stat.inter="min",
xlab=NULL, ylab=NULL, legend=TRUE, lab.nodes="nogap")
```

Arguments

`summary` a list produced by `barcode.summary`. From this list, the maximum intraspecific and the minimum interspecific distances per species are represented. To use any other intra- and interspecific distance, use the "inter" and "intra" options.

`stat.intra` a string, the inter-specific statistic used to estimate the quotient interspecific/intraspecific. Accepted values are "max", "min", "median", and "mean"

`stat.inter` a string, the inter-specific statistic used to estimate the quotient interspecific/intraspecific. Accepted values are "max", "min", "median", and "mean".

`xlab` a string, the x-axis label

`ylab` a string, the y-axis label

`legend` a logic, to show information about species showing and lacking barcode within the plot

`lab.nodes` a string to select the name of species to be represented in the plot: "gap" to represent species showing barcode gap; "nogap" to represent species lacking barcode gap; "all" for representing all species names. Other value will represent no names.

Value

A list with two elements:

`no.barcode.gap` a matrix containing the name of the species lacking barcode gap and their mean intra- and interspecific distances.

`barcode.gap` a matrix containing the name of the species showing barcode gap and their mean intra- and interspecific distances.

Author(s)

A.J. Muñoz-Pajares

See Also

`barcode.summary`

Examples

```
# my.dist<-matrix(abs(rnorm(100)),ncol=10,
# dimnames=list(paste("sp",rep(1:5,2),sep=""),
# paste("sp",rep(1:5,2),sep="")))
# my.dist<-as.matrix(as.dist(my.dist))
# sum<-barcode.summary(my.dist)
# barcode.gap(sum)
```

`barcode.quality` *Estimates of barcode quality*

Description

Provides several estimates of the quality of a barcode classification, comparing network modules with attributed species names

Usage

```
barcode.quality(dismat=NA,threshold=NA,refer2max=FALSE,save.file=FALSE,
modFileName="Modules_summary.txt",verbose=FALSE,output="list")
```

Arguments

`dismat`	a matrix containing the pairwise genetic distances between individual sequences
`threshold`	a numeric between 0 and 1, is the value of the maximum distance to be represented as a link in the network
`refer2max`	a logic, "TRUE" to refer the threshold value to the maximum distance in the input matrix (e.g., a value of 0.32 will represent a link between nodes showing distances equal or lower than 32% of the maximum distance found in the distance matrix). "FALSE" to refer the threshold to a specific value (e.g., a value of 0.32 will represent a link between nodes showing distances equal or lower than 0.32, regardless the maximum distance found in the distance matrix).
`save.file`	a logic, "TRUE" to save the summary of network modules, attributing every individual to a module.
`modFileName`	if save.file=TRUE, a string: the name of the file containing the summary of network modules.
`verbose`	a logic, "TRUE" to obtain a complete report of the quality estimation (see details).
`output`	if verbose=TRUE, a string controlling the type of object produced for the output, being either "matrix" or "list".

Details

This function assumes that the species names reflect the "real" taxonomic status and compare these names with the modules obtained in the network analysis. The quality is evaluated using different estimators:

$$Accuracy = \frac{T_{+} + T_{-}}{T_{+} + T_{-} + F_{+} + F_{-}}$$

$$Precision = \frac{T_{+}}{T_{+} + F_{+}}$$

$$Fscore = \frac{T_{+}}{T_{+} + F_{+} + F_{-}}$$

$$Qvalue = \frac{1}{N}\sum_{1}^{N}\frac{S_{link}}{S_{all}+S_{unlink}}$$

where T+ is the number of true positives (number of sequences with the same species name and classified in the same module); T- is the number of true negatives (number of sequences with different species name and classified in different modules); F+ represents false positive (number of sequences with different species name classified in the same module); F- is the number of false negative (number of sequences with the same species name classified in different modules); N is the number of nodes in the network, Slink is the number of nodes of the same species connected to the node i; Sunlink is the number of nodes of the same species belonging to a different module; and Sall is the number of all possible connections to other nodes of the same species.

Value

If verbose is set to "FALSE", a matrix with the estimators of the barcode quality. If verbose is set to "TRUE", either a matrix or a list (depending on the output option selected) containing the following elements:

`Number.of.modules`
Number of modules found in the network analysis.

`Number.of.species.per.module`
A matrix containing: The number of species classified in only one module (N.sp.mod.1); the maximum number of species found in a module (N.sp.mod.MAX); and the mean number of species found per module (N.sp.mod.MED).

`Number.of.species`
The number of species defined for the analysis.

`Number.of.modules.per.species`
A matrix containing: The number of modules composed of only one species (N.mod.sp.1); the maximum number of modules containing the same species; the mean number of modules containing the same species.

`Number.of.modules.fitting.defined.species`
The number of modules containing only one species but all the individuals of this species.

`Quality.estimates`
A matrix containing the Qvalue, Accuracy, Precision and Fscore of the barcode classification.

Author(s)

A.J. Muñoz-Pajares

Examples

```
# my.dist<-matrix(abs(rnorm(100)),ncol=10,
# dimnames=list(paste("sp",rep(1:5,2),sep=""),
# paste("sp",rep(1:5,2),sep="")))
# my.dist<-as.matrix(as.dist(my.dist))
#
# barcode.quality(dismat=my.dist,threshold=0.2,refer2max=FALSE,save.file=TRUE,
# modFileName="Modules_summary.txt",verbose=FALSE,output="list")
```

`barcode.summary` *Summary of the inter- and intraspecific distances*

Description

For every species, provides the minimum, maximum, mean and median values of inter- and intraspecific distances.

Usage

```
barcode.summary(dismat=NULL,save.distances=FALSE,folder.name="distance_matrices")
```

Arguments

`dismat` a symmetric matrix containing the pairwise genetic distances between individual sequences

`save.distances` a logic, "TRUE" to save the pairwise distances estimated per species (one file per species)

`folder.name` a string, if save.distance=TRUE, the name of the folder to save distances

Value

A list with two elements:

`Intraspecific` a matrix containing information about the intraspecific distances.

`Interspecific` a matrix containing information about the interspecific distances.

In both cases, the information provided is the minimum, maximum, median, mean, first and third quartile values.

Author(s)

A.J. Muñoz-Pajares

See Also

`barcode.gap`

Examples

```
# my.dist<-matrix(abs(rnorm(100)),ncol=10,
# dimnames=list(paste("sp",rep(1:5,2),sep=""),
# paste("sp",rep(1:5,2),sep="")))
# my.dist<-as.matrix(as.dist(my.dist))
# barcode.summary(my.dist)
```

BARRIEL *Indel distances following Barriel method*

Description

This function codifies gapped positions in a sequence alignment following the rationale of the method described by Barriel (1994). Based on the yielded indel coding matrix, this function also computes a pairwise indel distance matrix.

Usage

```
BARRIEL(inputFile = NA, align = NA, saveFile = TRUE,
outnameDist = paste(inputFile, "IndelDistanceBarriel.txt",
sep = "_"), outnameCode = paste(inputFile, "Barriel_coding.txt",
 sep = "_"), addExtremes = FALSE)
```

Arguments

`inputFile` the name of the fasta file to be analysed. Alternatively you can provide the name of a "DNAbin" class alignment stored in memory using the "align" option.

`align` the name of the "DNAbin" alignment to be analysed. See "?read.dna" in the ape package for details about reading alignments. Alternatively you can provide the name of the file containing the alignment in fasta format using the "inputFile" option.

`saveFile` a logical; if TRUE (default), it produces two output text files containing the distance matrix and the codified indel positions.

`outnameDist` if "saveFile" is set to TRUE (default), contains the name of the distance output file.

`outnameCode` if "saveFile" is set to TRUE (default), contains the name of the indel coding output file.

`addExtremes` a logical; if TRUE, additional nucleotide sites are included in both extremes of the alignment. This will allow estimating distances for alignments showing gaps in terminal positions, but see Details.

Details

It is recommended to estimate this distance matrix using only the unique sequences in the alignment. Repeated sequences increase computation time but do not provide additional information (because they produce duplicated rows and columns in the final distance matrix).

It is of critical importance to correctly identify indels homology in the provided alignment. For this reason, addExtremes is set to false by default, and computation may not be done unless flanking regions are homologous.

Value

A list with two elements:

`indel coding matrix`
: Describes the initial and final site of each gap and its presence or absence per sequence.

`distance matrix`
: Contains genetic distances based on comparing indel presence/absence between sequences.

Author(s)

A.J. Muñoz-Pajares

References

Barriel, V., 1994. Molecular phylogenies and how to code insertion/ deletion events. Life Sci. 317, 693-701, cited and described by Simmons, M.P., Müller, K. & Norton, A.P. (2007) The relative performance of indel-coding methods in simulations. Molecular Phylogenetics and Evolution, 44, 724–740.

See Also

`MCIC`, `SIC`, `FIFTH`

Examples

```
# cat(">Population1_sequence1",
# "A-AGGGTC-CT---G",
# ">Population1_sequence2",
# "TAA---TCGCT---G",
# ">Population1_sequence3",
# "TAAGGGTCGCT---G",
# ">Population1_sequence4",
# "TAA---TCGCT---G",
# ">Population2_sequence1",
# "TTACGGTCG---TTG",
# ">Population2_sequence2",
# "TAA---TCG---TTG",
# ">Population2_sequence3",
# "TAA---TCGCTATTG",
# ">Population2_sequence4",
# "TTACGGTCG---TTG",
# ">Population3_sequence1",
# "TTA---TCG---TAG",
# ">Population3_sequence2",
# "TTA---TCG---TAG",
# ">Population3_sequence3",
# "TTA---TCG---TAG",
# ">Population3_sequence4",
# "TTA---TCG---TAG",
#      file = "ex3.fas", sep = "\n")
#
# library(ape)
# BARRIEL(align=read.dna("ex3.fas",format="fasta"), saveFile = FALSE)
#
# # Analysing the same dataset, but using only unique sequences:
# uni<-GetHaplo(inputFile="ex3.fas",saveFile=FALSE)
# BARRIEL(align=uni, saveFile = FALSE)
```

colour.scheme *internal function for node colour scheme*

Description

This function is called during network representations to set node colours. If the number of colours defined by user do not match with the number of elements, the algorithm provide a default set of colours.

Usage

```
colour.scheme(def=NA, N=NA, colors=c("green2","red","yellow","blue","DarkOrchid1",
"gray51","chocolate","cyan4","saddle brown","aquamarine","chartreuse","chocolate1",
"DarkOrchid3","gray18","gold","DarkOrchid4","green4","gray29", "sienna3","tan1","blue4",
"limegreen","gray73","bisque3","deeppink","red4","OliveDrab4","gray95", "salmon",
"DeepPink4","green yellow","gray4","hot pink","pink2","dark orange","gold3"))
```

Arguments

def	a vector containing the set of colours defined by user
N	a numeric representing the number of elements to be coloured
colors	a vector with default colours to be used if 'def' is different from 'N'

Details

If the number of elements is higher than the number of colours (35 by default), colours are randomly selected.

Value

a vector of strings representing 'N' colours

Author(s)

A. J. Muñoz-Pajares

Examples

```
# colour.scheme(def=c("blue","red"),N=4)
# Colors<-colour.scheme(def=c("blue","red"),N=4,colors=c("black","gray33","gray66","orange","red"))
# plot(c(1:4),col=Colors,pch=16)
#
# #Given 10 individuals classified into three groups,
# #this will provide the colour for each individual:
# group<-c(1,1,1,2,2,2,1,2,3,3) # defining groups
# colour.scheme(N=length(unique(group)))[group]
#
```

compare.dist *Threshold to discriminate species comparing intra- and interspecific distance distributions*

Description

This function implements the Lefebure's method to quantify the overlap between two distributions and to determine the best threshold value to discriminate them.

Usage

```
compare.dist(distr1=NULL,distr2=NULL,N=50,
normalize=TRUE,main=NA,col1="gray",col2="black",
col.border1="gray",col.border2="black",
col.line1="gray",col.line2="black",
Ylab=c("Abundance","Abundance","Success"),
Xlab=c("data1","data2","Threshold"))
```

Arguments

distr1	a matrix containing the pairwise genetic distances between individual sequences
distr2	a vector containing the minimum and maximum value in the x-axis
N	a numeric, the number of categories for the x-axis.
normalize	a logic, "TRUE" to display percentage and "FALSE" for number of occurrences in the Y axis.
main	a vector with two elements containing the main titles of both plots.
col1	a string, the color to fill the histogram for distribution 1
col2	a string, the color to fill the histogram for distribution 2
col.border1	a string, the color for the border around the histogram for distribution 1
col.border2	a string, the color for the border around the histogram for distribution 2
col.line1	a string, the color for the line representing distribution 1
col.line2	a string, the color for the line representing distribution 2

`Ylab`	a three strings vector containing the labels of Y axes for the three plots to be represented.
`Xlab`	a three strings vector containing the labels of X axes for the three plots to be represented

Value

The estimated threshold and its success of identification.

Author(s)

A.J. Muñoz-Pajares

References

Lefébure T, Douady CJ, Gouy M, Gibert J (2006). Relationship between morphological taxonomy and molecular divergence within Crustacea: Proposal of a molecular threshold to help species delimitation. Mol Phylogenet Evol 40: 435–447.

Examples

```
# ## Weak overlap
# intra<-rnorm(mean=0.08,sd=0.04,n=100)
# inter<-rnorm(mean=0.38,sd=0.10,n=1000)
# intra[intra<0]<-0
# inter[inter<0]<-0
# compare.dist(distr1=intra,distr2=inter,N=50)
#
# # Strong overlap
# distr1<-rnorm(5000,mean=0.25,sd=0.070)
# distr2<-rnorm(5000,mean=0.31,sd=0.075)
# N<-50
# compare.dist(distr1,distr2,N)
#
```

`distance.comb` *Distance matrices combination*

Description

This function allows combining distance matrices. The weight of each matrix must be defined by user.

Usage

```
distance.comb(matrices = NA, alphas = NA, method = "Corrected",
saveFile = TRUE, na.rm = FALSE)
```

Arguments

`matrices`	a vector of strings containing the names of the matrices to be combined
`alphas`	a vector of numerics containing the weight of each matrix. By default, combination is done giving the same same weight to all matrices.
`method`	a string defining whether each distance matrix must be divided by its maximum value before the combination ("Corrected") or not ("Uncorrected"). Consequently, if the "Corrected" method is chosen, both matrices will range between 0 and 1 before being combined.
`saveFile`	a logical; if TRUE (default), the output matrix is saved a text file.
`na.rm`	a logical; if TRUE, missing values are removed before the computation proceeds.

Details

Matrices do not require to contain rows in the same order. The algorithm will search and combine rows of matrices with identical names. If row names of any input matrix are not defined, the algorithm will combine rows in order.

Value

A matrix containing the weighted combination of the original matrices

Author(s)

A. J. Muñoz-Pajares

See Also

nt.gap.comb

Examples

```
# mat1<-matrix(rep(1,16),ncol=4)
# mat2<-matrix(rep(2,16),ncol=4)
# mat3<-matrix(rep(3,16),ncol=4)
# mat4<-matrix(rep(6,16),ncol=4)
# mat5<-matrix(c(rep(1,4),rep(3,4),rep(1,4),rep(1,4)),ncol=4)
# mat6<-matrix(c(rep(1,4),rep(4,4),rep(4,4),rep(4,4)),ncol=4)
# mat7<-mat6
# colnames(mat5)<-c("a","b","c","d")
# colnames(mat6)<-c("b","a","c","d")
# row.names(mat5)<-c("a","b","c","d")
# row.names(mat6)<-c("b","a","c","d")
#
# # Matrices with information about the same elements (a-d)
# # but in different order, are automatically
# # sorted before combination....
#
# distance.comb(matrices=c("mat5","mat6"),saveFile=FALSE,method="Uncorrected")
```

```
#
# # ... but this is not possible if any of the matrices lack row names:
#
# distance.comb(matrices=c("mat5","mat7"),saveFile=FALSE,method="Uncorrected")
#
# # More examples:
# distance.comb(matrices=c("mat1","mat2","mat3","mat4"),alphas=rep(0.25,4),
# saveFile=FALSE,method="Uncorrected")
#
# distance.comb(matrices=c("mat1","mat2","mat3","mat4"),alphas=rep(0.25,4),
# saveFile=FALSE,method="Corrected")
#
# distance.comb(matrices=c("mat1","mat2","mat3","mat4"),alphas=c(0.66,0.33,0,0),
# saveFile=FALSE,method="Uncorrected")
#
# distance.comb(matrices=c("mat1","mat2","mat3","mat6"),alphas=c(0.66,0.33,0,0),
# saveFile=FALSE,method="Uncorrected")
```

double.plot *Haplotype and population networks including mutations and haplotype frequencies.*

Description

This function makes a double plot by dividing the active device into two parts. The left part is used to represent the input alignment as a haplotypic network displaying mutations. The right part is used to represent the same input alignment as a population network displaying nodes as pie charts.

Usage

