

Package ‘sidier’

February 19, 2013

Type Package

Title What the package does (short line)

Version 1.0

Date 2013-02-16

Author A.J. Munoz-Pajares

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

Depends R (>= 2.10.1), ape, igraph, network

Description SIDIER provides functions for reading and writing fasta sequences, finding unique haplotypes, estimating genetic distances based on gap positions and lengths, combining distance matrices and estimating and plotting percolation networks.

License GPL (>=2)

R topics documented:

sidier-package	1
FindHaplo	2
GetHaplo	3
HapPerPop	5
MCIC	6
nt.gap.comb	7
perc.thr	9
pop.dist	11

sidier-package	<i>SIDIER: Substitution and Indel Distances to Infer Evolutionary Relationships</i>
----------------	---

Description

sidier is a library and R package for evolutionary reconstruction based on substitutions and insertion-deletion (indels) analyses in a distance-based framework.

Details

Package: sidier
 Type: Package
 Version: 1.0
 Date: 2013-02-16
 License: GPL-2

Author(s)

A. Jesus Munoz Pajares\ Maintainer: A. Jesus Munoz Pajares <ajesusmp@ugr.es>

References

- Muller K. (2006). Incorporating information from length-mutational events into phylogenetic analysis. *Molecular Phylogenetics and Evolution*, 38, 667-676.
- Munoz-Pajares, AJ. SIDIER: Substitution and Indel Distances to Infer Evolutionary Relationships.
- Munoz-Pajares, A.J., Abdelaziz, M., Gomez, J.M., Perfectti, F. Combining indels and substitutions information for the reconstruction of evolutionary haplotype relationships.
- Munoz-Pajares, A.J., Abdelaziz, M., Herrador, M.B., Gomez, J.M., Perfectti, F. Phylogeography and colonization pathways of the *Erysimum nevadense* species complex based on a plastidial indel-rich region distance analysis.
- 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.

FindHaplo

*Find equal haplotypes**Find equal haplotypes*

Description

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

Usage

```
FindHaplo(readfile = T, input = NA, align = NA, saveFile = T, outname = "FindHaplo.txt")
```

Arguments

readfile	a logical; if TRUE (default), the input alignment is provided as a fasta format in a text file. If FALSE, the alignment is provided as an R object.
input	the name of the fasta file to be analysed.
align	the name of the alignment to be analysed (if "readfile" is set to FALSE,). See "read.dna" in ape package for details about reading alignments.
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).

Value

A matrix showing the assigned haplotype name to each sequence in the alignment.

Author(s)

A.J. Munoz-Pajares

See Also

HapPerPop

Examples

```
cat(">Population1_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCATC-----",
">Population2_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCATCAATATTATATCGGCGATC",
">Population3_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----GTA-----TCGATGGCGCGGCATC-----",
">Population4_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----",
">Population5_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTA-----TATCGGCGATC")
file = "ex1.fas", sep = "\n")
example1 <- read.dna("ex1.fas", format = "fasta")
# Reading the alignment from an object:
FindHaplo(readfile=FALSE,align=example1)
# Reading the alignment directly from file:
FindHaplo(input="ex1.fas")
```

GetHaplo	<i>Get sequences of unique haplotypes</i>
----------	---

Description

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

Usage

```
GetHaplo(readfile = T, input = NA, align = NA, saveFile = T, outname = "Haplotyp
```

Arguments

readfile	a logical; if TRUE (default) input alignment is provided as a fasta format in a text file. If FALSE, the alignment is provided as an R object.
input	the name of the fasta file to be analysed.
align	the name of the alignment to be analysed (if "readfile" is set to FALSE,). See "read.dna" in ape package for details about reading alignments.
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" assign 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.

Details

If two equal sequences are not identically aligned, they will be considered as different haplotypes. To avoid misleading results in uncertain alignments it is recommended to use as input the original unaligned sequences, including gaps after the last nucleotide of short sequences to make all sequence lengths equal.

Value

A file containing unique sequences from the input file.

