Package 'phylter'

July 23, 2025

Title Detect and Remove Outliers in Phylogenomics Datasets

Version 0.9.11

Imports ape, ggplot2, reshape2, Rfast, RSpectra, stats, utils

Suggests knitr, rmarkdown

LinkingTo RcppEigen (>= 0.3.2.9.0), Rcpp (>= 0.12.6), RcppArmadillo
(>= 0.7.600.1.0)

License GPL (>= 2)

Description Analyzis and filtering of phylogenomics datasets.
It takes an input either a collection of gene trees (then transformed to matrices) or directly a collection of gene matrices and performs an iterative process to identify
what species in what genes are outliers, and whose elimination significantly improves the concordance between the input matrices. The methods builds upon the Distatis approach (Abdi et al. (2005) <doi:10.1101/2021.09.08.459421>), a generalization of classical multidimensional scaling to multiple distance matrices.

LazyData true

Depends R (>= 4.0)

BugReports <https://github.com/damiendevienne/phylter/issues>

Encoding UTF-8

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation yes

Author Damien M. de Vienne [aut],
Stéphane Dray [aut],
Théo Tricou [aut],
Aurélie Siberchicot [aut, cre]

Maintainer Aurélie Siberchicot <aurelie.siberchicot@univ-lyon1.fr>

Repository CRAN

Date/Publication 2023-08-24 08:20:03 UTC

Contents

---

carnivora	*125 genes trees for 53 carnivora species*

Description

This dataset contains 125 carnivora gene trees for 53 carnivora species. It is a subset of the dataset used in Allio et al. (2021), obtained by randomly selecting 125 genes for which all 53 species were present.

Usage

```
carnivora
```

Format

A list of named trees of class multiPhylo

Source

References

Allio, R., Tilak, M. K., Scornavacca, C., Avenant, N. L., Kitchener, A. C., Corre, E., ... & Delsuc, F. (2021). High-quality carnivoran genomes from roadkill samples enable comparative species delineation in aardwolf and bat-eared fox. Elife, 10, e63167.

Examples

```
data(carnivora)
class(carnivora)
```

---

detect.outliers   *Detection of outliers in 1D and 2D data*

Description

Functions to detect outliers, in matrices or in arrays.

Usage

```
detect.outliers(X, k = 3, test.island = TRUE, normalizeby = "row")

detect.outliers.array(arr, nbsp, k = 3)
```

Arguments

X	2D matrix (gene x species) obtained with the `Dist2WR()` function.
k	strength of outlier detection. High values (typically 3) correspond to less outliers detected than with lower ones (e.g. 1.5).
test.island	should islands of outliers be treated as such. If TRUE (the default), only the highest value in an island of outliers is removed. This prevents erroneous outliers due to hicthiking effect to be removed.
normalizeby	Should the 2D matrix be normalized prior to outlier detection, and how. Can be "row" (the default),"col" or "none". Normalization is done by dividing columns or rows by their median.
arr	Array of values, typically the weight of each gene matrix (alpha values).
nbsp	Number of species in the analysis

Details

These functions detect outliers either in matrices or in arrays. For the method to be adapted to skewed data, as is the case here, the outlier detection method used is the adjusted Tukey proposed by Hubert and Vandervieren (2008).

Value

detect.outliers: A matrix with outliers detected in the 2D matric. Each row x contains the gene (x[1]) where the species (x[2]) is an outlier.

detect.outliers.array: An array listing the outliers detected (if any)

Functions

- detect.outliers(): detect outliers in 2D matrix
- detect.outliers.array(): detects outliers in 1D array

References

de Vienne D.M., Ollier S. et Aguileta G. (2012) Phylo-MCOA: A Fast and Efficient Method to Detect Outlier Genes and Species in Phylogenomics Using Multiple Co-inertia Analysis. Molecular Biology and Evolution 29 : 1587-1598.

Hubert, M. and Vandervieren, E. (2008). An adjusted boxplot for skewed distributions. Computational Statistics and Data Analysis, 52, 5186-5201.