```
double.plot(align = NA, indel.method = "MCIC",
substitution.model = "raw",pairwise.deletion = TRUE,
network.method.mut = "percolation", network.method.pie = "percolation",
range = seq(0, 1, 0.01), addExtremes = FALSE, alpha.mut = "info",
alpha.pie = "info", combination.method.mut = "Corrected",
combination.method.pie = "Corrected", na.rm.row.col.mut = FALSE,
na.rm.row.col.pie = FALSE, save.distance.mut = FALSE,
save.distance.name.mut = "DistanceMatrix_threshold_Mutations.txt",
save.distance.pie = FALSE, save.distance.name.pie =
"DistanceMatrix_threshold_Pies.txt", modules=FALSE,
modules.col=NA, bgcol = NA, label.col.mut = "black",
label.col.pie = "black", label.mut = NA,label.pie = NA,
label.sub.str.mut = NA,label.sub.str.pie = NA, colInd = "red",
colSust = "black", lwd.mut = 1, InScale=1, SuScale=1, lwd.edge = 1.5,
cex.mut = 1, cex.label.mut = 1, cex.label.pie = 1, cex.vertex = 1,
main=c("Haplotypes","Populations"),
NameIniPopulations = NA, NameEndPopulations = NA, NameIniHaplotypes = NA,
NameEndHaplotypes = NA, cex.pie = 1, HaplosNames = NA, offset.label = 1.5)
```

Arguments

`align` a 'DNAbin' object; the alignment to be analysed. See "read.dna" in the ape package for details about reading alignments.

`indel.method` a sting; the method to define indel events in your alignment. The available methods are:

-"MCIC": (Default) Estimates indel events following the rationale of the Modified Complex Indel Coding (Muller, 2006).

-"SIC": Estimates indel events following the rationale of Simmons and Ochoterrena (2001).

-"FIFTH": Estimates indel events following the rationale of the fifth state: each gap within the alignment is treated as an independent mutation event.

-"BARRIEL": Estimates indel events following the rationale of Barriel (1994): singleton gaps are not taken into account.

`substitution.model` a string; the substitution evolutionary model to estimate the distance matrix. By default is set to "raw" and estimates the pairwise proportion of variant sites. See the evolutionary models available using ?dist.dna from the ape package.

`pairwise.deletion` a logical; if TRUE (default) substitutions found in regions being a gap in other sequences will account for the distance matrix. If FALSE, sites being a gap in at least one sequence will be removed before distance estimation.

`network.method.mut` a string; the method to build the haplotypic network. The available methods are:

-"percolation": computes a network using the percolation network method following Rozenfeld et al. (2008). See ?perc.thr for details

-"NINA": computes a network using the No Isolation Nodes Allowed method. See ?NINA.thr for details.

-"zero": computes a network connecting all nodes showing distances equal to zero. See ?zero.thr for details.

`network.method.pie` a string; the method to build the population network. The available methods are the same than for 'network.method.mut'

`range` a numeric vector between 0 and 1, is the range of thresholds (referred to the maximum distance in the input matrix) to be screened (by default, 101 values from 0 to 1). This option is used for "percolation" and "NINA" network methods and ignored for "zero" method.

`addExtremes` a logical; if TRUE, additional nucleotide sites are included in both extremes of the alignment. This will allow estimating distances for alignments showing gaps in terminal positions. This option is used for "SIC", "FIFTH" and "BARRIEL" indel methods and ignored for "MCIC" method.

`alpha.mut` a numeric between 0 and 1, is the weight given to the indel genetic distance matrix in the combination to represent the haplotypic network. By definition, the weight of the substitution genetic matrix is the complementary value (i.e., 1-alpha). The value "info" (default) will use the proportion of informative substitutions per informative indel event as weight. It is also possible to define multiple weights to estimate different combinations.

`alpha.pie` a numeric between 0 and 1, is the weight given to the indel genetic distance matrix in the combination to represent the population network. By definition, the weight of the substitution genetic matrix is the complementary value (i.e., 1-alpha). The value "info" (default) will use the proportion of informative substitutions per informative indel event as weight. It is also possible to define multiple weights to estimate different combinations.

`combination.method.mut`
a string defining whether each distance matrix must be divided by its maximum value before the combination ("Corrected") or not ("Uncorrected"). Consequently, if the "Corrected" method is chosen, both matrices will range between 0 and 1 before being combined. This option affects the haplotype network depiction.

`combination.method.pie`
a string defining whether each distance matrix must be divided by its maximum value before the combination ("Corrected") or not ("Uncorrected"). Consequently, if the "Corrected" method is chosen, both matrices will range between 0 and 1 before being combined. This option affects the population network depiction.

`na.rm.row.col.mut`
a logical; if TRUE, distance matrix missing values are removed.

`na.rm.row.col.pie`
a logical; if TRUE, distance matrix missing values are removed.

`save.distance.mut`
a logical; if TRUE, the distance matrix used to build the haplotypic network will be saved as a file.

`save.distance.name.mut`
a string; if save.distance.mut=TRUE, it defines the name of the file to be saved.

`save.distance.pie`
a logical; if TRUE, the distance matrix used to build the population network will be saved as a file.

`save.distance.name.pie`
a string; if save.distance.pie=TRUE, it defines the name of the file to be saved.

`modules` a logical. If TRUE, nodes colours are set according to modules in the network of haplotypes.

`modules.col` (if modules=TRUE) a vector of strings defining the colour of nodes belonging to different modules in the network. If 'NA' (or there are less colours than haplotypes), colours are automatically selected.

`bgcol` a vector of strings; the colour of the background for each node in the haplotypic network. The same colours will be used to represent haplotypes in the population network. If set to 'NA' (default), colours are automatically defined.

`label.col.mut` a vector of strings; the colour of labels for each node in the haplotypic network. Can be equal for all nodes (if only one colour is defined) or customized (if several colours are defined).

`label.col.pie` a vector of strings; the colour of labels for each node in the population network. Can be equal for all nodes (if only one colour is defined) or customized (if several colours are defined).

`label.mut` a vector of strings; labels for each node in the haplotypic network. By default are the sequence names. (See "substr" function in base package to automatically reduce name lengths)

`label.pie` a vector of strings; labels for each node in the population network. By default are the sequence names. (See "substr" function in base package to automatically reduce name lengths)

`label.sub.str.mut`
a vector of two numerics; if node labels are a substring of sequence names, these two numbers represent the initial and final character of the string to be represented in the haplotypic network. See Example for details.

`label.sub.str.pie`
a vector of two numerics; if node labels are a substring of sequence names, these two numbers represent the initial and final character of the string to be represented in the population network. See Example for details.

`colInd` a strings; the color used to represent indels.

`colSust` a strings; the color used to represent substitutions.

`lwd.mut` a numeric; the width of the line (marks perpendicular to the edge line) representing mutations (1 by default).

`InScale` a numeric; the number of indels each mark represents. By default is set to 1 (that is, 1 mark= 1 indel). If set to 10, then 1 mark=10 indels. In that case, if there are 25 indels in a given edge it is represented by three marks (being one of them half width than the other two).

`SuScale` a numeric; the number of substitutions each mark represents. By default is set to 1 (that is, 1 mark= 1 substitution). If set to 10, then 1 mark=10 substitutions In that case, if there are 25 substitutions in a given edge it is represented by three marks (being one of them half width than the other two).

`lwd.edge` a numeric; the width of the edge linking nodes (1.5 by default).

`cex.mut` a numeric; the length of the line (perpendicular to the edge line) representing mutations (1 by default).

`cex.label.mut` a numeric; the size of the node labels in the haplotypic network.

`cex.label.pie` a numeric; the size of the node labels in the population network.

`cex.vertex` a numeric; the size of the nodes in the haplotypic network.

`main` a vector with two elements defining the title for the left and right plots, respectively. Alternatively, may be set to "summary" to display the main options selected for representing the networks. Finally, if set to "", the algorithm will show no title for any network.

`NameIniPopulations`
a numeric; Position of the initial character of population names within sequence names. If 'NA' (default), it is set to 1.

`NameEndPopulations`
a numeric; Position of the last character of population names within sequence names. If 'NA' (default), it is set to the first "_" character in the sequences name.

`NameIniHaplotypes`
: a numeric; Position of the initial character of haplotype names within sequence names. If 'NA' (default), haplotype names are defined by the algorithm and the value is set accordingly.

`NameEndHaplotypes`
: a numeric; Position of the last character of haplotype names within sequence names. If 'NA' (default), haplotype names are defined by the algorithm and the value is set accordingly.

`cex.pie` a numeric; the size of the nodes in the population network.

`HaplosNames` a sting; the name of the haplotypes (if different from default: H1...Hn)

`offset.label` a numeric, the separation between node and label.

Author(s)

A. J. Muñoz-Pajares

References

Barriel, V., 1994. Molecular phylogenies and how to code insertion/ deletion events. Life Sci. 317, 693-701, cited and described by Simmons, M.P., Müller, K. & Norton, A.P. (2007) The relative performance of indel-coding methods in simulations. Molecular Phylogenetics and Evolution, 44, 724–740.

Muller K. (2006). Incorporating information from length-mutational events into phylogenetic analysis. Molecular Phylogenetics and Evolution, 38, 667-676.

Rozenfeld AF, Arnaud-Haond S, Hernandez-Garcia E, Eguiluz VM, Serrao EA, Duarte CM. (2008). Network analysis identifies weak and strong links in a metapopulation system. Proceedings of the National Academy of Sciences,105, 18824-18829.

Simmons, M.P. & Ochoterena, H. (2000). Gaps as Characters in Sequence-Based Phylogenetic Analyses. Systematic Biology, 49, 369-381.

See Also

`mutation.network`, `pie.network`

Examples

```
# cat(">Population1_sequence1",
# "TTATAAAATCTA----TAGC",
# ">Population1_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population1_sequence3",
# "TTATAAAAATTA----TAGC",
# ">Population1_sequence4",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence1",
# "TTAT----TCGAGGGGTAGC",
# ">Population2_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence3",
# "TTATAAAA--------TAGC",
# ">Population2_sequence4",
# "TTAT----TCGAGGGGTAGC",
# ">Population3_sequence1",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence2",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence3",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence4",
# "TTAT----TCGA----TAGC",
#      file = "ex2.fas", sep = "\n")
# library(ape)

# Double plot (computation time may exceed 5 seconds)
#double.plot(align=read.dna(file="ex2.fas",format="fasta"))

# Other options
#data(ex_alignment1) # this will read a fasta file with the name 'alignExample'
#double.plot(alignExample,label.sub.str.mut=c(7,9))
#double.plot(alignExample,label.sub.str.mut=c(7,9),InScale=3,SuScale=2,lwd.mut=1.5)
```

Example_Spatial.plot_Alignment

example alignment #1 (fasta format)

Description

fasta sequences alignment to test some function within this package

Usage

```
data(ex_alignment1)
```

Details

Because fasta file examples are not automatically loaded into R environment, 'ex_alignment1' function generates a fasta file (named Example_Spatial.plot_Alignment) that is stored as a 'DNAbin' object named alignExample

Author(s)

A. J. Muñoz-Pajares

See Also

ex_alignment1, alignExample

Examples

```
# data(ex_alignment1) # this will read a fasta file with the name 'alignExample'
# alignExample
```

ex_alignment1 *example alignment #1*

Description

This function will generate an example alignment to test some function within this package

Usage

```
data(ex_alignment1)
```

Details

Because fasta file examples are not automatically loaded into R environment, 'ex_alignment1' function generates a fasta file (named Example_Spatial.plot_Alignment) that is stored as a 'DNAbin' object named alignExample

Author(s)

A. J. Muñoz-Pajares

See Also

alignExample, Example_Spatial.plot_Alignment

Examples

```
# data(ex_alignment1) # this will read a fasta file with the name 'alignExample'
# alignExample
```

ex_BLAST *example BLAST output*

Description

BLAST output

Usage

```
data(ex_BLAST)
```

Value

a four columns data.frame containing the qseqid (as character, not as factor!), evalue, percentage of identity and query coverage, respectively.

Author(s)

A. J. Muñoz-Pajares

Examples

```
# data(ex_BLAST)
# ex_BLAST
```

ex_Coords *example coordinates*

Description

Geographic coordinates of populatons to test some function within this package

Usage

```
data(ex_Coords)
```

Value

a three columns matrix containing the population name, longitude and latitude, respectively.

Author(s)

A. J. Muñoz-Pajares

Examples

```
# data(ex_Coords)
# ex_Coords
```

FIFTH *Indel distances following the fifth state rationale*

Description

This function computes an indel distance matrix following the rationale of the fifth state. For that, each gap within the alignment is treated as an independent mutation event.

Usage

```
FIFTH(inputFile = NA, align = NA, saveFile = TRUE,
outname = paste(inputFile,"IndelDistanceFifthState.txt",
 sep = "_"), addExtremes = FALSE)
```

Arguments

`inputFile`	the name of the fasta file to be analysed. Alternatively you can provide the name of a "DNAbin" class alignment stored in memory using the "align" option.
`align`	the name of the alignment to be analysed. See "read.dna" in ape package for details about reading alignments. Alternatively you can provide the name of the file containing the alignment in fasta format using the "inputFile" option.
`saveFile`	a logical; if TRUE (default), it produces an output text file containing the resulting distance matrix.
`outname`	if "saveFile" is set to TRUE (default), contains the name of the output file.
`addExtremes`	a logical; if TRUE, additional nucleotide sites are included in both extremes of the alignment. This will allow estimating distances for alignments showing gaps in terminal positions.

Details

It is recommended to estimate this distance matrix using only the unique sequences in the alignment. Repeated sequences increase computation time but do not provide additional information (because they produce duplicated rows and columns in the final distance matrix).

It is of critical importance to correctly identify indels homology in the provided alignment. For this reason, addExtremes is set to false by default, and computation may not be done unless flanking regions were homologous.

Value

A matrix containing the genetic distances estimated as indels pairwise differences.

Author(s)

A. J. Muñoz-Pajares

See Also

BARRIEL, MCIC, SIC

Examples

```
# # This will generate an example file in your working directory:
# cat(">Population1_sequence1",
# "A-AGGGTC-CT---G",
# ">Population1_sequence2",
# "TAA---TCGCT---G",
# ">Population1_sequence3",
# "TAAGGGTCGCT---G",
# ">Population1_sequence4",
# "TAA---TCGCT---G",
# ">Population2_sequence1",
# "TTACGGTCG---TTG",
# ">Population2_sequence2",
# "TAA---TCG---TTG",
# ">Population2_sequence3",
# "TAA---TCGCTATTG",
# ">Population2_sequence4",
# "TTACGGTCG---TTG",
# ">Population3_sequence1",
# "TTA---TCG---TAG",
# ">Population3_sequence2",
# "TTA---TCG---TAG",
# ">Population3_sequence3",
# "TTA---TCG---TAG",
# ">Population3_sequence4",
# "TTA---TCG---TAG",
#      file = "ex3.fas", sep = "\n")
#
# # Reading the alignment directly from file and saving no output file:
# library(ape)
# FIFTH (align=read.dna("ex3.fas",format="fasta"), saveFile = FALSE)
#
# # Analysing the same dataset, but using only unique sequences:
# uni<-GetHaplo(inputFile="ex3.fas",saveFile=FALSE)
# FIFTH (align=uni, saveFile = FALSE)
```

---

filter.whole.taxo   *Get consensus taxonomy*

Description

Given the taxonomy of multiple BLAST hits for a query sequence, provides the most likely tanoxomy for the query sequence taking BLAST percentage of identity values into account

Usage

```
filter.whole.taxo(whole.taxo)
```

Arguments

`whole.taxo`	data.frame containing BLAST results and taxonomy information. Can be produced by `assign.whole.taxo`

Details

The expected input data.frame must contain information about BLAST hits (particularly, a "pident" column with the percentage of identity) and seven additional columns containing the name of kingdom, phylum, class, order, family, genus, and species for every subject sequence.

Depending on the "pident" value, taxonomy for the subject sequence will be retained until species (if pident>=97), genus (97>pident>=90), family (90>pident>=85), order (85>pident>=80), family (80>pident>=75), or class (75>pident>=0). For taxonomic levels showing pident lower that these thresholds, "low_pident" is returned.

Value

a data.frame contining all the information provided in the input data.frame and seven additional columns containing the name of kingdom, phylum, class, order, family, genus, and species for this sequence after filtering by BLAST percentage of identity.

Author(s)

A. J. Muñoz-Pajares

See Also

`get.majority.taxo`,`assign.whole.taxo`

Examples

```
#data(ex_BLAST)
#TAXO <- assign.whole.taxo(ex_BLAST)
#FILT_TAXO <- filter.whole.taxo(TAXO)
```

FilterHaplo *Filter haplotypes by occurrence*

Description

Provides a subset of a original alignment composed only of haplotypes showing the range of occurrences provided

Usage

```
FilterHaplo(inputFile=NA, align=NA, Nmin=0, Nmax=NULL,
 saveFile=FALSE, outname="FilterHaplo.txt")
```

Arguments

`inputFile` the name of a sequence alignment file in fasta format to be analysed. Alternatively you can provide the name of a "DNAbin" class alignment stored in memory using the "align" option.

`align` the name of the "DNAbin" alignment to be analysed. See "?read.dna" in the ape package for details about reading alignments. Alternatively you can provide the name of the file containing the alignment in fasta format using the "inputFile" option.

`Nmin` Minimum occurrence of an haplotype to be included in the subset

`Nmax` Maximum occurrence of an haplotype to be included in the subset

`saveFile` a logical; if TRUE (default), function output is saved as a text file in fasta format

`outname` if "saveFile" is set to TRUE (default), contains the name of the output file.

Author(s)

A. J. Muñoz-Pajares

Examples

```
# cat(">Population1_sequence1",
# "TTATAAAATCTA----TAGC",
# ">Population1_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population1_sequence3",
# "TTATAAAAATTA----TAGC",
# ">Population1_sequence4",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence1",
# "TTAT----TCGA----TAGC",
# ">Population2_sequence2",
# "TTAT----TCGA----TAGC",
# ">Population2_sequence3",
# "TTAT----TCGA----TAGC",
# ">Population2_sequence4",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence1",
# "TTAT----TCGAGGGGTAGC",
# ">Population3_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population3_sequence3",
# "TTATAAAA--------TAGC",
# ">Population3_sequence4",
# "TTAT----TCGAGGGGTAGC",
#      file = "ex2.fas", sep = "\n")
# library(ape)
# example<-read.dna(file="ex2.fas",format="fasta")
#
# # Exclude unique haplotypes
# FilterHaplo(align=example,Nmin=2)
#
# # Include only unique haplotypes
# FilterHaplo(align=example,Nmax=1)
#
# # Filter haplotypes appearing between 2 and 4 times
# FilterHaplo(align=example,Nmax=4,Nmin=2)
```

`FindHaplo` *Find equal haplotypes*

Description

This function assigns the same name to equal haplotypes in a sequence alignment.

Usage

```
FindHaplo(inputFile = NA, align = NA,
saveFile = TRUE, outname = "FindHaplo.txt")
```

Arguments

`inputFile`	the name of a sequence alignment file in fasta format to be analysed. Alternatively you can provide the name of a "DNAbin" class alignment stored in memory using the "align" option.
`align`	the name of the "DNAbin" alignment to be analysed. See "?read.dna" in the ape package for details about reading alignments. Alternatively you can provide the name of the file containing the alignment in fasta format using the "inputFile" option.
`saveFile`	a logical; if TRUE (default), function output is saved as a text file.
`outname`	if "saveFile" is set to TRUE (default), contains the name of the output file ("FindHaplo.txt" by default).

Details

The algorithm identifies identical sequences even if they are wrongly aligned (see example).

Value

A two columns matrix containing the original sequence name and the haplotype name assigned to each sequence in the input alignment.

Author(s)

A. J. Muñoz-Pajares

See Also

GetHaplo, HapPerPop

Examples

```
#
#  #generating an alignment file:
# cat(”>Population1_sequence1”,
# ”TTATAGGTAGCTTCGATATTG”,
# ”>Population2_sequence1”,
# ”TTA---GTAGCTTCGAAATTG”,
# ”>Population3_sequence1”,
# ”TTA---GTA---TCG---TAG”,
# ”>Population4_sequence1”,
# ”TTATAGGTA---TCG---TTG”,
# ”>Population5_sequence1”,
# ”TTA------------AAATTG”,
# file = ”ex1.fas”, sep = ”\n”)
#
#   # Reading the alignment directly from file:
#   FindHaplo(inputFile=”ex1.fas”)
#
#   # Reading the alignment from an object:
# library(ape)
# example1<-read.dna(file=”ex1.fas”,format=”fasta”)
# FindHaplo(align=example1)
#
#  #generating a new alignment file with identical sequences wrongly aligned:
# cat(”>Pop1_seq1”,
# ”TTATTCTA--------TAGC”,
# ”>Pop1_seq2”,
# ”TTAT----TCTA----TAGC”,
# ”>Pop1_seq3”,
# ”TAAT----TCTA------AC”,
#      file = ”ex2.2.fas”, sep = ”\n”)
#
# # Note that seq1 and seq2 are equal, but the alignment is different.
# # However, this function identifies seq1 and seq2 as identical:
# FindHaplo(inputFile=”ex2.2.fas”)
#
```

genbank.sp.names   *Species names from genbank accessions*

Description

Given a set of sequences in fasta format, this function extract the species names of every accession.

Usage