Author(s)

A.J. Munoz-Pajares

Examples

```
cat(">Population1_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population1_sequence2",
"TTATAGCTGTCGGGCTA-----GTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population1_sequence3",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population1_sequence4",
"TTATAGCTGTCGGGCTA-----GTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population2_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGGATC",
">Population2_sequence2",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGGATC",
">Population2_sequence3",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGGATC",
">Population2_sequence4",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGGATC",
">Population3_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----",
">Population3_sequence2",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----",
">Population3_sequence3",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----",
">Population3_sequence4",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----",
      file = "ex2.fas", sep = "\n")
example2 <- read.dna("ex2.fas", format = "fasta")
alin<-read.dna(file="ex2.fas",format="fasta")

# Reading the alignment from an object and saving haplotypes names as sequential numbers
```

```
GetHaplo(readfile=FALSE, align=alin, outname="Haplotypes_sequentialNames.txt", seqsNames="In

# Reading the alignment directly from file and saving using sequence input names:
GetHaplo(input="ex2.fas")
```

HapPerPop

Returns the number of haplotypes per population.

Description

Given a two column matrix, this function returns the number of haplotypes per population. 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 haplotype. The desired matrix can be obtained using "FindHaplo".

Usage

```
HapPerPop(readfile = T, sep = " ", header = F, inputFile = NA, input = NA, saveF
```

Arguments

readfile	a logical; if TRUE (default) the input matrix is provided in a text file. If FALSE, the matrix is provided as an R object.
sep	the character separating columns in the input matrix (space, by default).
header	a logical value indicating whether the input matrix contains the names of the variables as its first line. (Default=FALSE).
inputFile	(if readfile=TRUE) the name of the file containing the input matrix.(if readfile=TRUE) the name of the file containing the input matrix.
input	a logical; if TRUE (default), the two ouput matrices computed are saved as two different text files.
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

Two output matrices are estimated, one giving the abundance of each haplotype per population (named weighted matrix) and the other representing presence/absence of each haplotype per population by 1/0 (named interaction matrix).

Value

A list containing two matrices.

comp1	The first matrix contains the weighted matrix, that is, the number of haplotypes (columns) found per population (rows).
comp2	The second is the interaction matrix, containing information about the presence or absence of each haplotype (columns) per population (rows).

Author(s)

A.J. Munoz-Pajares

See Also

FindHaplo

Examples

```

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")
example2_2 <- read.table("3_FindHaplo_Example2_modified.txt", header=TRUE)
FH<-read.table("3_FindHaplo_Example2_modified.txt", header=TRUE)

# Reading the alignment from an object and saving the two computed distance matrices:
HapPerPop(readfile=FALSE, input=FH, header=TRUE, saveFile=FALSE)

# Reading the alignment directly from file, displaying only the weighted matrix:
HapPerPop(readfile=TRUE, inputFile="3_FindHaplo_Example2_modified.txt", header=TRUE, saveFile=FALSE)

```

MCIC

*Modified Complex Indel Coding as distance matrix***Description**

This function computes the insertion-deletion (indel) distance matrix following the rationale of the Modified Complex Indel Coding (Muller, 2006) to estimate transition matrices, as described in Munoz-Pajares.

Usage

```
MCIC(readfile = T, input = NA, align = NA, saveFile = T, outname = paste(input,
```

Arguments

<code>readfile</code>	a logical; if TRUE (default) input alignment is provided as a fasta format in a text file. If FALSE, the alignment is provided as an R object.
<code>input</code>	the name of the fasta file to be analysed.
<code>align</code>	the name of the alignment to be analysed (if "readfile" is set to FALSE,). See "read.dna" in ape package for details about reading alignments.

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.

Value

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

Author(s)

A.J. Munoz-Pajares

References

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

Munoz-Pajares, AJ. SIDIER: Substitution and Indel Distances to Infer Evolutionary Relationships.

Examples

```
cat(">Population1_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population1_sequence2",
"TTATAGCTGTCGGGCTA-----GTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population1_sequence3",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population1_sequence4",
"TTATAGCTGTCGGGCTA-----GTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population2_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC",
">Population2_sequence2",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC",
">Population2_sequence3",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC",
">Population2_sequence4",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC",
">Population3_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----",
">Population3_sequence2",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----",
">Population3_sequence3",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----",
">Population3_sequence4",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----",
      file = "ex2.fas", sep = "\n")

# Reading the alignment directly from file and saving no output file:
MCIC (input="ex2.fas", saveFile = FALSE)
```