```
genbank.sp.names(sequences)
```

Arguments

`sequences` a DNAbin object containing all the information of genbank accesions as sequence names.

Value

a DNAbin object with genus and species as sequence names.

Author(s)

A.J. Muñoz-Pajares

`get.majority.taxo` *Get majority taxonomy for a sequence*

Description

Given filtered taxonomy of multiple BLAST hits, provides the consensus taxonomy for every query sequence

Usage

```
get.majority.taxo(filtered.taxo,verbose=TRUE)
```

Arguments

`filtered.taxo` a data.frame containing (at least) the output of a BLAST analysis and the filtered taxonomy for every subject sequence. This format is produced by `filter.whole.taxo`

`verbose` a logical, if TRUE details on the calculation are shown.

Details

The expected input data.frame must contain information about filtered taxonomy for every subject sequence and a unique code for every query sequence.

Nonmeaningful names (including "unidentified", "sp", "low_pident") are coerced to "Uninformative".

The output dataframe includes a column with a summary of the alternative classifications found for taxonomy at every level. For example: "10|10|10|10|10|10|9+1" means that all 10 matches have identical taxonomy up to genus, but two species has been identified, being the taxonomy of 9 of the subject sequences identical and different from the remaining subject.

Value

a data.frame with the followin columns: "qseqid", the unique identifier of every original query sequence; seven columns containing the filtered taxonomy ("kingdom.final", "phylum.final", "class.final", "order.final", "family.final", "genus.final", and "species.final"); "values": The frequency of the different taxonomical names provides for every level, separated by "|" (see details).

Author(s)

A. J. Muñoz-Pajares

See Also

filter.whole.taxo,assign.whole.taxo

Examples

```
# data(ex_BLAST)
# TAXO <- assign.whole.taxo(ex_BLAST)
# FILT_TAXO <- filter.whole.taxo(TAXO)
# MAJ_TAXO <- get.majority.taxo(TAXO)
```

GetHaplo *Get sequences of unique haplotypes*

Description

This function returns the subset of unique sequences composing a given alignment.

Usage

```
GetHaplo(inputFile = NA, align = NA, saveFile = TRUE,
outname = "Haplotypes.txt", format = "fasta",
seqsNames = NA, silent = FALSE)
```

Arguments

inputFile	the name of a sequence alignment file in fasta format to be analysed. Alternatively you can provide the name of a "DNAbin" class alignment stored in memory using the "align" option.
align	the name of the "DNAbin" alignment to be analysed. See "?read.dna" in the ape package for details about reading alignments. Alternatively you can provide the name of the file containing the alignment in fasta format using the "inputFile" option.
saveFile	a logical; if TRUE (default), function output is saved as a text file
outname	if "saveFile" is set to TRUE (default), contains the name of the output file ("Haplotypes.txt" by default).

format	format of the DNA sequences to be saved: "interleaved", "sequential", or "fasta" (default). See "write.dna" in ape package for details.
seqsNames	names for each DNA sequence saved: Three choices are possible: if n unique sequences are found, "Inf.Hap" assigns names from H1 to Hn (according to input order). The second option is to define a vector containing n names. By default, input sequence names are used.
silent	a logical; if TRUE (default), it prints the number of unique sequences found and the name of the output file.

Details

The algorithm identifies identical sequences even if they are wrongly aligned (see example).

Value

A file containing unique sequences from the input file.

Author(s)

A. J. Muñoz-Pajares

See Also

FindHaplo

Examples

```
#  #generating an alignment file:
# cat(">Population1_sequence1",
# "TTATAAAATCTA----TAGC",
# ">Population1_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population1_sequence3",
# "TTATAAAAATTA----TAGC",
# ">Population1_sequence4",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence1",
# "TTAT----TCGAGGGGTAGC",
# ">Population2_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence3",
# "TTATAAAA--------TAGC",
# ">Population2_sequence4",
# "TTAT----TCGAGGGGTAGC",
# ">Population3_sequence1",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence2",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence3",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence4",
# "TTAT----TCGA----TAGC",
#      file = "ex2.fas", sep = "\n")
#
# # Getting unique haplotypes reading the alignment from a file and setting
# #haplotype names:
#  GetHaplo(inputFile="ex2.fas",outname="ex2_unique.fas",seqsNames=
# c("HaploK001","HaploK002","HaploS001","HaploR001","HaploR002","HaploR003"))
# # Reading the alignment from an object and using original sequence names:
#     library(ape)
#     example2 <- read.dna("ex2.fas", format = "fasta")
#  GetHaplo(align=example2,outname="Haplotypes_DefaultNames.txt")
# # Reading the alignment from an object and using new haplotype names:
#  GetHaplo(align=example2,outname="Haplotypes_sequentialNames.txt",
# seqsNames="Inf.Hap")
#
#
#  #generating a new alignment file with identical sequences wrongly alined:
# cat(">Pop1_seq1",
# "TTATTCTA--------TAGC",
# ">Pop1_seq2",
# "TTAT----TCTA----TAGC",
# ">Pop1_seq3",
# "TAAT----TCTA------AC",
#      file = "ex2.2.fas", sep = "\n")
#
# # Note that seq1 and seq2 are equal, but the alignment is different.
# # However, this function identifies seq1 and seq2 as identical:
#  a<-GetHaplo(inputFile="ex2.2.fas",saveFile=FALSE)
#
```

HapPerPop *Returns the number of haplotypes per population.*

Description

Given a two column matrix, this function returns the number of haplotypes per population (weighted matrix) and the presence/absence of haplotypes per population (interaction matrix). The input matrix must contain one row per individual. The first column must contain the population name, while the second must contain the name of the haplotypes. This input matrix can be obtained using the "FindHaplo" function.

Usage

```
HapPerPop(inputFile = NA, sep = " ", input = NA, header = FALSE,
NameIniPopulations = NA, NameEndPopulations = NA, saveFile = TRUE,
Wname = NA, Iname = NA)
```

Arguments

`inputFile` the name of the file containing the two columns input matrix. Alternatively you can provide the name of a matrix stored in memory using the "input" option.

`sep` the character separating columns in the input matrix (space, by default).

`input` the two columns input matrix stored in memory. Alternatively you can provide the name of the file containing the input matrix using the "inputFile" option.

`header` a logical value indicating whether the input matrix contains the names of the variables as its first line. (Default=FALSE).

`NameIniPopulations` Position within the input matrix rownames of the initial character referring population name. This option is useful if names contained in the first column includes more information than the population name (e.g., marker name, individual details...).

`NameEndPopulations` Position within the input matrix rownames of the last character referring population name. This option is useful if names contained in the first column includes more information than the population name (e.g., marker name, individual details...).

`saveFile` a logical; if TRUE (default), the two ouput matrices computed are saved as two different text files.

`Wname` the name given to the output weighted matrix file.

`Iname` the name given to the output interaction matrix file

Details

If both NameIniPopulations and NameEndPopulations are not defined, the names contained in the input matrix first column are used as population identifiers.

Value

A list containing two matrices:

`Weighted` The first matrix (named weighted matrix) contains the abundance of each haplotype per population, represented by the number of haplotypes (columns) found per population (rows).

`Interaction` The second matrix (named interaction matrix) contains information about the presence or absence of each haplotype (columns) per population (rows) represented by 1 or 0, respectively.

Author(s)

A. J. Muñoz-Pajares

See Also

`FindHaplo`

Examples

```
## Not run:
# cat("Sequence.Name Haplotype.Name",
# "Population1 H1",
# "Population1 H2",
# "Population1 H3",
# "Population1 H2",
# "Population2 H4",
# "Population2 H5",
# "Population2 H6",
# "Population2 H4",
# "Population3 H7",
# "Population3 H7",
# "Population3 H7",
# "Population3 H7",
#      file = "3_FindHaplo_Example2_modified.txt", sep = "\n")
#
# # Reading the alignment directly from file:
#  HapPerPop(inputFile="3_FindHaplo_Example2_modified.txt",header=TRUE,
# saveFile=FALSE)
#
# cat("Sequence.Name Haplotype.Name",
# "Population1id1 H1",
# "Population1id2 H2",
# "Population1id3 H3",
# "Population1id4 H2",
# "Population2id1 H4",
# "Population2id2 H5",
# "Population2id3 H6",
# "Population2id4 H4",
# "Population3id1 H7",
# "Population3id2 H7",
# "Population3id3 H7",
# "Population3id4 H7",
#      file = "4_FindHaplo_Example2_modified.txt", sep = "\n")
#
# # Reading the alignment directly from file. First column includes population
# # and individual names. Consequently, 12 populations are considered:
#  HapPerPop(inputFile="4_FindHaplo_Example2_modified.txt",header=TRUE,
# saveFile=FALSE)
#
# # Population names within the input matrix first column goes from
# # character 1 to 11. Now 3 populations are considered:
#  HapPerPop(inputFile="4_FindHaplo_Example2_modified.txt",header=TRUE,
# saveFile=FALSE,NameIniPopulations=1, NameEndPopulations=11)
#
# # If population names are set from character 1 to 3, all samples would
# # be treated as a single population
#  HapPerPop(inputFile="4_FindHaplo_Example2_modified.txt",header=TRUE,
# saveFile=FALSE,NameIniPopulations=1, NameEndPopulations=3)
#
# # Reading the alignment directly from file, displaying only the
# # weighted matrix:
#  HapPerPop(inputFile="4_FindHaplo_Example2_modified.txt",header=TRUE,
# saveFile=FALSE,NameIniPopulations=1, NameEndPopulations=11)[[1]]
#
# # Reading the alignment from an object and saving the two computed
# # distance matrices:
#  FH<-read.table("3_FindHaplo_Example2_modified.txt",header=TRUE)
#  HapPerPop(input=FH,header=TRUE,saveFile=FALSE)
#
## End(Not run)
```

`inter.intra.plot` *Histogram of the intra- and interspecific distances*

Description

Plot histogram for inter and intra-specific distances together

Usage

```
inter.intra.plot(dismat=NA, xlim=NULL,ylim=NULL,
intra.col="gray",intra.density=0,intra.n=30,plot="N",
inter.col="black",inter.density=0,inter.n=30,legend=TRUE,
main="",xlab="Genetic distances",ylab=NULL)
```

Arguments

`dismat` a symmetric matrix containing the pairwise genetic distances between individual sequences.

`xlim` a vector containing the minimum and maximum value in the x-axis

`ylim` a vector containing the minimum and maximum value in the y-axis

`intra.col` the colour for the intraspecific distance distribution

`intra.density` a numeric, the density of shading lines for the intraspecific distance distribution

`intra.n` a numeric, the number of categories to represent in the intraspecific distance distribution

`plot` a string, "freq" to represent frequency values in the y-axis and "N" for number of occurrences

`inter.col` the colour for the interspecific distance distribution

`inter.density` a numeric, the density of shading lines for the interspecific distance distribution

`inter.n` a numeric, the number of categories to represent in the interspecific distance distribution

`legend` a logic, "TRUE" to show plot legend

`main` a string containing the title of the plot

`xlab` a string with the label of the x-axis

`ylab` a string with the label of the x-axis

Value

A list with two elements:

`Intraspecific` a vector containing all the intraspecific distances.

`Interspecific` a vector containing all the interspecific distances.

Author(s)

A.J. Muñoz-Pajares

Examples

```
# # Generating a distance matrix:
#
# my.mat<-matrix(nrow=100,ncol=100,
# dimnames=list(paste(”sp”,rep(1:2,50),
# sep=””),paste(”sp”,rep(1:2,50),sep=””)))
# L<-my.mat[seq(1,nrow(my.mat),2),seq(1,ncol(my.mat),2)]
# my.mat[seq(1,nrow(my.mat),2),seq(1,ncol(my.mat),2)]<-rnorm(0.15,n=L,sd=0.01)
# my.mat[seq(2,nrow(my.mat),2),seq(2,ncol(my.mat),2)]<-rnorm(0.15,n=L,sd=0.01)
# my.mat[seq(1,nrow(my.mat),2),seq(2,ncol(my.mat),2)]<-rnorm(0.3,n=L,sd=0.04)
# my.mat[seq(2,nrow(my.mat),2),seq(1,ncol(my.mat),2)]<-rnorm(0.3,n=L,sd=0.04)
# #Converting to symmetric
# my.mat<-as.matrix(as.dist(my.mat))
# inter.intra.plot(dismat=my.mat)
# inter.intra.plot(dismat=my.mat,intra.n=10)
# inter.intra.plot(dismat=my.mat,plot=”Freq”,intra.n=10)
```

MCIC *Modified Complex Indel Coding as distance matrix*

Description

This function computes an indel distance matrix following the rationale of the Modified Complex Indel Coding (Muller, 2006) to estimate transition matrices.

Usage

```
MCIC(inputFile = NA, align = NA, saveFile = TRUE, outname =
paste(inputFile, ”IndelDistanceMatrixMullerMod.txt”), silent = FALSE)
```

Arguments

`inputFile` the name of the fasta file to be analysed. Alternatively you can provide the name of a "DNAbin" class alignment stored in memory using the "align" option.

`align` the name of the alignment to be analysed. See "read.dna" in ape package for details about reading alignments. Alternatively you can provide the name of the file containing the alignment in fasta format using the "inputFile" option.

saveFile	a logical; if TRUE (default), function output is saved as a text file.
outname	if "saveFile" is set to TRUE (default), contains the name of the output file.
silent	a logical; if FALSE (default), it prints the number of unique sequences found and the name of the output file.

Details

It is recommended to estimate this distance matrix using only the unique sequences in the alignment. Repeated sequences increase computation time but do not provide additional information (because they produce duplicated rows and columns in the final distance matrix).

Value

A matrix containing the genetic distances estimated as indels pairwise differences.

Author(s)

A. J. Muñoz-Pajares

References

Muller K. (2006). Incorporating information from length-mutational events into phylogenetic analysis. Molecular Phylogenetics and Evolution, 38, 667-676.

Examples

```
# # This will generate an example file in your working directory:
# cat(">Population1_sequence1",
# "A-AGGGTC-CT---G",
# ">Population1_sequence2",
# "TAA---TCGCT---G",
# ">Population1_sequence3",
# "TAAGGGTCGCT---G",
# ">Population1_sequence4",
# "TAA---TCGCT---G",
# ">Population2_sequence1",
# "TTACGGTCG---TTG",
# ">Population2_sequence2",
# "TAA---TCG---TTG",
# ">Population2_sequence3",
# "TAA---TCGCTATTG",
# ">Population2_sequence4",
# "TTACGGTCG---TTG",
# ">Population3_sequence1",
# "TTA---TCG---TAG",
# ">Population3_sequence2",
# "TTA---TCG---TAG",
# ">Population3_sequence3",
# "TTA---TCG---TAG",
# ">Population3_sequence4",
# "TTA---TCG---TAG",
#       file = "ex3.fas", sep = "\n")
#
# # Reading the alignment directly from file and saving no output file:
# MCIC (input="ex3.fas", saveFile = FALSE)
#
# # Analysing the same dataset, but using only unique sequences:
# uni<-GetHaplo(inputFile="ex3.fas",saveFile=FALSE)
# MCIC (align=uni, saveFile = FALSE)
#
```

mergeNodes *Merges nodes showing distance values equal to zero*

Description

This function returns a new distance matrix merging rows (and columns) showing distance values equal to zero. It also deals with missing data.

Usage

```
mergeNodes(dis, na.rm.row.col = FALSE, save.distance = FALSE,
save.distance.name = "Merged_Distance.txt")
```

Arguments

dis	the input distance matrix
na.rm.row.col	a logical; if TRUE, missing values are removed before the computation proceeds.
save.distance	a logical; if TRUE, the new distance matrix will be saved in a file.
save.distance.name	a string; if save.distance is set to TRUE, it defines the name of the file to be saved.

Details

In some circumstances you may get distance matrices showing off-diagonal zeros. In such cases you may consider that the existence of these off-diagonal zeros suggests that some of the groups you defined (e.g., populations) are not genetically different. Thus, you must re-define groups to get a matrix composed only by different groups using the 'mergeNodes' function and estimate a percolation network using the 'perc.thr' function. On the other hand, you may consider that, despite the off- diagonal zeros, the groups you defined are actually different. In that case you may not be able to estimate a percolation threshold, but you can represent the original distance matrix using the 'NINA.thr' or the 'zero.thr' functions.

'mergeNodes' select all rows (and columns) showing a distance equal to zero and generates a new row (and column). The distance between the new merged and the remaining rows (or columns) in the matrix is estimated as the arithmetic mean of the selected elements. The biological interpretation of the new matrix could be hard if the original matrix shows a large number of off-diagonal zeros.

’perc.thr’ estimates a threshold to represent a distance matrix as a network. To estimate this threshold, the algorithm represents as a link all distances lower than a range of thresholds (by default, select 101 values from 0 to 1), defined as the percentage of the maximum distance in the input matrix. For each threshold a network is built and the number of clusters (that is, the number of isolated groups of nodes) in the network is also estimated. Finally, the algorithm selects the lower threshold connecting a higher number of nodes. Note that the resulting network may show isolated nodes if it is necessary to represent a large number of links to connect a low number of nodes.

’NINA.thr’ is identical to ’perc.thr’, but, in the last step, the algorithm selects the lower threshold connecting all nodes in a single cluster. The information provided by this function may be limited if the original distance matrix shows high variation.

’zero.thr’ represents as a link only distances equal to zero. The information provided by this function may be limited if the original matrix shows few off-diagonal zeros.

Value

a distance matrix with merged rows and columns

Author(s)

A. J. Muñoz-Pajares

See Also

NINA.thr, zero.thr, perc.thr

Examples

```
#EXAMPLE 1: FEW OFF-DIAGONAL ZEROS

#Generating a distance matrix:
Dis1<-matrix(c(
0.00,0.77,0.28,0.94,0.17,0.14,0.08,0.49,0.64,0.01,
0.77,0.00,0.12,0.78,0.97,0.02,0.58,0.09,0.36,0.33,
0.28,0.12,0.00,0.70,0.73,0.06,0.50,0.79,0.80,0.94,
0.94,0.78,0.70,0.00,0.00,0.78,0.04,0.42,0.25,0.85,
0.17,0.97,0.73,0.00,0.00,0.30,0.55,0.12,0.68,0.99,
0.14,0.02,0.06,0.78,0.30,0.00,0.71,1.00,0.64,0.88,
0.08,0.58,0.50,0.04,0.55,0.71,0.00,0.35,0.84,0.76,
0.49,0.09,0.79,0.42,0.12,1.00,0.35,0.00,0.56,0.81,
0.64,0.36,0.80,0.25,0.68,0.64,0.84,0.56,0.00,0.62,
0.01,0.33,0.94,0.85,0.99,0.88,0.76,0.81,0.62,0.00),ncol=10)
colnames(Dis1)<-c(paste("Pop",c(1:10),sep=""))
row.names(Dis1)<-colnames(Dis1)

# No percolation threshold can be found.
#perc.thr(Dis1)

# #Check Dis1 and merge populations showing distances equal to zero:
# Dis1
# Dis1_Merged<-mergeNodes(dis=Dis1)
# #Check the merged matrix. A new "population" has been
# #defined merging populations 4 and 5.
# #Distances between the merged and the remaining populations are estimated as the arithmetic mean.
# Dis1_Merged
# # It is now possible to estimate a percolation threshold
# perc.thr(dis=Dis1_Merged,ptPDF=FALSE, estimPDF=FALSE, estimOutfile=FALSE)

# EXAMPLE 2: TOO MANY OFF-DIAGONAL ZEROS
# #Generating a distance matrix:
# Dis2<-matrix(c(
# 0.00,0.77,0.28,0.00,0.17,0.14,0.00,0.49,0.64,0.01,
# 0.77,0.00,0.12,0.00,0.97,0.02,0.00,0.09,0.36,0.33,
# 0.28,0.12,0.00,0.70,0.73,0.06,0.50,0.79,0.00,0.94,
# 0.00,0.00,0.70,0.00,0.00,0.78,0.04,0.00,0.00,0.00,
# 0.17,0.97,0.73,0.00,0.00,0.30,0.55,0.12,0.00,0.00,
# 0.14,0.02,0.06,0.78,0.30,0.00,0.71,1.00,0.64,0.00,
# 0.00,0.00,0.50,0.04,0.55,0.71,0.00,0.35,0.84,0.00,
# 0.49,0.09,0.79,0.00,0.12,1.00,0.35,0.00,0.56,0.81,
# 0.64,0.36,0.00,0.00,0.00,0.64,0.84,0.56,0.00,0.62,
# 0.01,0.33,0.94,0.00,0.00,0.00,0.00,0.81,0.62,0.00),ncol=10)
# colnames(Dis2)<-c(paste(”Pop”,c(1:10),sep=””))
# row.names(Dis2)<-colnames(Dis2)
#
# # No percolation threshold can be found
# #perc.thr(Dis2)
#
# #Check Dis2 and merge populations showing distances equal to zero:
# Dis2
# Dis2_Merged<-mergeNodes(dis=Dis2)
#
# #Check the merged matrix. Many new ”populations” have been defined
# #and both the new matrix and the resulting network are difficult
# #to interpret:
# Dis2_Merged
# perc.thr(dis=Dis2_Merged,ptPDF=FALSE, estimPDF=FALSE, estimOutfile=FALSE)
#
# #Instead of percolation network, representing zeros
# #as the lowest values may be informative:
# zero.thr(dis=Dis2,ptPDF=FALSE)
#
# # Adjusting sizes and showing modules:
# zero.thr(dis=Dis2,ptPDF=FALSE,cex.label=0.8,cex.vertex=1.2,modules=TRUE)
#
# #In the previous example, the 'zero.thr' method is unuseful:
# zero.thr(dis=Dis1,ptPDF=FALSE)
#
# #In both cases, the 'No Isolation Nodes Allowed' method yields an informative matrix:
# NINA.thr(dis=Dis1,modules=TRUE)
# NINA.thr(dis=Dis2,modules=TRUE)
```

mutation.network *Haplotype network depiction including mutations*

Description

This function represents an alignment as a network and displays mutations (substitutions and indels) as marks in edges.

Usage

```
mutation.network(align = NA, indel.method = "MCIC",
substitution.model = "raw", pairwise.deletion = TRUE,
network.method = "percolation", range = seq(0, 1, 0.01),
merge.dist.zero=TRUE, addExtremes = FALSE, alpha = "info",
combination.method = "Corrected", na.rm.row.col = FALSE,
modules = FALSE, moduleCol = NA,
modFileName = "Modules_summary.txt", save.distance = FALSE,
save.distance.name = "DistanceMatrix_threshold.txt",
silent = FALSE, bgcol = "white", label.col = "black",
label = NA, label.sub.str = NA, colInd = "red",
colSust = "black", lwd.mut = 1, lwd.edge = 1.5,
cex.mut = 1, cex.label = 1, cex.vertex = 1, main = "",
InScale = 1, SuScale = 1, legend = NA, legend.bty = "o",
legend.pos="bottomright", large.range = FALSE, pies = FALSE,
NameIniPopulations = NA, NameEndPopulations = NA,
NameIniHaplotypes = NA, NameEndHaplotypes = NA,
HaplosNames = NA,verbose = TRUE)
```

Arguments

`align` a ’DNAbin’ object; the alignment to be analysed. See "read.dna" in the ape package for details about reading alignments.

`indel.method` a sting; the method to define indel events in your alignments. The available methods are:

-"MCIC": (Default) Estimates indel events following the rationale of the Modified Complex Indel Coding (Muller, 2006).

-"SIC": Estimates indel events following the rationale of Simmons and Ochoterrena (2000).

-"FIFTH": Estimates indel events following the rationale of the fifth state: each gap within the alignment is treated as an independent mutation event.

-"BARRIEL": Estimates indel events following the rationale of Barriel (1994): singleton gaps are not taken into account.

`substitution.model`

a string; the substitution evolutionary model to estimate the distance matrix. By default is set to "raw" and estimates the pairwise proportion of variant sites. See the evolutionary models available using ?dist.dna from the ape package.

`pairwise.deletion`

a logical; if TRUE (default) substitutions found in regions being a gap in other sequences will account for the distance matrix. If FALSE, sites being a gap in at least one sequence will be removed before distance estimation.

`network.method` a string; the method to build the network. The available methods are:

-"percolation": computes a network using the percolation network method following Rozenfeld et al. (2008). See ?perc.thr for details

-"NINA": computes a network using the No Isolation Nodes Allowed method. See ?NINA.thr for details.

-"zero": computes a network connecting all nodes showing distances equal to zero. See ?zero.thr for details.

`range` a numeric vector between 0 and 1, is the range of thresholds (referred to the maximum distance in the input matrix) to be screened (by default, 101 values from 0 to 1). This option is used for "percolation" and "NINA" network methods and ignored for "zero" method.

`merge.dist.zero`

a logical; if TRUE, nodes showing a distance equal to zero are merged using the mergeNodes() function

`addExtremes` a logical; if TRUE, additional nucleotide sites are included in both extremes of the alignment. This will allow estimating distances for alignments showing gaps in terminal positions. This option is used for "SIC", "FIFTH" and "BARRIEL" indel methods and ignored for "MCIC" method.

`alpha` a numeric between 0 and 1, is the weight given to the indel genetic distance matrix for the combination. By definition, the weight of the substitution genetic matrix is the complementary value (i.e., 1-alpha). The value "info" (default) will use the proportion of informative substitutions per informative indel event as weight. It is also possible to define multiple weights to estimate different combinations.

`combination.method`

a string defining whether each distance matrix must be divided by its maximum value before the combination ("Corrected") or not ("Uncorrected"). Consequently, if the "Corrected" method is chosen, both matrices will range between 0 and 1 before being combined.

`na.rm.row.col` a logical; if TRUE, distance matrix missing values are removed.

`modules` a logical, If TRUE, nodes belonging to different modules are represented as different colours (defined by 'moduleCol'). Modules (defined as subsets of nodes that conform densely connected subgraphs) are estimated by means of random walks (see 'igraph' package for details).

`moduleCol` (if modules=TRUE) a vector of strings defining the colour of nodes belonging to different modules in the network. If 'NA' (or there are less colours than haplotypes), colours are automatically selected

`modFileName` (if modules=TRUE) a string, the name of the file to be generated containing a summary of module results (sequence name, module, and colour in network)

`save.distance` a logical; if FALSE (default), the distance matrix used for computation is saved in a file

`save.distance.name`

if save.distace=TRUE, a string defining the file name

`silent` a logical; if FALSE (default), displays a list containing the number of indels and substitutions represented in the network.

`bgcol` a vector of strings; the colour of the background for each node in the network. Can be equal for all nodes (if only one colour is defined), customized (if several colours are defined), or can represent different modules (see "modules" option). If set to ’NA’ (default) or if less colours than haplotypes are defined, colours are automatically selected.

`label.col` a vector of strings; the colour of labels for each node in the network. Can be equal for all nodes (if only one colour is defined) or customized (if several colours are defined).

`label` a vector of strings; labels for each node. By default are the sequence names. (See ’label.sub.str’ to automatically reduce name lengths)

`label.sub.str` a vector of two numerics; if node labels are a substring of sequence names, these two numbers represent the initial and final character of the string to be represented. See Example for details.

`lwd.edge` a numeric; the width of the edge linking nodes (1.5 by default).

`colInd` a strings; the colour used to represent indels (red by default).

`colSust` a strings; the colour used to represent substitutions (black by default).

`lwd.mut` a numeric; the width of the line (perpendicular to the edge line) representing mutations (1 by default).

`cex.mut` a numeric; the length of the line (perpendicular to the edge line) representing mutations (1 by default).

`cex.vertex` a numeric; the size of the nodes.

`cex.label` a numeric; the size of the node labels.

`main` if set to "summary" the main options selected for representing the network are displayed in title. The default value ("") shows no title for the network.

`InScale` a numeric; the number of indels each mark represents. By default is set to 1 (that is, 1 mark= 1 indel). If set to 10, then 1 mark=10 indels. In that case, if there are 25 indels in a given edge it is represented by three marks (being one of them half width than the other two).

`SuScale` a numeric; the number of substitutions each mark represents. By default is set to 1 (that is, 1 mark= 1 substitution). If set to 10, then 1 mark=10 substitutions In that case, if there are 25 substitutions in a given edge it is represented by three marks (being one of them half width than the other two).

`legend` a logic; if TRUE, plots a legend representing the scale of marks (that is, the number of mutations represented by a mark).

`legend.bty` a letter; the type of box to be drawn around the legend. The allowed values are ’o’ (default, producing a four-sides box) and ’n’ (producing no box).

`legend.pos` a string, defines legend position ("bottomright" by default). Other possible values are: ‘"bottomright"’, ‘"bottom"’, ‘"bottomleft"’, ‘"left"’, ‘"topleft"’, ‘"top"’, ‘"topright"’, ‘"right"’ and ‘"center"’.

`large.range` a logic, TRUE for representing node size according to three categories: haplotypes appearing less than 10 times, between 10 and 100 times and more than 100 times

`pies` a logic, TRUE for representing nodes as pies and FALSE for representing nodes as points

`NameIniPopulations`
a numeric; Position of the initial character of population names within sequence names. If not provided, it is set to 1. It is used only if NameEndPopulations is also defined.

`NameEndPopulations`
a numeric; Position of the last character of population names within sequence names. If not provided, it is set to the first "_" character in the sequences name. It is used only if NameIniPopulations is also defined.

`NameIniHaplotypes`
a numeric; Position of the initial character of haplotype names within sequence names. If not provided, haplotype names are given and the value is set accordingly. It is used only if NameEndHaplotypes is also defined.

`NameEndHaplotypes`
a numeric; Position of the last character of haplotype names within sequence names. If not provided, haplotype names are given and the value is set accordingly. It is used only if NameIniHaplotypes is also defined.

`HaplosNames` a sting; the name of the haplotypes (if different from default: H1...Hn)

`verbose` a logical, if TRUE details on the calculation are shown.

Details

Despite the large list of options, the only mandatory option for this function is the alignment to be represented. The remaining options can be classified into four groups:

1- options defining the computation of both indel and substitution distances (indel.method, substitution.model, pairwise.deletion).

2- options defining the combination of these two distance matrices (alpha, combination.method, na.rm.row.col, addExtremes, save.distance, save.distance.name).

3- options defining the computation of the network (network.method, range).

4- options customizing the resulting network (modules, moduleCol, modFileName, bgcol, label.col, label, label.sub.str, colInd, colSust, lwd.mut, lwd.edge, cex.mut, cex.label, cex.vertex, main).

Although the 'indel.method' option affect both the distance estimation and the number of mutations represented in the network, the 'substitution.model' and 'pairwise.deletion' options only affect the distance matrix computation. The number of substitutuions represented in the network are always estimated using the "N" model and the pairwise deletion of indels.

Author(s)

A. J. Muñoz-Pajares

References

Barriel, V., 1994. Molecular phylogenies and how to code insertion/ deletion events. Life Sci. 317, 693-701, cited and described by Simmons, M.P., Müller, K. & Norton, A.P. (2007) The relative performance of indel-coding methods in simulations. Molecular Phylogenetics and Evolution, 44, 724–740.

Muller K. (2006). Incorporating information from length-mutational events into phylogenetic analysis. Molecular Phylogenetics and Evolution, 38, 667-676.

Paradis, E., Claude, J. & Strimmer, K. (2004). APE: analyses of phylogenetics and evolution in R language. Bioinformatics, 20, 289-290.

Rozenfeld AF, Arnaud-Haond S, Hernandez-Garcia E, Eguiluz VM, Serrao EA, Duarte CM. (2008). Network analysis identifies weak and strong links in a metapopulation system. Proceedings of the National Academy of Sciences,105, 18824-18829.

Simmons, M.P. & Ochoterena, H. (2000). Gaps as Characters in Sequence-Based Phylogenetic Analyses. Systematic Biology, 49, 369-381.

See Also

plot.network, double.plot

Examples

```
# cat(">Population1_sequence1",
# "TTATAAAATCTA----TAGC",
# ">Population1_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population1_sequence3",
# "TTATAAAAATTA----TAGC",
# ">Population1_sequence4",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence1",
# "TTAT----TCGAGGGGTAGC",
# ">Population2_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence3",
# "TTATAAAA--------TAGC",
# ">Population2_sequence4",
# "TTAT----TCGAGGGGTAGC",
# ">Population3_sequence1",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence2",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence3",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence4",
# "TTAT----TCGA----TAGC",
#      file = "ex2.fas", sep = "\n")
#
# library(ape)
#
# #Network with default options
# mutation.network (align=read.dna(file="ex2.fas",format="fasta"))
#
# #Using more options:
# mutation.network (align=read.dna(file="ex2.fas",format="fasta"),modules=TRUE)
#
# #A more complex alignment
# data(ex_alignment1) # this will read a fasta file with the name 'alignExample'
# mutation.network (align=alignExample,modules=TRUE,
# InScale=2, SuScale=2,legend=TRUE,lwd.mut=1.8)
#
```

mutationSummary *Summary of observed mutations*

Description

This function computes the number of substitutions and indels observed in a given alignment.

Usage

```
mutationSummary(align, addExtremes = FALSE, output = "brief")
```

Arguments

`align` the name of the "DNAbin" alignment to be analysed. See "?read.dna" in the ape package for details about reading alignments.

`addExtremes` a logical; if TRUE, additional nucleotide sites are included in both extremes of the alignment. This will allow estimating distances for alignments showing gaps in terminal positions.

`output` a string; defines the kind of output. Two values are accepted: - "brief" (default) produces an output showing the number of mutations (sites and events). - "detailed" produces an output showing the number of mutations (sites and events), the position of each mutation, and the state of these sites per sequence (A, T, C, G or - for substitutions and 1 or 0 for indels).

Value

A list containing:

`Sites` A matrix containing: the number of sites per sequence (Length); the number of constant and variable sites; the number of singletons and informative sites.

`Events` A matrix containing: the number of substitution (singletons, informative, and total) and indel (singletons, informative, and total) events

`Constants.Alignment` A matrix showing constant sites in the alignment (Only shown if output=="detailed").

`Variables.Alignment` A matrix showing variable sites in the alignment (Only shown if output=="detailed").

`Singletons.Alignment` A matrix showing singleton sites in the alignment (Only shown if output=="detailed").

`Inforatives.Alignment` A matrix showing informative sites in the alignment (Only shown if output=="detailed").

`Substitutions` A matrix showing substitution sites in the alignment (Only shown if output=="detailed").

`Subst.Single` A matrix showing singleton substitution sites in the alignment (Only shown if output=="detailed").

`Subst.Info` A matrix showing informative substitution sites in the alignment (Only shown if output=="detailed").

`Gaps` A matrix showing gap sites in the alignment (Only shown if output=="detailed").

`Gaps.Single` A matrix showing singleton gap sites in the alignment (Only shown if output=="detailed").

`Gaps.Info` A matrix showing informative gap sites in the alignment (Only shown if output=="detailed").

Author(s)

A. J. Muñoz-Pajares

Examples

```
# cat(">Population1_sequence1",
# "A-AGGGTC-CT---G",
# ">Population1_sequence2",
# "TAA---TCGCT---G",
# ">Population1_sequence3",
# "TAAGGGTCGCT---G",
# ">Population1_sequence4",
# "TAA---TCGCT---G",
# ">Population2_sequence1",
# "TTACGGTCG---TTG",
# ">Population2_sequence2",
# "TAA---TCG---TTG",
# ">Population2_sequence3",
# "TAA---TCGCTATTG",
# ">Population2_sequence4",
# "TTACGGTCG---TTG",
# ">Population3_sequence1",
# "TTA---TCG---TAG",
# ">Population3_sequence2",
# "TTA---TCG---TAG",
# ">Population3_sequence3",
# "TTA---TCG---TAG",
# ">Population3_sequence4",
# "TTA---TCG---TAG",
#      file = "ex3.fas", sep = "\n")
#
# # Reading the alignment directly from file and saving no output file:
# library(ape)
# mutationSummary (align=read.dna("ex3.fas",format="fasta"))
# mutationSummary (align=read.dna("ex3.fas",format="fasta"),output="detailed")
#
# #A more complex alignment
# data(ex_alignment1) # this will read a fasta file with the name 'alignExample'
# mutationSummary(align=alignExample,addExtremes=TRUE)
#
```

NINA.thr *No Isolated Nodes Allowed network*

Description

Given a distance matrix, this function computes a network connecting all nodes with the minimum number of links.

Usage