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, range = seq(0, 1, 0.1), method = "Corrected")
```

Arguments

DISTnuc	a matrix containing substitution genetic distances. See "dist.dna" in "ape" package.
DISTgap	a matrix containing indel genetic distances. See MCMC function in this package.
range	a numeric between 0 and 1, is the weights given to the indel genetic distance matrix in the combination. By definition, the weight of the substitution genetic matrix is the complementary value.
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.

Value

A list containing the estimated combination of substitution and indel distance matrices.

Author(s)

A.J. Munoz-Pajares

See Also

MCIC

Examples

```
cat(">Population1_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population1_sequence2",
"TTATAGCTGTCGGGCTA-----GTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population1_sequence3",
"GGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population1_sequence4",
"TTATAGCTGTCGGGCTA-----GTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----",
">Population2_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC-----")
```



```

">Population2_sequence2",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population2_sequence3",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population2_sequence4",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population3_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----
">Population3_sequence2",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----
">Population3_sequence3",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----
">Population3_sequence4",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----
    file = "ex2.fas", sep = "\n")
    example2 <- read.dna("ex2.fas", format = "fasta")

# 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:
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, range=seq(0,1,0.1), method="Corrected", saveFile=FALSE, DIST
# Obtaining the arithmetic mean of both matrices using both the corrected and the uncorrected
nt.gap.comb(DISTgap=distGap, range=0.5, method="Both", saveFile=FALSE, DISTnuc=DISTnt)

```

perc.thr

Percolation threshold network

Description

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

Usage

```
perc.thr(dis, threshold = seq(0, 1, 0.01), ptPDF = TRUE, ptPDFname = "Percolated
```

Arguments

dis	the distance matrix to be represented
threshold	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? If estimPDF=TRUE (default) the value of <s> for each threshold is also saved

<code>estimPDFname</code>	if <code>estimPDF=TRUE</code> (default), defines the name of the pdf file containing the percolation threshold estimation ("PercThr Estimation.pdf" by default).
<code>estimOutfile</code>	a logical, must the matrix containing percolation threshold estimation variables be saved as a pdf file?
<code>estimOutName</code>	if <code>estimOutfile=TRUE</code> (default), contains the name of the text file containing the percolation threshold estimation ("PercThr Estimation.txt" by default).
<code>appendOutfile</code>	a logical, if <code>estimOutfile=TRUE</code> , it defines whether results must be appended to an existing file with the same name (TRUE) or the existing file must be replaced (FALSE).
<code>plotALL</code>	a logical, must all the networks calculated during the percolation threshold estimation be saved as different pdf files? (FALSE, by default). If TRUE, for each value defined in threshold, one file is generated.
<code>bgcol</code>	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).
<code>label.col</code>	string, 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),
<code>label</code>	string, labels for each node. By default are the column names of the distance matrix (dis). (See <code>substr</code> function in base package to automatically reduce name lengths).
<code>modules</code>	a logical, must nodes belonging to different modules be represented as different colours?

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., `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.

Author(s)

A.J. Munoz-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.

Munoz-Pajares, AJ. SIDIER: Substitution and Indel Distances to Infer Evolutionary Relationships.

Examples

```
cat(">Population1_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCATC-----",
">Population1_sequence2",
"TTATAGCTGTCGGGCTA-----GTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCATC-----")
```

```

">Population1_sequence3",
"GGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
">Population1_sequence4",
"TTATAGCTGTCGGGCTA-----GTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
">Population2_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population2_sequence2",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population2_sequence3",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population2_sequence4",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population3_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----
">Population3_sequence2",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----
">Population3_sequence3",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----
">Population3_sequence4",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATC-----
    file = "ex2.fas", sep = "\n")
    example2 <- read.dna("ex2.fas", format = "fasta")

# 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:
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, range=0.5, method="Corrected", saveFile=FA
# Estimating the percolation threshold of the combined distance, modifying labels:
#perc.thr(dis=as.data.frame(CombinedDistance$Corrected), label=paste(substr(row.names(as.d
# Estimating the percolation threshold of the combined distance, modifying labels:
#perc.thr(dis=as.data.frame(CombinedDistance$Corrected), label=paste(substr(row.names(as.d
# The same network showing different modules as different colours (randomly selected):
#perc.thr(dis=as.data.frame(CombinedDistance$Corrected), label=paste(substr(row.names(as.d

```

pop.dist

Distances among populations

Description

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

Usage