```
NINA.thr(dis, range = seq(0, 1, 0.01), ptPDF = TRUE,
ptPDFname = "NINA_Network.pdf", estimPDF = TRUE,
estimPDFname = "NINA.ThresholdEstimation.pdf",
estimOutfile = TRUE, cex.label = 1, cex.vertex = 1,
estimOutName = "NINA.ThresholdEstimation.txt",
appendOutfile = TRUE, plotALL = FALSE, bgcol = "white",
label.col = "black", label = colnames(dis), modules = FALSE,
moduleCol = NA, modFileName = "Modules_summary.txt", ncs = 4,
na.rm.row.col = FALSE)
```

Arguments

`dis`	the input distance matrix
`range`	a numeric vector between 0 and 1, is the range of thresholds (referred to the maximum distance in the input matrix) to be screened (by default, 101 values from 0 to 1).
`ptPDF`	a logical, must the resulting network be saved as a pdf file?
`ptPDFname`	if ptPDF=TRUE, the name of the pdf file containing the network to be saved ("NINA_Network.pdf", by default)
`estimOutfile`	a logical, must the value of <s> for each threshold (NINA threshold estimation) be saved as a text file?
`estimOutName`	if estimOutfile=TRUE (default), contains the name of the text file containing the NINA threshold estimation ("PercThr Estimation.txt" by default).
`appendOutfile`	a logical, if estimOutfile=TRUE, it defines whether results must be appended to an existing file with the same name (TRUE) or the existing file must be replaced (FALSE).
`estimPDF`	a logical, must the NINA threshold estimation plot be saved as a pdf file?
`estimPDFname`	if estimPDF=TRUE (default), defines the name of the pdf file containing the NINA threshold estimation plot (by default).
`cex.label`	a numeric; the size of the node labels.
`cex.vertex`	a numeric; the size of the nodes.
`plotALL`	a logical, must all the networks calculated during the NINA threshold estimation (defined by "range" option) be saved as different pdf files? (FALSE, by default). If TRUE, for each value defined in threshold, one pdf file is generated.

`bgcol` the colour of the background for each node in the network. Can be equal for all nodes (if only one colour is defined), customized (if several colours are defined), or can represent different modules (see "modules" option).

`label.col` vector of strings defining the colour of labels for each node in the network. Can be equal for all nodes (if only one colour is defined) or customized (if several colours are defined).

`label` vector of strings, labels for each node. By default are the column names of the distance matrix (dis). (See substr function in base package to automatically set a string subset from column names).

`modules` a logical, If TRUE, nodes belonging to different modules are represented as different colours (defined by 'moduleCol'). Modules (defined as subsets of nodes that conform densely connected subgraphs) are estimated by means of random walks (see 'igraph' package for details).

`moduleCol` (if modules=TRUE) a vector of strings defining the colour of nodes belonging to different modules in the network. If 'NA' (or there are less colours than haplotypes), colours are automatically selected

`modFileName` (if modules=TRUE) the name of a generated file containing a summary of module results

`ncs` a numeric; number of decimal places to display threshold in plot title.

`na.rm.row.col` a logical; if TRUE, missing values are removed before the computation proceeds.

Details

In some circumstances you may get distance matrices showing off-diagonal zeros. In such cases you may consider that the existence of these off-diagonal zeros suggests that some of the groups you defined (e.g., populations) are not genetically different. Thus, you must re-define groups to get a matrix composed only by different groups using the 'mergeNodes' function and estimate a percolation network using the 'perc.thr' function. On the other hand, you may consider that, despite the off- diagonal zeros, the groups you defined are actually different. In that case you may not be able to estimate a percolation threshold, but you can represent the original distance matrix using the 'NINA.thr' or the 'zero.thr' functions.

'mergeNodes' select all rows (and columns) showing a distance equal to zero and generates a new row (and column). The distance between the new merged and the remaining rows (or columns) in the matrix is estimated as the arithmetic mean of the selected elements. The biological interpretation of the new matrix could be hard if the original matrix shows a large number of off-diagonal zeros.

'perc.thr' estimates a threshold to represent a distance matrix as a network. To estimate this threshold, the algorithm represents as a link all distances lower than a range of thresholds (by default, select 101 values from 0 to 1), defined as the percentage of the maximum distance in the input matrix. For each threshold a network is built and the number of clusters (that is, the number of isolated groups of nodes) in the network is also estimated. Finally, the algorithm selects the lower threshold connecting a higher number of nodes. Note that the resulting network may show isolated nodes if it is necessary to represent a large number of links to connect a low number of nodes.

'NINA.thr' is identical to 'perc.thr', but, in the last step, the algorithm selects the lower threshold connecting all nodes in a single cluster. The information provided by this function may be limited if the original distance matrix shows high variation.

'zero.thr' represents as a link only distances equal to zero. The information provided by this function may be limited if the original matrix shows few off-diagonal zeros.

Author(s)

A. J. Muñoz-Pajares

See Also

`mergeNodes`, `zero.thr`, `perc.thr`

Examples

```
#EXAMPLE 1: FEW OFF-DIAGONAL ZEROS
#Generating a distance matrix:
Dis1<-matrix(c(
0.00,0.77,0.28,0.94,0.17,0.14,0.08,0.49,0.64,0.01,
0.77,0.00,0.12,0.78,0.97,0.02,0.58,0.09,0.36,0.33,
0.28,0.12,0.00,0.70,0.73,0.06,0.50,0.79,0.80,0.94,
0.94,0.78,0.70,0.00,0.00,0.78,0.04,0.42,0.25,0.85,
0.17,0.97,0.73,0.00,0.00,0.30,0.55,0.12,0.68,0.99,
0.14,0.02,0.06,0.78,0.30,0.00,0.71,1.00,0.64,0.88,
0.08,0.58,0.50,0.04,0.55,0.71,0.00,0.35,0.84,0.76,
0.49,0.09,0.79,0.42,0.12,1.00,0.35,0.00,0.56,0.81,
0.64,0.36,0.80,0.25,0.68,0.64,0.84,0.56,0.00,0.62,
0.01,0.33,0.94,0.85,0.99,0.88,0.76,0.81,0.62,0.00),ncol=10)
colnames(Dis1)<-c(paste("Pop",c(1:10),sep=""))
row.names(Dis1)<-colnames(Dis1)

# No percolation threshold can be found.
#perc.thr(Dis1)

#Check Dis1 and merge populations showing distances equal to zero:
# Dis1
# Dis1_Merged<-mergeNodes(dis=Dis1)
#Check the merged matrix. A new "population" has been defined merging populations 4 and 5.
#Distances between the merged and the remaining populations are estimated as the arithmetic mean.
# Dis1_Merged
# It is now possible to estimate a percolation threshold
# perc.thr(dis=Dis1_Merged,ptPDF=FALSE, estimPDF=FALSE, estimOutfile=FALSE)

# EXAMPLE 2: TOO MANY OFF-DIAGONAL ZEROS
#Generating a distance matrix:
# Dis2<-matrix(c(
# 0.00,0.77,0.28,0.00,0.17,0.14,0.00,0.49,0.64,0.01,
# 0.77,0.00,0.12,0.00,0.97,0.02,0.00,0.09,0.36,0.33,
# 0.28,0.12,0.00,0.70,0.73,0.06,0.50,0.79,0.00,0.94,
# 0.00,0.00,0.70,0.00,0.00,0.78,0.04,0.00,0.00,0.00,
# 0.17,0.97,0.73,0.00,0.00,0.30,0.55,0.12,0.00,0.00,
# 0.14,0.02,0.06,0.78,0.30,0.00,0.71,1.00,0.64,0.00,
# 0.00,0.00,0.50,0.04,0.55,0.71,0.00,0.35,0.84,0.00,
# 0.49,0.09,0.79,0.00,0.12,1.00,0.35,0.00,0.56,0.81,
# 0.64,0.36,0.00,0.00,0.00,0.64,0.84,0.56,0.00,0.62,
# 0.01,0.33,0.94,0.00,0.00,0.00,0.00,0.81,0.62,0.00),ncol=10)
# colnames(Dis2)<-c(paste("Pop",c(1:10),sep=""))
# row.names(Dis2)<-colnames(Dis2)
#
# # No percolation threshold can be found
# #perc.thr(Dis2)
#
# #Check Dis2 and merge populations showing distances equal to zero:
# Dis2
# Dis2_Merged<-mergeNodes(dis=Dis2)
#
# #Check the merged matrix. Many new "populations" have been defined
# #and both the new matrix and the resulting network
# #are difficult to interpret:
# Dis2_Merged
# perc.thr(dis=Dis2_Merged,ptPDF=FALSE, estimPDF=FALSE, estimOutfile=FALSE)
#
# #Instead of percolation network, representing zeros as the lowest values
# #may be informative:
# zero.thr(dis=Dis2,ptPDF=FALSE)
# # Adjusting sizes and showing modules:
# zero.thr(dis=Dis2,ptPDF=FALSE,cex.label=0.8,cex.vertex=1.2,modules=TRUE)
#
# #In the previous example, the 'zero.thr' method is unuseful:
# zero.thr(dis=Dis1,ptPDF=FALSE)
#
# #In both cases, the 'No Isolation Nodes Allowed' method
# #yields an informative matrix:
# NINA.thr(dis=Dis1)
# NINA.thr(dis=Dis2)
```

---

nt.gap.comb *substitution and indel distance combinations*

Description

This function obtains a lineal combination from two original matrices. The weight of each matrix in the combination must be defined. If it is a range of values, several matrices are computed.

Usage

```
nt.gap.comb(DISTnuc = NA, DISTgap = NA, alpha = seq(0, 1, 0.1),
method = "Corrected", saveFile = TRUE, align = NA, silent = FALSE)
```

Arguments

`DISTnuc` a matrix containing substitution genetic distances. See "dist.dna" in "ape" package.

`DISTgap` a matrix containing indel genetic distances.

`alpha` a numeric between 0 and 1, is the weight given to the indel genetic distance matrix in the combination. By definition, the weight of the substitution genetic matrix is the complementary value (i.e., 1-alpha). The value "info" will use the proportion of informative substitutions per informative indel event as weight. It is also possible to define multiple weights to estimate different combinations (See examples to obtain 11 corrected combined matrices using a range of alpha values).

`method` a string defining whether each distance matrix must be divided by its maximum value before the combination ("Corrected") or not ("Uncorrected"). Consequently, if the "Corrected" method is chosen, both matrices will range between 0 and 1 before to be combined.

`saveFile` a logical; if TRUE (default), each output matrix is saved in a different text file.

`align` if alpha="info" must contain the name of the alignment to be analysed. See "read.dna" in ape package for details about reading alignments.

`silent` a logical; if FALSE (default), it prints the number of unique sequences found and the name of the output file.

Value

If "alpha" is a single value, this function generates a data frame containing the estimated combination of substitution and indel distance matrices. If "alpha" is a vector of values, this function generates a list of data frames.

Author(s)

A. J. Muñoz-Pajares

See Also

`MCIC`,`BARRIEL`,`SIC`,`FIFTH`

Examples

```
# cat(">Population1_sequence1",
# "TTATAAAATCTA----TAGC",
# ">Population1_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population1_sequence3",
# "TTATAAAAATTA----TAGC",
# ">Population1_sequence4",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence1",
# "TTAT----TCGAGGGGTAGC",
# ">Population2_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence3",
# "TTATAAAA--------TAGC",
# ">Population2_sequence4",
# "TTAT----TCGAGGGGTAGC",
# ">Population3_sequence1",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence2",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence3",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence4",
# "TTAT----TCGA----TAGC",
#      file = "ex2.fas", sep = "\n")
#
#  # Estimating indel distances after reading the alignment from file:
# distGap<-MCIC(input="ex2.fas",saveFile=FALSE)
#  # Estimating substitution distances after reading the alignment from file:
# library(ape)
# align<-read.dna(file="ex2.fas",format="fasta")
# dist.nt<-dist.dna(align,model="raw",pairwise.deletion=TRUE)
# DISTnt<-as.matrix(dist.nt)
#  # Obtaining 11 corrected combined matrices using a range of alpha values:
# nt.gap.comb(DISTgap=distGap, alpha=seq(0,1,0.1), method="Corrected",
# saveFile=FALSE, DISTnuc=DISTnt)
#  # Obtaining the arithmetic mean of both matrices using both the corrected
#  # and the uncorrected methods.
# nt.gap.comb(DISTgap=distGap, alpha=0.5, method="Uncorrected", saveFile=FALSE,
#  DISTnuc=DISTnt)
#  # Obtaining a range of combinations...
# Range01<-nt.gap.comb(DISTgap=distGap, alpha=seq(0,1,0.1), method="Uncorrected",
#  saveFile=FALSE, DISTnuc=DISTnt)
#  # ...and displaying the arithmetic mean (alpha=0.5 is the element number 6
#  # in the resulting data frame):
# Range01[[6]]
```

perc.thr *Percolation threshold network*

Description

This function computes the percolation network following Rozenfeld et al. (2008), as described in Muñoz-Pajares (2013).

Usage

```
perc.thr(dis, range = seq(0, 1, 0.01), ptPDF = TRUE,
ptPDFname = "PercolatedNetwork.pdf", estimPDF = TRUE,
estimPDFname = "PercThr Estimation.pdf", estimOutfile = TRUE,
estimOutName = "PercThresholdEstimation.txt", cex.label = 1,
cex.vertex = 1, appendOutfile = TRUE, plotALL = FALSE,
bgcol = "white", label.col = "black", label = colnames(dis),
modules = FALSE, moduleCol = NA, modFileName = "Modules_summary.txt",
ncs = 4, na.rm.row.col = FALSE, merge = FALSE,save.distance = FALSE,
save.distance.name = "DistanceMatrix_Perc.thr.txt")
```

Arguments

`dis` the distance matrix to be represented

`range` a numeric vector between 0 and 1, is the range of thresholds (referred to the maximum distance in a matrix) to be screened (by default, 101 values from 0 to 1).

`ptPDF` a logical, must the percolated network be saved as a pdf file?

`ptPDFname` if ptPDF=TRUE, the name of the pdf file containing the percolation network to be saved ("PercolatedNetwork.pdf", by default)

`estimPDF` a logical, must the percolation threshold estimation be saved as a pdf file?

`estimPDFname` if estimPDF=TRUE (default), defines the name of the pdf file containing the percolation threshold estimation ("PercThr Estimation.pdf" by default).

`estimOutfile` a logical, must the value of <s> for each threshold be saved as a text file?

`estimOutName` if estimOutfile=TRUE (default), contains the name of the text file containing the percolation threshold estimation ("PercThr Estimation.txt" by default).

`cex.label` a numeric; the size of the node labels.

`cex.vertex` a numeric; the size of the nodes.

`appendOutfile` a logical, if estimOutfile=TRUE, it defines whether results must be appended to an existing file with the same name (TRUE) or the existing file must be replaced (FALSE).

`plotALL` a logical, must all the networks calculated during the percolation threshold estimation (defined by "range" option) be saved as different pdf files? (FALSE, by default). If TRUE, for each value defined in threshold, one file is generated.

`bgcol` the colour of the background for each node in the network. Can be equal for all nodes (if only one colour is defined), customized (if several colours are defined), or can represent different modules (see "modules" option).

`label.col` the colour of labels for each node in the network. Can be equal for all nodes (if only one colour is defined) or customized (if several colours are defined).

`label` a vector of strings, labels for each node. By default are the column names of the distance matrix (dis). (See "substr" function in base package to automatically reduce name lengths).

`modules` a logical, must nodes belonging to different modules be represented as different colours?

`moduleCol` (if modules=TRUE) a vector of strings defining the colour of nodes belonging to different modules in the network. If 'NA' (or there are less colours than haplotypes), colours are automatically selected.

`modFileName` (if modules=TRUE) the name of the file to be generated containing a summary of module results (sequence name, module, and colour in network)

`ncs` a numeric; number of decimal places to display threshold in plot title.

`na.rm.row.col` a logical; if TRUE, missing values are removed before the computation proceeds.

`merge` a logical, if TRUE, merges rows (and columns) showing distance values equal to zero.

`save.distance` a logical; if TRUE, the new distance matrix will be saved in a file.

`save.distance.name`
a string; if save.distance is set to TRUE, it defines the name of the file to be saved.

Details

By default, percolation threshold is estimated with an accuracy of 0.01, but it may be increased by setting the decimal places in threshold function (e.g., range=seq(0,1,0.0001)). However, it may strongly increase computation times (in this example, it is required to estimate 100,001 instead of 101 networks). It is also possible to increase accuracy with a low increase in computation time by repeating the process and increasing decimal places only in a range close to a previously estimated percolation threshold. For example, if the estimated percolation threshold is 0.48, it is possible to define a second round using range=seq(0.47,0.49,0.0001), which provide an accurary of 0.0001 estimating only 201 networks.

'perc.thr' estimates a threshold to represent a distance matrix as a network. To estimate this threshold, the algorithm represents as a link all distances lower than a range of thresholds (by default, select 101 values from 0 to 1), defined as the percentage of the maximum distance in the input matrix. For each threshold a network is built and the number of clusters (that is, the number of isolated groups of nodes) in the network is also estimated. Finally, the algorithm selects the lower threshold connecting a higher number of nodes. Note that the resulting network may show isolated nodes if it is necessary to represent a large number of links to connect a low number of nodes.

Author(s)

A. J. Muñoz-Pajares

References

Rozenfeld AF, Arnaud-Haond S, Hernandez-Garcia E, Eguiluz VM, Serrao EA, Duarte CM. (2008). Network analysis identifies weak and strong links in a metapopulation system. Proceedings of the National Academy of Sciences,105, 18824-18829.

Muñoz-Pajares, A.J. (2013). SIDIER: substitution and indel distances to infer evolutionary relationships. Methods in Ecology and Evolution, 4, 1195-1200

See Also

`single.network`, `NINA.thr`, `zero.thr`, `mergeNodes`

Examples

```
# cat(">Population1_sequence1",
# "TTATAAAATCTA----TAGC",
# ">Population1_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population1_sequence3",
# "TTATAAAAATTA----TAGC",
# ">Population1_sequence4",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence1",
# "TTAT----TCGAGGGGTAGC",
# ">Population2_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence3",
# "TTATAAAA--------TAGC",
# ">Population2_sequence4",
# "TTAT----TCGAGGGGTAGC",
# ">Population3_sequence1",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence2",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence3",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence4",
# "TTAT----TCGA----TAGC",
#      file = "ex2.fas", sep = "\n")
#
#  # Estimating indel distances after reading the alignment from file:
# distGap<-MCIC(input="ex2.fas",saveFile=FALSE)
#  # Estimating substitution distances after reading the alignment from file:
# library(ape)
# align<-read.dna(file="ex2.fas",format="fasta")
# dist.nt <-dist.dna(align,model="raw",pairwise.deletion=TRUE)
# DISTnt<-as.matrix(dist.nt)
#
#
#  # Obtaining the arithmetic mean of both matrices using the corrected method:
# CombinedDistance<-nt.gap.comb(DISTgap=distGap, alpha=0.5, method="Corrected",
# saveFile=FALSE, DISTnuc=DISTnt)
#  # Estimating the percolation threshold of the combined distance, modifying
#  # labels:
# perc.thr(dis=CombinedDistance,label=paste(substr(row.names(
# CombinedDistance),11,11),substr(row.names(CombinedDistance),21,21),sep="-"))
#
#  # The same network showing different modules as different colours
#  # (randomly selected):
# perc.thr(dis=as.data.frame(CombinedDistance),label=paste(substr(row.names(
# as.data.frame(CombinedDistance)),11,11),substr(row.names(as.data.frame(
# CombinedDistance)),21,21),sep="-"), modules=TRUE)
#
#  # The same network showing different modules as different colours
#  # (defined by user):
# perc.thr(dis=as.data.frame(CombinedDistance),label=paste(substr(row.names(
# as.data.frame(CombinedDistance)),11,11),substr(row.names(as.data.frame(
# CombinedDistance)),21,21),sep="-"), modules=TRUE,moduleCol=c("pink",
# "lightblue","lightgreen"))
#
```

pie.network *Population network depiction including haplotype frequencies*

Description

This function represents an alignment as a population network and displays nodes as pie charts where haplotype frequencies are proportional to the area depicted in different colours.

Usage

```
pie.network(align = NA, indel.method = "MCIC", substitution.model = "raw",
pairwise.deletion = TRUE, network.method = "percolation",
range = seq(0, 1, 0.01), addExtremes = FALSE, alpha = "info",
combination.method = "Corrected", na.rm.row.col = FALSE,
NameIniPopulations = NA, NameEndPopulations = NA, NameIniHaplotypes = NA,
NameEndHaplotypes = NA, save.distance = FALSE,
save.distance.name = "DistanceMatrix_threshold.txt",
pop.distance.matrix = NULL, Haplos = NULL, HaplosPerPop = NULL,
col.pie = NA, label.col = "black", label = NA, label.sub.str = NA,
cex.label = 1, cex.pie = 1, main = "", HaplosNames = NA,
offset.label = 1.5, pie.size = "equal", coord = NULL, get.coord = TRUE)
```

Arguments

`align` a 'DNAbin' object; the alignment to be analysed. See "read.dna" in the ape package for details about reading alignments. Other inputs are available: Use a distance matrix instead an alignment using the 'align' option or provide a list of haplotypes and frequencies per population using 'Haplos' and 'HapPerPop' options

`indel.method` a sting; the method to define indel events in your alignments. The available methods are:

-"MCIC": (Default) Estimates indel events following the rationale of the Modified Complex Indel Coding (Muller, 2006).

-"SIC": Estimates indel events following the rationale of Simmons and Ochoterrena (2000).

-"FIFTH": Estimates indel events following the rationale of the fifth state: each gap within the alignment is treated as an independent mutation event.

-"BARRIEL": Estimates indel events following the rationale of Barriel (1994): singleton gaps are not taken into account.

`substitution.model` a string; the substitution evolutionary model to estimate the distance matrix. By default is set to "raw" and estimates the pairwise proportion of variant sites. See the evolutionary models available using ?dist.dna from the ape package.

`pairwise.deletion` a logical; if TRUE (default) substitutions found in regions being a gap in other sequences will account for the distance matrix. If FALSE, sites being a gap in at least one sequence will be removed before distance estimation.

`network.method` a string; the method to build the network. The available methods are:

-"percolation": computes a network using the percolation network method following Rozenfeld et al. (2008). See ?perc.thr for details

-"NINA": computes a network using the No Isolation Nodes Allowed method. See ?NINA.thr for details.

-"zero": computes a network connecting all nodes showing distances equal to zero. See ?zero.thr for details.

`range` a numeric vector between 0 and 1, is the range of thresholds (referred to the maximum distance in the input matrix) to be screened (by default, 101 values from 0 to 1). This option is used for "percolation" and "NINA" network methods and ignored for "zero" method.

`addExtremes` a logical; if TRUE, additional nucleotide sites are included in both extremes of the alignment. This will allow estimating distances for alignments showing gaps in terminal positions. This option is used for "SIC", "FIFTH" and "BARRIEL" indel methods and ignored for "MCIC" method.

`alpha` a numeric between 0 and 1, is the weight given to the indel genetic distance matrix in the combination. By definition, the weight of the substitution genetic matrix is the complementary value (i.e., 1-alpha). The value "info" will use the proportion of informative substitutions per informative indel event as weight. It is also possible to define multiple weights to estimate different combinations.

`combination.method`

a string defining whether each distance matrix must be divided by its maximum value before the combination ("Corrected") or not ("Uncorrected"). Consequently, if the "Corrected" method is chosen (default option), both matrices are corrected to range between 0 and 1 before being combined.

`na.rm.row.col` a logical; if TRUE, removes rows and columns showing missing values within the distance matrix.

`NameIniPopulations`

a numeric; Position of the initial character of population names within sequence names. If not provided, it is set to 1. It is used only if NameEndPopulations is also defined.

`NameEndPopulations`

a numeric; Position of the last character of population names within sequence names. If not provided, it is set to the first "_" character in the sequences name. It is used only if NameIniPopulations is also defined.

`NameIniHaplotypes`

a numeric; Position of the initial character of haplotype names within sequence names. If not provided, haplotype names are given and the value is set accordingly. It is used only if NameEndHaplotypes is also defined.

`NameEndHaplotypes`

a numeric; Position of the last character of haplotype names within sequence names. If not provided, haplotype names are given and the value is set accordingly. It is used only if NameIniHaplotypes is also defined.

`save.distance` a logical; if TRUE, the distance matrix used to build the network will be saved as a file.

`save.distance.name`

a string; if save.distance=TRUE, the name of the file to be saved.

`pop.distance.matrix`

a matrix containing the population distances. Alternatively, it can be estimated from a given sequence alignment using 'align'. Alternatively, you can provide a list of haplotypes and frequencies using 'Haplos' and 'HapPerPop'

`Haplos` a two columns matrix containing sequence names and haplotype names as reported by `FindHaplo`. Alternatively, you can define an input alignment using 'align' or a distance matrix using 'pop.distance.matrix'.

`HaplosPerPop` a matrix containing the number of haplotypes found per population, as reported by `HapPerPop` (Weighted matrix).Alternatively, you can define an input alignment using 'align' or a distance matrix using 'pop.distance.matrix'.

`col.pie` a vector of strings; the colour to represent each haplotype. If 'NA' (or there are less colours than haplotyes), colours are automatically selected.

`label.col` a vector of strings; the colour of labels for each node in the network. Can be equal for all nodes (if only one colour is defined) or customized (if several colours are defined).

`label` a vector of strings; labels for each node. By default are the sequence names. (See "substr" function in base package to automatically reduce name lengths)

`label.sub.str` a vector of two numerics; if node labels are a substring of sequence names, these two numbers represent the initial and final character of the string to be represented. See Example for details.

`cex.label` a numeric; the size of the node labels.

`cex.pie` a numeric; the size of the nodes (pie charts).

`main` a sting; if set to "summary" the main options selected for representing the network are displayed in title. The default value ("") shows no title for the network.

`HaplosNames` a sting; the name of the haplotypes (if different from default: H1...Hn)

`offset.label` a numeric; the separation between node and label.

`pie.size` a string to define the ratio of pies representing populations. Possible values are: "equal" (default) to give the same size to all pies; "radius" to make the pie radius proportional to the population sample size; "area" to make the pie area proportional to the population sample size; or "points" to display simple vertices instead of pies representing haplotypes per population.

`coord` a two columns matrix containing coordinates where each haplotypes must be represented.

`get.coord` a logical, TRUE to obtain coordinates of nodes within the network

Details

It is recommended to use equal length names with population and individual names separated by '_' (e.g., Pop01_id001...Pop23_id107) and set population (both, NameIniPopulations and NameEndPopulations,) and haplotype (both, NameIniHaplotypes and NameIniHaplotypes) identifiers accordingly. If any of these identifiers is not provided, the algorithm will behave as follows:

-If only haplotype name identifiers are defined, population names are assumed between character 1 and the first symbol '_' in sequences name.

-If only population name identifiers are defined, haplotype are automatically found and named using the 'HapPerPop' function.

-If both are not defined, population names are assumed between character 1 and the first symbol '_' in sequences name and haplotypes are automatically found and named using the 'HapPerPop' function.

Author(s)

A. J. Muñoz-Pajares

References

Barriel, V., 1994. Molecular phylogenies and how to code insertion/ deletion events. Life Sci. 317, 693-701, cited and described by Simmons, M.P., Müller, K. & Norton, A.P. (2007) The relative performance of indel-coding methods in simulations. Molecular Phylogenetics and Evolution, 44, 724–740.

Muller K. (2006). Incorporating information from length-mutational events into phylogenetic analysis. Molecular Phylogenetics and Evolution, 38, 667-676.

Paradis, E., Claude, J. & Strimmer, K. (2004). APE: analyses of phylogenetics and evolution in R language. Bioinformatics, 20, 289-290.

Rozenfeld AF, Arnaud-Haond S, Hernandez-Garcia E, Eguiluz VM, Serrao EA, Duarte CM. (2008). Network analysis identifies weak and strong links in a metapopulation system. Proceedings of the National Academy of Sciences,105, 18824-18829.

Simmons, M.P. & Ochoterena, H. (2000). Gaps as Characters in Sequence-Based Phylogenetic Analyses. Systematic Biology, 49, 369-381.

See Also

mutation.network, double.plot

Examples

```
# cat(">Population1_sequence1",
# "TTATAAAATCTA----TAGC",
# ">Population1_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population1_sequence3",
# "TTATAAAAATTA----TAGC",
# ">Population1_sequence4",
# "TAAT----TCTA----TAAC",
# ">Population2_sequence1",
# "TTAT----TCGA----TAGC",
# ">Population2_sequence2",
# "TTAT----TCGA----TAGC",
# ">Population2_sequence3",
# "TTAT----TCGA----TAGC",
# ">Population2_sequence4",
# "TTAT----TCGA----TAGC",
# ">Population3_sequence1",
# "TTAT----TCGAGGGGTAGC",
# ">Population3_sequence2",
# "TAAT----TCTA----TAAC",
# ">Population3_sequence3",
# "TTATAAAA--------TAGC",
# ">Population3_sequence4",
# "TTAT----TCGAGGGGTAGC",
#      file = "ex2.fas", sep = "\n")
# library(ape)
# example<-read.dna(file="ex2.fas",format="fasta")
#
# # The input format is recognized, and names identifiers can be omitted:
#   pie.network(align=example)
#
# # Is identical to:
#   pie.network(align=example, NameIniPopulations=1,NameEndPopulations=11)
#
# # Using different colours:
#   pie.network(align=example, NameIniPopulations=1,NameEndPopulations=11,
#   col.pie=c("red","blue","pink","orange","black","grey"))
#
# # col.pie is omitted if less colours than haplotypes are defined:
#   pie.network(align=example, NameIniPopulations=1,NameEndPopulations=11,
#   col.pie=c("red","blue","pink"))
#
# # and also if more colours than haplotypes are defined:
#    pie.network(align=example, NameIniPopulations=1,NameEndPopulations=11,
#   col.pie=c("red","blue","green","purple","pink","orange","gray"))
#
```

pop.dist *Distances among populations*

Description

This function computes the population pairwise distance matrix based on the frequency of haplotypes per population and the haplotypes pairwise distance matrix. It is mandatory to define haplotype and population names in the input file. See example for details.

Usage

```
pop.dist(DistFile = NA, distances = NA, HaploFile = NA, Haplos = NA,
 outType = "O", logfile = TRUE, saveFile = TRUE, NameIniPopulations
 = NA, NameEndPopulations = NA, NameIniHaplotypes = NA,
 NameEndHaplotypes = NA)
```

Arguments

`DistFile` the name of the file containing the distance matrix among haplotypes to be analysed. Alternatively, you can define a distance matrix stored in memory using 'distances'.

`distances` the distance matrix among haplotypes (stored in memory) to be analysed. Alternatively, you can define the name of a file containing the distance matrix using 'DistFile'.

HaploFile
: the name of the file containing the matrix with the number of haplotypes found per population (see 'HapPerPop' to obtain this matrix). Alternatively, you can define a matrix stored in memory using 'Haplos'.

Haplos
: the name of the matrix (stored in memory) containing the number of haplotypes found per population (see 'HapPerPop' to obtain this matrix). Alternatively, you can define the name of a file containing the matrix using 'HaplosFile'.

outType
: a string; the format of output matrix. "L" for lower diagonal hemi-matrix; "7" for upper diagonal hemi-matrix; "O" for both hemi-matrices (default).

logfile
: a logical; if TRUE (default), it saves a file containing the names of the matrices used for computation (inputDist and HaploFile).

saveFile
: a logical; if TRUE (default), function output is saved as a text file.

NameIniPopulations
: a numeric indicating the position of the initial character of population names within the individual name in the matrix containing the number of haplotypes found per population (see example for details).

NameEndPopulations
: a numeric indicating the position of the last character of population names within the individual name in the matrix containing the number of haplotypes found per population (see example for details). If NA (default), NameIniPopulations and NameEndPopulations are set to use the 'Haplos' (or HaploFile) matrix row names as population names.

NameIniHaplotypes
: a numeric indicating the position of the initial character of haplotype names within the individual name in the distance matrix (see example for details).

NameEndHaplotypes
: a numeric indicating the position of the last character of haplotype names within the individual name in the distance matrix (see example for details). If NA (default), NameIniHaplotypes and NameEndHaplotypes are set to use the 'distances' (or DistFile) matrix row names as haplotype names.

Details

Each element in the population distance matrix is calculated as the arithmetic mean of the distances among all the sequences sampled in the two compared populations, as follows:

$$dist(i,j) = \frac{\sum_{k=1}^{m} \sum_{l=1}^{n} dist(H_{ki}, H_{lj})}{m * n}$$

where *dist(i,j)* represents the distance between populations *i* and *j*, *m* and *n* are the number of sequences in populations *i* and *j*, respectively, and *dist(H_ki,H_lj)* is the distance between the *k-th* sequence found in population *i* and the *l-th* sequence found in population *j*.

Value

A matrix containing the genetic distances among populations, based on the haplotype distances and their frequencies per populations.

Author(s)

A. J. Muñoz-Pajares

Examples

```
# cat(” H1 H2 H3 H4 H5”,
# ”Population1 1 2 1 0 0”,
# ”Population2 0 0 0 4 1”,
# ”Population3 0 1 0 0 3”,
#      file = ”4_Example3_HapPerPop_Weighted.txt”, sep = ”\n”)
#
#
# cat(”H1 H2 H3 H4 H5”,
# ”H1 0 1 2 3 1”,
# ”H2 1 0 3 4 2”,
# ”H3 2 3 0 1 1”,
# ”H4 3 4 1 0 2”,
# ”H5 1 2 1 2 0”,
#      file = ”4_Example3_IndelDistanceMatrixMullerMod.txt”, sep = ”\n”)
#      example3_2 <- read.table(”4_Example3_IndelDistanceMatrixMullerMod.txt”
# ,header=TRUE)
#
# # Checking row names to estimate NameIniHaplotypes,NameEndHaplotypes:
#  row.names(read.table(file=”4_Example3_IndelDistanceMatrixMullerMod.txt”))
# ## [1] ”H1” ”H2” ”H3” ”H4” ”H5” NameIniHaplotypes=1. NameEndHaplotypes=2
# # Checking row names to estimate NameIniPopulations, and NameEndPopulations
#  row.names(read.table(file=”4_Example3_HapPerPop_Weighted.txt”))
# ## [1] ”Population1” ”Population2” ”Population3”
# ## NameIniPopulations=1 NameEndPopulations =11
#
# # Reading files. Distance matrix must contain haplotype names. Abundance
# # matrix must contain both, haplotype and population names:
#
# pop.dist (DistFile=”4_Example3_IndelDistanceMatrixMullerMod.txt”,
# HaploFile=”4_Example3_HapPerPop_Weighted.txt”, outType=”O”,
# NameIniHaplotypes=1,NameEndHaplotypes=2,NameIniPopulations=1,
# NameEndPopulations=11)
```

rule *Threshold to discriminate species.*

Description

Threshold to discriminate species showing a ratio interspecific/intraspecific distances higher than a given value.

Usage

```
rule(summary=NULL,rule=NULL,stat.intra="max",
stat.inter="min",pch.intra=16, pch.inter=16,
pch.out=21,col.intra="gray",col.inter="black",
col.out="black",label=F)
```

Arguments

`summary` a list produced by `barcode.summary`. From this list, the maximum intraspecific and the minimu interspecific distances per species are represented. To use any other intra- and interspecific distance, use the "inter" and "intra" options.

`rule` a numeric. Only species showing interspecific distances higher than 'rule' times the intraspecific distances will be considered for threshold estimation.

`stat.intra` a string, the inter-specific statistic used to estimate the quotient interspecific/intraspecific. Accepted values are "max", "min", "median", and "mean"

`stat.inter` a string, the inter-specific statistic used to estimate the quotient interspecific/intraspecific. Accepted values are "max", "min", "median", and "mean".

`pch.intra` Either an integer or single character defining the symbol to be used for intraspecific distances. Only species showing a ratio interspecific/intraspecific higher than the value defined by 'rule' are affected by 'pch.intra'.

`pch.inter` Either an integer or single character defining the symbol to be used for interspecific distances Only species showing a ratio interspecific/intraspecific higher than the value defined by 'rule' are affected by 'pch.inter'.

`pch.out` Either an integer or single character defining the symbol to be used for species showing a ratio interspecific/intraspecific lower than the value defined by 'rule'.

`col.intra` Either an integer or string defining the colour for intraspecific distances showing a ratio interspecific/intraspecific higher than the value defined by 'rule'.

`col.inter` Either an integer or string defining the colour for interspecific distances showing a ratio interspecific/intraspecific higher than the value defined by 'rule'.

`col.out` Either an integer or string defining the colour forspecies showing a ratio interspecific/intraspecific lower than the value defined by 'rule'.

`label` a string to set node labels on those species showing a ratio interspecific/intraspecific higher than the value defined ("rule"), on species showing a ratio interspecific/intraspecific higher than the value defined ("norule"), on all species ("all"). Any other value will produce no lable

Value

A list with two elements:

`Intraspecific` a matrix containing information about the intraspecific distances.

`Interspecific` a matrix containing information about the interspecific distances.

In both cases, the information provided is the minimum, maximum, median, mean, first and third quartile values.

Author(s)

A.J. Muñoz-Pajares

Examples

```
my.dist<-matrix(c(0,0.3,0.24,0.45,0.23,0.01,0.11,0.34,0.64,0.34,
0.3,0,0.32,0.75,0.65,0.53,0.012,0.52,0.15,0.52,0.24,0.32,
0,0.92,0.36,0.62,0.85,0.008,0.82,0.65,0.45,0.75,0.92,0,
0.22,0.56,0.74,0.46,0.005,0.73,0.23,0.65,0.36,0.22,0,
0.34,0.24,0.42,0.35,0.009,0.01,0.53,0.62,0.56,0.34,0,
0.23,0.73,0.23,0.63,0.11,0.012,0.85,0.74,0.24,0.23,0,
0.25,0.63,0.54,0.34,0.52,0.008,0.46,0.42,0.73,0.25,0,
0.32,0.41,0.64,0.15,0.82,0.005,0.35,0.23,0.63,0.32,0,
0.23,0.34,0.52,0.65,0.73,0.009,0.63,0.54,0.41,0.23,0),
ncol=10,dimnames=list(paste("sp",rep(1:5,2),sep=""),
paste("sp",rep(1:5,2),sep="")))