```
pop.dist(DistFile = T, inputDist = NA, distances = NA, HaploFile = T, inputHaplo
```

Arguments

DistFile	a logical; if TRUE (default) input distance matrix among haplotypes is provided as a matrix in a text file. If FALSE, the matrix must be provided as an R object.
inputDist	the name of the file containing the distance matrix among haplotypes.
distances	the name of the distance matrix among haplotypes to be analysed (if "DistFile" is set to FALSE,).
HaploFile	a logical; if TRUE (default) the input matrix containing the number of haplotypes found per population is provided as a matrix in a text file. If FALSE, the matrix must be provided as an R object. See HapPerPop for details on how to estimate such matrix.
inputHaplo	the name of the file containing the matrix with the number of haplotypes found per population.
Haplos	the name of the matrix containing the number of haplotypes found per population (if "DistFile" is set to FALSE,).
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 matrix names used (input-Dist and HaploFile)
saveFile	a logical; if TRUE (default), function output is saved as a text file.
NameIni	a numeric indicating the position of the initial character of population name within the individual name in the distance matrix.
NameEnd	a numeric indicating the position of the last character of population name within the individual name in the distance matrix.

Value

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

Author(s)

A.J. Munoz-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)
```

```

cat(">Population1_sequence1",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
">Population1_sequence2",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
">Population1_sequence3",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
">Population1_sequence4",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
">Population2_sequence1",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population2_sequence2",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population2_sequence3",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population2_sequence4",
"TTATAGCTGTCGGGCTAGTAGCTGTATCAGTC-----TCGATGGCGCGGCGCATCAATATTATATCGGCGATC
">Population3_sequence1",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
">Population3_sequence2",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
">Population3_sequence3",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
">Population3_sequence4",
"GGGGAGCTGTCGGGCTAGTAGCTGTATCAGTCGTACGTAGTAGTCGTGTCGATCGATGGCGCGGCGCATC-----
    file = "5_Example4.fas", sep = "\n")
    example4 <- read.table("5_Example4.fas",header=TRUE)

# Reading files. Distance matrix must contain haplotype names. Abundance matrix must contain

pop.dist (DistFile=TRUE, inputDist="4_Example3_IndelDistanceMatrixMullerMod.txt", HaploFile="5_Example4.fas")

# It may be convenient to manually modify files to get the appropriate names. However, an alternative is to use the
# Estimating distances between unique haplotypes
uniqueHaplo<-GetHaplo(input="5_Example4.fas",saveFile=FALSE)
distGap<-MCIC(readfile=FALSE,align=uniqueHaplo,saveFile=FALSE)
dist.nt <-dist.dna(uniqueHaplo,model="raw",pairwise.deletion=TRUE)
DISTnt<-as.matrix(dist.nt)

# Replacing sequence names by haplotype names in both distance matrices
for (Hi in 1:length(colnames(distGap)))
colnames(distGap)[Hi]<-FindHaplo(input="5_Example4.fas",saveFile=FALSE)[which(colnames(distGap)==colnames(distGap)[Hi])]
for (Hi in 1:length(colnames(DISTnt)))
colnames(DISTnt)[Hi]<-FindHaplo(input="5_Example4.fas",saveFile=FALSE)[which(colnames(DISTnt)==colnames(DISTnt)[Hi])]
row.names(DISTnt)<-colnames(DISTnt)

#Combining distance matrices and setting haplotype names
CombinedDistance<-as.data.frame(nt.gap.comb(DISTgap=distGap, range=0.5, method="CorrectedPairwiseDist"))
colnames(CombinedDistance)<-row.names(CombinedDistance)

# Estimating haplotype abundance per population and setting population names:
Haplotypes<-FindHaplo(input="5_Example4.fas",saveFile=FALSE)
Haplotypes[,1]<-substr(Haplotypes[,1],1,11)
Weighted<-as.data.frame(HapPerPop(readfile=FALSE,header=TRUE,input=Haplotypes)[1])
colnames(Weighted)<-substr(colnames(Weighted),10,11)

```

```
# Estimating population distances
pop.dist (DistFile=FALSE, distances=CombinedDistance, HaploFile=FALSE, Haplos=Weighted, c
```

Index

FindHaplo, [2](#)

GetHaplo, [3](#)

HapPerPop, [5](#)

MCIC, [6](#)

nt.gap.comb, [7](#)

perc.thr, [9](#)

pop.dist, [11](#)

sidier (*sidier-package*), [1](#)

sidier-package, [1](#)