# rule(barcode.summary(my.dist),rule=10)
```

SIC *Indel distances following the Simple Index Coding method*

Description

This function codifies gapped positions in a sequence alignment following the rationale of the method described by Simmons and Ochoterrena (2000). Based on the yielded indel coding matrix, this function also computes a pairwise indel distance matrix.

Usage

```
SIC(inputFile = NA, align = NA, saveFile = TRUE,
outnameDist=paste(inputFile,"IndelDistanceSIC.txt",
sep = "_"), outnameCode = paste(inputFile,
"SIC_coding.txt", sep = "_"), addExtremes = FALSE)
```

Arguments

inputFile the name of the fasta file to be analysed. Alternatively you can provide the name of a "DNAbin" class alignment stored in memory using the "align" option.

align the name of the alignment to be analysed. See "read.dna" in ape package for details about reading alignments. Alternatively you can provide the name of the file containing the alignment in fasta format using the "inputFile" option.

saveFile a logical; if TRUE (default), it produces two output text files containing the distance matrix and the codified indel positions.

outnameDist if "saveFile" is set to TRUE (default), contains the name of the distance output file.

outnameCode if "saveFile" is set to TRUE (default), contains the name of the indel coding output file.

addExtremes a logical; if TRUE, additional nucleotide sites are included in both extremes of the alignment. This will allow estimating distances for alignments showing gaps in terminal positions, but see Details.

Details

It is recommended to estimate this distance matrix using only the unique sequences in the alignment. Repeated sequences increase computation time but do not provide additional information (because they produce duplicated rows and columns in the final distance matrix).

Value

A list with two elements:

indel coding matrix
Describes the initial and final site of each gap and its presence or absence per sequence.

distance matrix
Contains genetic distances based on comparing indel presence/absence between sequences.

Author(s)

A. J. Muñoz-Pajares

References

Simmons, M.P. & Ochoterena, H. (2000). Gaps as Characters in Sequence-Based Phylogenetic Analyses. Systematic Biology, 49, 369-381.

See Also

BARRIEL, MCIC, FIFTH

Examples

```
# # This will generate an example file in your working directory:
# cat(">Population1_sequence1",
# "A-AGGGTC-CT---G",
# ">Population1_sequence2",
# "TAA---TCGCT---G",
# ">Population1_sequence3",
# "TAAGGGTCGCT---G",
# ">Population1_sequence4",
# "TAA---TCGCT---G",
# ">Population2_sequence1",
# "TTACGGTCG---TTG",
# ">Population2_sequence2",
# "TAA---TCG---TTG",
# ">Population2_sequence3",
# "TAA---TCGCTATTG",
# ">Population2_sequence4",
# "TTACGGTCG---TTG",
# ">Population3_sequence1",
# "TTA---TCG---TAG",
# ">Population3_sequence2",
# "TTA---TCG---TAG",
# ">Population3_sequence3",
# "TTA---TCG---TAG",
# ">Population3_sequence4",
# "TTA---TCG---TAG",
#      file = "ex3.fas", sep = "\n")
# library(ape)
# SIC (align=read.dna("ex3.fas",format="fasta"), saveFile = FALSE)
#
# # Analysing the same dataset, but using only unique sequences:
# uni<-GetHaplo(inputFile="ex3.fas",saveFile=FALSE)
# SIC (align=uni, saveFile = FALSE)
```

---

simplify.network *Network showing modules as nodes*

Description

This function modifies node coordinates to allow a clearer depiction of complex networks. Nodes are moved along the axis connecting the original position to the module centroid. The magnitude of such movement is defined by user.

Usage

```
simplify.network(node.names=NA,modules=NA,coordinates=NA,network=NA,
shift = 0.5,max.lwd.edge =2,min.lwd.edge =1,max.vertex.size=4,
min.vertex.size=2,label.size=1/2.5,bgcol="white",main="")
```

Arguments

`node.names`	a vector containing the names of nodes
`modules`	a vector containing the module assigned to each node
`coordinates`	a two columns matrix containing the X and Y coordinates of each node in the original network
`network`	a matrix describing the original network. Can be either a 0/1 matrix or a weighted matrix. Row names must contain node names.
`shift`	a numeric defining the magnitude of node shift,limited between 0 (coinciding with the original location) and 1 (coinciding with the module centroid location).
`max.lwd.edge`	if shift=1,a numeric defining the line width for the maximum number of connections between modules

`min.lwd.edge` if shift=1,a numeric defining the line width for the minimum number of connections between modules

`max.vertex.size` if shift=1,a numeric defining the size of the node representing the largest module

`min.vertex.size` if shift=1,a numeric defining the size of the node representing the smallest module

`label.size` a numeric defining the size of node labels,referred to its particular node size

`bgcol` a vector of strings representing the background colour for each node

`main` a string,the title for the plot (no title by default)

Details

If 'shift=1',all nodes belonging to a module are represented as a single node depicted in the module centroid. In that case,node size is proportional to the number of element in this module and edge widths are proportional to the number of connections found between modules.

Author(s)

A. J. Muñoz-Pajares

Examples

```
#
# inputMatrix<-matrix(c(1,1,1,1.2,2,1,0.8,1,3,2,1.2,1,4,2,2,2.2,
#  5,3,1.8,2,6,3,2.2,2,7,3,1.7,2.1,8,3,2.2,2.2),ncol=4,byrow=TRUE)
# colnames(inputMatrix)<-c("node","module","x","y")
#
# network<-matrix(c(1,1,0,0,1,1,0,0,1,1,1,0,0,0,0,0,0,
#   1,1,1,0,0,0,0,0,0,1,1,0,0,0,0,1,0,0,0,1,1,
#   1,1,1,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,
#   0,1,1,1,1),ncol=8)
# colnames(network)<-c(1:8)
# row.names(network)<-c(1:8)
#
# i1<-0
# simplify.network(node.names=inputMatrix[,1],modules=inputMatrix[,2],
# coordinates=inputMatrix[,3:4],network=network,shift = i1,
# bgcol=c("red","red","blue","blue","green","green","green","green"),
# main=paste("shift=",i1))
#
# i1<-0.5
# simplify.network(node.names=inputMatrix[,1],modules=inputMatrix[,2],
# coordinates=inputMatrix[,3:4],network=network,shift = i1,
# bgcol=c("red","red","blue","blue","green","green","green","green"),
# main=paste("shift=",i1))
#
# i1<-1.0
# simplify.network(node.names=inputMatrix[,1],modules=inputMatrix[,2],
# coordinates=inputMatrix[,3:4],network=network,shift = i1,
# bgcol=c("red","red","blue","blue","green","green","green","green"),
# main=paste("shift=",i1))
#
# network<-as.matrix(as.dist(matrix(sample(c(1,0),10000,replace=TRUE),ncol=100)))
# inputMatrix<-matrix(nrow=100,ncol=4)
# inputMatrix[,1]<-1:100
# inputMatrix[,2]<-c(rep(1,30),rep(2,20),rep(3,20),rep(4,20),rep(5,10))
# inputMatrix[,3]<-c(
# sample(seq(-40,0,0.01),30,rep=TRUE),
# sample(seq(-40,0,0.01),20,rep=TRUE),
# sample(seq(0,40,0.01),20,rep=TRUE),
# sample(seq(0,40,0.01),20,rep=TRUE),
# sample(seq(-20,20,0.01),10,rep=TRUE))
# inputMatrix[,4]<-c(
# sample(seq(0,40,0.01),30,rep=TRUE),
# sample(seq(-40,0,0.01),20,rep=TRUE),
# sample(seq(0,40,0.01),20,rep=TRUE),
# sample(seq(-40,0,0.01),20,rep=TRUE),
# sample(seq(-20,20,0.01),10,rep=TRUE))
# cols<-c("red","green","yellow","blue","turquoise")
#
# simplify.network(node.names=inputMatrix[,1],network=network,shift=0,
# coordinates=inputMatrix[,3:4],modules=inputMatrix[,2],bgcol=cols[inputMatrix[,2]])
#
# simplify.network(node.names=inputMatrix[,1],network=network,shift=1,
# coordinates=inputMatrix[,3:4],modules=inputMatrix[,2],bgcol=cols[inputMatrix[,2]])
#
```

simuEvolution *Simulate sequences evolution*

Description

This function simulates the evolution of a set of sequences. It is necessary to define evolution topology, substitution rate, indel rate and insertion/deletion rate in a matrix (see details).

Usage

```
simuEvolution(input, seqL, iLength, nReplicates)
```

Arguments

input	Matrix defining evolution topology and mutation rates.
seqL	Length of the simulated sequences.
iLength	Length of indel mutations.
nReplicates	Number of independent sequence sets to be simulated.

Details

Evolution details must be provided in a file consisting in five columns separated by spaces. The first two columns define topology by indicating the ancestor and the derived sequence, respectively. The remaining columns provide rates for substitutions and indels as well as the ratio between insertions and deletions. The simulation is performed over the complete alignment. To test the effect of alignment method over the simulated sequences it will be necessary to degap the yielded sequences.

Value

For each replicate, two files are generated: one containing the alignment with all the generated sequences and the other containing only tips sequences (i.e., sequences that are not the ancestor of any other sequence).

Author(s)

A. J. Muñoz-Pajares

Examples

```
#Generating matrix defining evolution:
Input<-matrix(c(1,rep(2:8,2),2:16,rep(0.03,15),rep(0.008,15),rep(0.5,15)),ncol=5)
#Simulating 2 replicates of the evolutionary process:
# simuEvolution(input=Input, seqL=1000, iLength=20, nReplicates=2)
```

---

single.network *Plot a network given a threshold*

Description

This function plots a network connecting nodes showing distances equal or lower than the defined threshold value.

Usage

```
single.network(dis, threshold = NA, ptPDF = TRUE, ptPDFname = "Network.pdf",
bgcol = "white", label.col = "black", label = colnames(dis), modules = FALSE,
moduleCol = NA, modFileName = "Modules_summary.txt", na.rm.row.col = FALSE,

cex.vertex = 1, plot = TRUE, get.coord = FALSE, refer2max = TRUE

)
```

Arguments

`dis` the distance matrix to be represented

`threshold` a numeric between 0 and 1, is the value of the maximum distance to be considered as a link. This value is referred to the maximum distance in the input matrix (e.g., a value of 0.32 will represent a link between nodes showing distances equal or lower than 32% of the maximum distance found in the distance matrix).

`ptPDF` a logical, must the percolation network be saved as a pdf file?

`ptPDFname` if ptPDF=TRUE, the name of the pdf file containing the percolation network to be saved ("percolationNetwork.pdf", by default)

`bgcol` string defining the colour of the background for each node in the network. Can be equal for all nodes (if only one colour is defined), customized (if several colours are defined), or can represent different modules (see modules option).

`label.col` vector of strings defining the colour of labels for each node in the network. Can be equal for all nodes (if only one colour is defined) or customized (if several colours are defined).

`label` vector of strings, labels for each node. By default are the column names of the distance matrix (dis). (See substr function in base package to automatically reduce name lengths).

`modules` a logical, must nodes belonging to different modules be represented as different colours?

`moduleCol` (if modules=TRUE) vector of strings, defining the colour of nodes belonging to different modules in the network.

`modFileName` (if modules=TRUE) the name of a generated file containing a summary of module results

`na.rm.row.col` a logical; if TRUE, missing values are removed before the computation proceeds.

`cex.vertex` a numeric, the size of vertex

`plot` a logical, TRUE to plot the inferred network

`get.coord` a logical, TRUE to obtain coordinates of nodes within the network

`refer2max` a logic, "TRUE" to refer the threshold value to the maximum distance in the input matrix (e.g., a value of 0.32 will represent a link between nodes showing distances equal or lower than 32% of the maximum distance found in the distance matrix). "FALSE" to refer the threshold to a specific value (e.g., a value of 0.32 will represent a link between nodes showing distances equal or lower than 0.32, regardless the maximum distance found in the distance matrix).

Author(s)

A. J. Muñoz-Pajares

See Also

`perc.thr`, `NINA.thr`

Examples

```
#generating distance matrix:
dis<-matrix(nrow=12,c(0.0000,0.5000,0.1875,0.5000,0.6250,0.5000,0.2500,0.6250,
0.3750,0.3750,0.3750,0.3750,0.5000,0.0000,0.7500,0.0000,0.6250,0.0000,0.8750,
0.6250,0.3750,0.3750,0.3750,0.3750,0.1875,0.7500,0.0000,0.7500,0.8750,0.7500,
0.2500,0.8750,0.6250,0.6250,0.6250,0.6250,0.5000,0.0000,0.7500,0.0000,0.6250,
0.0000,0.8750,0.6250,0.3750,0.3750,0.3750,0.3750,0.6250,0.6250,0.8750,0.6250,
0.0000,0.6250,0.5000,0.0000,0.2500,0.2500,0.2500,0.2500,0.5000,0.0000,0.7500,
0.0000,0.6250,0.0000,0.8750,0.6250,0.3750,0.3750,0.3750,0.3750,0.2500,0.8750,
0.2500,0.8750,0.5000,0.8750,0.0000,0.5000,0.5000,0.5000,0.5000,0.5000,0.6250,
0.6250,0.8750,0.6250,0.0000,0.6250,0.5000,0.0000,0.2500,0.2500,0.2500,0.2500,
0.3750,0.3750,0.6250,0.3750,0.2500,0.3750,0.5000,0.2500,0.0000,0.0000,0.0000,
0.0000,0.3750,0.3750,0.6250,0.3750,0.2500,0.3750,0.5000,0.2500,0.0000,0.0000,
0.0000,0.0000,0.3750,0.3750,0.6250,0.3750,0.2500,0.3750,0.5000,0.2500,0.0000,
0.0000,0.0000,0.0000,0.3750,0.3750,0.6250,0.3750,0.2500,0.3750,0.5000,0.2500,
0.0000,0.0000,0.0000,0.0000))
row.names(dis)<-c("Population1_sequence1","Population1_sequence2",
"Population1_sequence3","Population1_sequence4","Population2_sequence1",
"Population2_sequence2","Population2_sequence3","Population2_sequence4",
"Population3_sequence1","Population3_sequence2","Population3_sequence3",
"Population3_sequence4")
colnames(dis)<-row.names(dis)

#Representing distances equal or lower than 37% of the maximum distance:
# single.network(dis=dis,threshold=0.37,label=paste(substr(row.names(dis),11,11),
# substr(row.names(dis),21,21),sep="-"))
```

---

single.network.module *Get modules and network given a threshold*

Description

Gets details on modules and connections in the network connecting nodes showing distances equal or lower than the defined threshold value.

Usage

```
single.network.module(dis,threshold=NA,refer2max=TRUE,out="module",
save.file=FALSE,modFileName="Modules_summary.txt")
```

Arguments

`dis` the distance matrix to be represented

`threshold` a numeric between 0 and 1, is the value of the maximum distance to be considered as a link. This value is referred to the maximum distance in the input matrix (e.g., a value of 0.32 will represent a link between nodes showing distances equal or lower than 32% of the maximum distance found in the distance matrix).

`refer2max` a logic, "TRUE" to refer the threshold value to the maximum distance in the input matrix (e.g., a value of 0.32 will represent a link between nodes showing distances equal or lower than 32% of the maximum distance found in the distance matrix). "FALSE" to refer the threshold to a specific value (e.g., a value of 0.32 will represent a link between nodes showing distances equal or lower than 0.32, regardless the maximum distance found in the distance matrix).

`out` a string, the type of output, "module" to get a matrix with two columns giving each sequence name and the module it belongs to, and "network" to get a square matrix representing connection (1) or lack of connection (0) between sequences in the network.

`save.file` a logic, "TRUE" to save the summary of network modules, attributing every individual to a module.

`modFileName` (if modules=TRUE) the name of a generated file containing a summary of module results

Author(s)

A. J. Muñoz-Pajares

See Also

`perc.thr`, `NINA.thr`

Examples

```
#generating distance matrix:
dis<-matrix(nrow=12,c(0.0000,0.5000,0.1875,0.5000,0.6250,0.5000,0.2500,0.6250,
0.3750,0.3750,0.3750,0.3750,0.5000,0.0000,0.7500,0.0000,0.6250,0.0000,0.8750,
0.6250,0.3750,0.3750,0.3750,0.3750,0.1875,0.7500,0.0000,0.7500,0.8750,0.7500,
0.2500,0.8750,0.6250,0.6250,0.6250,0.6250,0.5000,0.0000,0.7500,0.0000,0.6250,
0.0000,0.8750,0.6250,0.3750,0.3750,0.3750,0.3750,0.6250,0.6250,0.8750,0.6250,
0.0000,0.6250,0.5000,0.0000,0.2500,0.2500,0.2500,0.2500,0.5000,0.0000,0.7500,
0.0000,0.6250,0.0000,0.8750,0.6250,0.3750,0.3750,0.3750,0.3750,0.2500,0.8750,
0.2500,0.8750,0.5000,0.8750,0.0000,0.5000,0.5000,0.5000,0.5000,0.5000,0.6250,
0.6250,0.8750,0.6250,0.0000,0.6250,0.5000,0.0000,0.2500,0.2500,0.2500,0.2500,
0.3750,0.3750,0.6250,0.3750,0.2500,0.3750,0.5000,0.2500,0.0000,0.0000,0.0000,
0.0000,0.3750,0.3750,0.6250,0.3750,0.2500,0.3750,0.5000,0.2500,0.0000,0.0000,
0.0000,0.0000,0.3750,0.3750,0.6250,0.3750,0.2500,0.3750,0.5000,0.2500,0.0000,
0.0000,0.0000,0.0000,0.3750,0.3750,0.6250,0.3750,0.2500,0.3750,0.5000,0.2500,
0.0000,0.0000,0.0000,0.0000))
row.names(dis)<-c("Population1_sequence1","Population1_sequence2",
"Population1_sequence3","Population1_sequence4","Population2_sequence1",
"Population2_sequence2","Population2_sequence3","Population2_sequence4",
"Population3_sequence1","Population3_sequence2","Population3_sequence3",
"Population3_sequence4")
colnames(dis)<-row.names(dis)

# #Representing distances equal or lower than 37% of the maximum distance:
# single.network.module(dis=dis,threshold=0.37)
# single.network.module(dis=dis,threshold=0.37,out="network")
```

```
#
# # Compare these outputs with:
# single.network(dis=dis,threshold=0.37,label=paste(substr(row.names(dis),11,11),
# substr(row.names(dis),21,21),sep="-"))
```

spatial.plot *spatial plot of populations*

Description

This function estimates the phylogeographic relationships among populations, displaying nodes according to geographic coordinates on maps.

Usage

```
spatial.plot(dis=NULL, align=NA, X=NULL, Y=NULL, indel.method="MCIC",
substitution.model="raw", pairwise.deletion=TRUE, alpha="info",
combination.method="Corrected", na.rm.row.col=FALSE, addExtremes=FALSE,
NameIniPopulations=NA, NameEndPopulations=NA, NameIniHaplotypes=NA,
NameEndHaplotypes=NA, HaplosNames=NA, save.distance=FALSE,
save.distance.name="DistanceMatrix_threshold.txt",
network.method="percolation", range=seq(0,1,0.01), modules=FALSE,
moduleCol=NA, modFileName="Modules_summary.txt", bgcol="white",
label.col="black", label=NA, label.sub.str=NA, label.pos= "b",
cex.label=1,cex.vertex=1,vertex.size="equal", plot.edges=TRUE,
lwd.edge=1,to.ggmap=FALSE, plot.ggmap=FALSE, zoom.ggmap=6,
maptype.ggmap="satellite", label.size.ggmap=3)
```

Arguments

`dis` a matrix; the distance matrix to be analysed. Alternatively, you can define an alignment using 'align' option.

`align` a 'DNAbin' object; the alignment to be analysed. See "read.dna" in the ape package for details about reading alignments. Alternatively, you can define a distance matrix using the 'dis' option.

`X` a vector; longitude for each population

`Y` a vector; latitude for each population

`indel.method` a sting; the method to define indel events in your alignments. The available methods are:

-"MCIC": (Default) Estimates indel events following the rationale of the Modified Complex Indel Coding (Muller, 2006).

-"SIC": Estimates indel events following the rationale of Simmons and Ochoterena (2000).

-"FIFTH": Estimates indel events following the rationale of the fifth state: each gap within the alignment is treated as an independent mutation event.

-"BARRIEL": Estimates indel events following the rationale of Barriel (1994): singleton gaps are not taken into account.

`substitution.model`
a string; the substitution evolutionary model to estimate the distance matrix. By default is set to "raw" and estimates the pairwise proportion of variant sites. See the evolutionary models available using ?dist.dna from the ape package.

`pairwise.deletion`
a logical; if TRUE (default) substitutions found in regions being a gap in other sequences will account for the distance matrix. If FALSE, sites being a gap in at least one sequence will be removed before distance estimation.

`network.method` a string; the method to build the network. The available methods are:

-"percolation": computes a network using the percolation network method following Rozenfeld et al. (2008). See ?perc.thr for details

-"NINA": computes a network using the No Isolation Nodes Allowed method. See ?NINA.thr for details.

-"zero": computes a network connecting all nodes showing distances equal to zero. See ?NINA.thr for details.

`range` a numeric vector between 0 and 1, is the range of thresholds (referred to the maximum distance in the input matrix) to be screened (by default, 101 values from 0 to 1). This option is used for "percolation" and "NINA" network methods and ignored for "zero" method.

`addExtremes` a logical; if TRUE, additional nucleotide sites are included in both extremes of the alignment. This will allow estimating distances for alignments showing gaps in terminal positions. This option is used for "SIC", "FIFTH" and "BARRIEL" indel methods and ignored for "MCIC" method.

`alpha` a numeric between 0 and 1, is the weight given to the indel genetic distance matrix in the combination. By definition, the weight of the substitution genetic matrix is the complementary value (i.e., 1-alpha). The value "info" will use the proportion of informative substitutions per informative indel event as weight. It is also possible to define multiple weights to estimate different combinations.

`combination.method`
a string defining whether each distance matrix must be divided by its maximum value before the combination ("Corrected") or not ("Uncorrected"). Consequently, if the "Corrected" method is chosen, both matrices will range between 0 and 1 before being combined.

`na.rm.row.col` a logical; if TRUE, distance matrix missing values are removed.

`modules` a logical, If TRUE, nodes belonging to different modules are represented as different colours (defined by 'moduleCol').

`moduleCol` (if modules=TRUE) a vector of strings defining the colour of nodes belonging to different modules in the network. If 'NA' (or there are less colours than haplotyes), colours are automatically selected

`modFileName` (if modules=TRUE) a string, the name of the file to be generated containing a summary of module results (sequence name, module, and colour in network)

`save.distance` a logical; if TRUE, the distance matrix used to build the network will be saved as a file.

`save.distance.name`
a string; if save.distance=TRUE, it defines the name of the file to be saved.

bgcol a vector of strings; the colour of the background for each node in the network. Can be equal for all nodes (if only one colour is defined), customized (if several colours are defined), or can represent different modules (see "modules" option). If set to 'NA' (default) or if less colours than haplotyes are defined, colours are automatically selected.

label.col a vector of strings; the colour of labels for each node in the network. Can be equal for all nodes (if only one colour is defined) or customized (if several colours are defined).

label a vector of strings; labels for each node. By default are the sequence names. (See "substr" function in base package to automatically reduce name lengths)

label.sub.str a vector of two numerics; if node labels are a sub-string of sequence names, these two numbers represent the initial and final character of the string to be represented. See Example for details.

label.pos a sting; position for vertex labels regarding vertex position (do not affect the ggmap output). Possible values are: "b" or "below" (default), "a" or "above"; "l" or "left"; "r" or "right" and "c" or "centre"

lwd.edge a numeric; the width of the edge linking nodes (1.5 by default).

cex.label a numeric; the size of the node labels.

cex.vertex a numeric; the size of the nodes.

NameIniPopulations

a numeric; Position of the initial character of population names within sequence names. If not provided, it is set to 1. It is used only if NameEndPopulations is also defined.

NameEndPopulations

a numeric; Position of the last character of population names within sequence names. If not provided, it is set to the first "_" character in the sequences name. It is used only if NameIniPopulations is also defined.

NameIniHaplotypes

a numeric; Position of the initial character of haplotype names within sequence names. If not provided, haplotype names are given and the value is set accordingly. It is used only if NameEndHaplotypes is also defined.

NameEndHaplotypes

a numeric; Position of the last character of haplotype names within sequence names. If not provided, haplotype names are given and the value is set accordingly. It is used only if NameIniHaplotypes is also defined.

plot.edges a logical; must the edges connecting nodes be potted?

HaplosNames a sting; the name of the haplotypes (if different from default: H1...Hn)

to.ggmap a logical; if TRUE, the algorithm generates a list with information required to represent the resulting network using ggmap (see details).

plot.ggmap a logical; if TRUE, populations (and edges producing a network if 'plot.edges' is set to TRUE) are represented within a map automatically downloaded according the population coordinates.

zoom.ggmap a numeric; sets the zoom of the map (higher values mean deeper zoom)

maptype.ggmap a string; types of maps implemented by 'ggplot' are: "terrain", "satellite", "roadmap", "hybrid", "toner", and "watercolor")

`label.size.ggmap`
a numeric; controls the labe size in the ggplot

`vertex.size` a string to define the ratio of vertices representing populations. Possible values are: "equal" (default) to give the same size to all vertices; or "area" to make the vertex area proportional to the population sample size.

Details

Despite the large list of options, the only mandatory options for this function are the geographic coordinates ('X' and 'Y' options) of the studied populations and either the alignment or the distance matrix ('align' or 'dis', respectively). The remaining options can be classified into five groups:

1- options defining the computation of both indel and substitution distances (indel.method, substitution.model, pairwise.deletion).

2- options defining the combination of these two distance matrices (alpha, combination.method, na.rm.row.col, addExtremes, NameIniPopulations, NameEndPopulations, NameIniHaplotypes, NameEndHaplotypes, HaplosNames, save.distance, save.distance.name).

3- options defining the computation of the network (network.method, range).

4- options customizing the resulting network (modules, moduleCol, modFileName, bgcol, label.col, label, label.sub.str, cex.label, cex.vertex, vertex.size, plot.edges, lwd.edge).

5- options dealing with map representation (to.ggmap, plot.ggmap, zoom.ggmap, maptype.ggmap, label.size.ggmap).

Although the 'indel.method' option affects both the distance estimation and the number of mutations represented in the network, the 'substitution.model' and 'pairwise.deletion' options only affect the distance matrix computation.

This function provides limited options for representing of the resulting population network within a map using the 'ggmap' package. To take advantage of the additional options implemented in ggmap, the 'to.ggmap' option generates a list with the following information:

1- location: centroid of the population coordinates (required to center the map)

2- colours: the colour to represent each population (useful, for example to represent modules)

3- coordinates: the geographic coordinates of the studied populations

4- network: the resulting population network, represented as a 1/0 matrix

5- links: a two column matrix representing the edges within the resulting network. Each row provides information on the two elements that are connected by a link.

Author(s)

A. J. Muñoz-Pajares

References

Barriel, V., 1994. Molecular phylogenies and how to code insertion/ deletion events. Life Sci. 317, 693-701, cited and described by Simmons, M.P., Müller, K. & Norton, A.P. (2007) The relative performance of indel-coding methods in simulations. Molecular Phylogenetics and Evolution, 44, 724–740.

Muller K. (2006). Incorporating information from length-mutational events into phylogenetic analysis. Molecular Phylogenetics and Evolution, 38, 667-676.

Paradis, E., Claude, J. & Strimmer, K. (2004). APE: analyses of phylogenetics and evolution in R language. Bioinformatics, 20, 289-290.

Rozenfeld AF, Arnaud-Haond S, Hernandez-Garcia E, Eguiluz VM, Serrao EA, Duarte CM. (2008). Network analysis identifies weak and strong links in a metapopulation system. Proceedings of the National Academy of Sciences,105, 18824-18829.

Simmons, M.P. & Ochoterena, H. (2000). Gaps as Characters in Sequence-Based Phylogenetic Analyses. Systematic Biology, 49, 369-381.

Examples

```
# library(ggplot2)
# data(ex_Coords)
# data(ex_alignment1) # this will read a fasta file with the name 'alignExample'

# A simple plot of the population network using geographic coordinates:
# spatial.plot (align=alignExample,X=ex_Coords[,2],Y=ex_Coords[,3])

# Changing vertex names and location:
# spatial.plot (align=alignExample,X=ex_Coords[,2],Y=ex_Coords[,3],
# cex.vertex=2,label=c(1:8),label.pos=”c”,modules=TRUE)

# Plotting network on a map:
# Uncomment the lines below. It would take more than 5 seconds to run
# spatial.plot (align=alignExample,X=ex_Coords[,2],Y=ex_Coords[,3],
# cex.vertex=2,label=c(1:8),modules=TRUE, plot.ggmap=TRUE)

# Displaying only population coordinates (sampling desing).
# Uncomment the lines below. It would take more than 5 seconds to run
# spatial.plot (align=alignExample,X=ex_Coords[,2],Y=ex_Coords[,3],
# cex.vertex=2,label=c(1:8), plot.ggmap=TRUE,plot.edges=FALSE,
# bgcol=c(”red”,”orange”,”green4”,”green1”,”yellow”,”brown”,”blue”,”purple”))
```

zero.thr *Zero distance networks*

Description

Given a distance matrix, this function computes a network connecting nodes showing distances equal to zero.

Usage

```
zero.thr(dis,ptPDF=TRUE,ptPDFname=”zero_Network.pdf”,cex.label=1,cex.vertex=1,
bgcol=”white”,label.col=”black”,label=colnames(dis),modules=FALSE,moduleCol=NA,
modFileName=”Modules_summary.txt”,ncs=4,na.rm.row.col=FALSE)
```

Arguments

dis the input distance matrix

ptPDF a logical, must the resulting network be saved as a pdf file?

ptPDFname if ptPDF=TRUE, the name of the pdf file containing the resulting network to be saved ("zero_Network.pdf", by default)

cex.label a numeric; the size of the node labels.

cex.vertex a numeric; the size of the nodes.

bgcol the background colour for each node in the network. Can be equal for all nodes (if only one colour is defined), customized (if several colours are defined), or can represent different modules (see "modules" option).

label.col vector of strings defining the colour of labels for each node in the network. Can be equal for all nodes (if only one colour is defined) or customized (if several colours are defined).

label vector of strings, labels for each node. By default are the column names of the distance matrix (dis). (See the 'substr' function in base package to automatically reduce name lengths).

modules a logical, must nodes belonging to different modules be represented with different colours? If TRUE, a text file containing information on modules for each node is also produced.

moduleCol (if modules=TRUE) a vector of strings, defining the colour of nodes belonging to different modules in the network. If 'NA' or less colours than modules are defined, colours are automatically defined.

modFileName (if modules=TRUE) the name of the text file containing a summary of module results

ncs a numeric; number of decimal places to display threshold in plot title.

na.rm.row.col a logical; if TRUE, missing values are removed before the computation proceeds.

Details

In some circumstances you may get distance matrices showing off-diagonal zeros. In such cases you may consider that the existence of these off-diagonal zeros suggests that some of the groups you defined (e.g., populations) are not genetically different. Thus, you must re-define groups to get a matrix composed only by different groups using the 'mergeNodes' function and estimate a percolation network using the 'perc.thr' function. On the other hand, you may consider that, despite the off- diagonal zeros, the groups you defined are actually different. In that case you may not be able to estimate a percolation threshold, but you can represent the original distance matrix using the 'NINA.thr' or the 'zero.thr' functions.

'mergeNodes' select all rows (and columns) showing a distance equal to zero and generates a new row (and column). The distance between the new merged and the remaining rows (or columns) in the matrix is estimated as the arithmetic mean of the selected elements. The biological interpretation of the new matrix could be hard if the original matrix shows a large number of off-diagonal zeros.

'perc.thr' estimates a threshold to represent a distance matrix as a network. To estimate this threshold, the algorithm represents as a link all distances lower than a range of thresholds (by default, select 101 values from 0 to 1), defined as the percentage of the maximum distance in the input matrix. For each threshold a network is built and the number of clusters (that is, the number of isolated groups of nodes) in the network is also estimated. Finally, the algorithm selects the lower threshold connecting a higher number of nodes. Note that the resulting network may show isolated nodes if it is necessary to represent a large number of links to connect a low number of nodes.

'NINA.thr' is identical to 'perc.thr', but, in the last step, the algorithm selects the lower threshold connecting all nodes in a single cluster. The information provided by this function may be limited if the original distance matrix shows high variation.

'zero.thr' represents as a link only distances equal to zero. The information provided by this function may be limited if the original matrix shows few off-diagonal zeros.

Value

A network connecting nodes showing a distance equal to zero.

Author(s)

A. J. Muñoz-Pajares

See Also

NINA.thr, perc.thr, mergeNodes

Examples

```
#EXAMPLE 1: FEW OFF-DIAGONAL ZEROS
#Generating a distance matrix:
Dis1<-matrix(c(
0.00,0.77,0.28,0.94,0.17,0.14,0.08,0.49,0.64,0.01,
0.77,0.00,0.12,0.78,0.97,0.02,0.58,0.09,0.36,0.33,
0.28,0.12,0.00,0.70,0.73,0.06,0.50,0.79,0.80,0.94,
0.94,0.78,0.70,0.00,0.00,0.78,0.04,0.42,0.25,0.85,
0.17,0.97,0.73,0.00,0.00,0.30,0.55,0.12,0.68,0.99,
0.14,0.02,0.06,0.78,0.30,0.00,0.71,1.00,0.64,0.88,
0.08,0.58,0.50,0.04,0.55,0.71,0.00,0.35,0.84,0.76,
0.49,0.09,0.79,0.42,0.12,1.00,0.35,0.00,0.56,0.81,
0.64,0.36,0.80,0.25,0.68,0.64,0.84,0.56,0.00,0.62,
0.01,0.33,0.94,0.85,0.99,0.88,0.76,0.81,0.62,0.00),ncol=10)
colnames(Dis1)<-c(paste("Pop",c(1:10),sep=""))
row.names(Dis1)<-colnames(Dis1)

# No percolation threshold can be found.
#perc.thr(Dis1)

#Check Dis1 and merge populations showing distances equal to zero:
Dis1
Dis1_Merged<-mergeNodes(dis=Dis1)
#Check the merged matrix. A new "population" has been defined merging populations 4 and 5.
#Distances between the merged and the remaining populations are estimated as the arithmetic mean.
Dis1_Merged
# It is now possible to estimate a percolation threshold
```

```
perc.thr(dis=Dis1_Merged,ptPDF=FALSE, estimPDF=FALSE, estimOutfile=FALSE)

# EXAMPLE 2: TOO MANY OFF-DIAGONAL ZEROS
#Generating a distance matrix:
Dis2<-matrix(c(
0.00,0.77,0.28,0.00,0.17,0.14,0.00,0.49,0.64,0.01,
0.77,0.00,0.12,0.00,0.97,0.02,0.00,0.09,0.36,0.33,
0.28,0.12,0.00,0.70,0.73,0.06,0.50,0.79,0.00,0.94,
0.00,0.00,0.70,0.00,0.00,0.78,0.04,0.00,0.00,0.00,
0.17,0.97,0.73,0.00,0.00,0.30,0.55,0.12,0.00,0.00,
0.14,0.02,0.06,0.78,0.30,0.00,0.71,1.00,0.64,0.00,
0.00,0.00,0.50,0.04,0.55,0.71,0.00,0.35,0.84,0.00,
0.49,0.09,0.79,0.00,0.12,1.00,0.35,0.00,0.56,0.81,
0.64,0.36,0.00,0.00,0.00,0.64,0.84,0.56,0.00,0.62,
0.01,0.33,0.94,0.00,0.00,0.00,0.00,0.81,0.62,0.00),ncol=10)
colnames(Dis2)<-c(paste(”Pop”,c(1:10),sep=””))
row.names(Dis2)<-colnames(Dis2)

# # No percolation threshold can be found
# #perc.thr(Dis2)
#
# #Check Dis2 and merge populations showing distances equal to zero:
# Dis2
# Dis2_Merged<-mergeNodes(dis=Dis2)
#
# #Check the merged matrix. Many new ”populations” have been defined and both the new
# #matrix and the resulting network are difficult to interpret:
# Dis2_Merged
# perc.thr(dis=Dis2_Merged,ptPDF=FALSE, estimPDF=FALSE, estimOutfile=FALSE)
#
# #Instead of percolation network, representing zeros as the lowest values may be informative:
# zero.thr(dis=Dis2,ptPDF=FALSE)
# # Adjusting sizes and showing modules:
# zero.thr(dis=Dis2,ptPDF=FALSE,cex.label=0.8,cex.vertex=1.2,modules=TRUE)
#
# #In the previous example, the 'zero.thr' method is unuseful:
# zero.thr(dis=Dis1,ptPDF=FALSE)
#
# #In both cases, the 'No Isolation Nodes Allowed' method yields an informative matrix:
# NINA.thr(dis=Dis1)
# NINA.thr(dis=Dis2)
```

Index