Examples

```
# Get the initial gene x species matrix
# from the carnivora dataset
data(carnivora)
mat <- phylter(carnivora, InitialOnly = TRUE, parallel = FALSE)$WR

# detect outliers in this matrix
outliers<-detect.outliers(mat)
outliers$cells # matrix where each row represents one cell in the input matrix
```

Dist2WR   *Compute gene x species matrix from the result of Distatis*

Description

Dist2WR computes the 2WR matrix from the results obtaind with DistatisFast (the fast version of distatis). For internal use mostly.

Usage

```
Dist2WR(Distatis)
```

Arguments

Distatis   output of the fonction DistatisFast.

Value

A matrix (gene=rows x species=col). Each cell represents a gene/species pair, whose value represents the distance between (i) the position of this species in this gene tree and (ii) the average position of this species in all other gene trees.

Examples

```
data(carnivora)
matrices<-phylter(carnivora, InitialOnly=TRUE, parallel=FALSE)$matrices
ds<-DistatisFast(matrices, parallel = FALSE)
WR<-Dist2WR(ds) #returns the gene x species matrix
```

DistatisFast	*Fast implementation the multivariate analysis method Distatis*

Description

New implementation of the DISTATIS method for K matrices of dimension IxI. This version of Distatis is faster than the original one because only the minimum required number of eigenvalues and eigenvectors is calculated. The difference in speed is particularly visible when the number of matrices is large.

Usage

```
DistatisFast(matrices, factorskept = "auto", parallel = TRUE)
```

Arguments

matrices	A list of K distance matrices, all of the same dimension (IxI).
factorskept	Number of factors to keep for the computation of the factor. When "auto" (the default), a brokenstick model is used for the choice of the number of components to keep.
parallel	Should the matrix products computations be parallelized? Default to TRUE. scores of the observations.

Value

Returns a list containing:

- `F`: projected coordinates of species in the compromise
- `PartialF`: list of projected coordinates of species for each gene.
- `alpha`: array of length K of the weight associated to each matrix.
- `lambda`: array of length K of the normalization factors used for each matrix. lambda=1 always.
- `RVmat`: a KxK matrix with RV correlation coefficient computed between all pairs of matrices.

- compromise: an IxI matrix representing the best compromise between all matrices. This matrix is the weighted average of all K matrices, using 'alpha' as a weighting array.

- quality: the quality of the compromise. This value is between 0 and 1

- matrices.dblcent: matrices after double centering describes how much of the variance of the K matrices is captured by the compromise.

References

Abdi, H., Valentin, D., O'Toole, A.J., & Edelman, B. (2005). DISTATIS: The analysis of multiple distance matrices. Proceedings of the IEEE Computer Society: International Conference on Computer Vision and Pattern Recognition_. (San Diego, CA, USA). pp. 42-47.

Examples

```
# Get a list of matrices
# from the carnivora dataset
data(carnivora)
matrices <- phylter(carnivora, InitialOnly = TRUE, parallel = FALSE)$matrices

# Perform a Distatis analysis on these matrices:
distatis <- DistatisFast(matrices, parallel = FALSE)

#distatis is a list with multiple elements:
distatis$alpha #weigh of each matrix (how much it correlates with others)
distatis$RVmat #RV matrix: correlation of each matrix with each other
distatis$compromise # distance matrix with "average" pairwise distance between species in matrices
# etc.
```

---

impMean | *Imputation of missing values in a collection of matrices*

Description

Impute missing data in a list of matrices. Matrices are first given the same dimension, then missing entries are filled by computing the average value in non-missing corresponding entries in all matrices.

Usage

```
impMean(matrices)
```

Arguments

matrices   A list of distance matrices.

Value

Returns a list of matrices with same dimensions, with rows and columns in the same order and missing data (if any) imputed.

Examples

```
data(carnivora)
matrices <- phylter(carnivora, InitialOnly = TRUE, parallel = FALSE)$matrices

# remove n species randomly (n between 1 and 5) in each matrix (to mimic missing data)
fun<-function(mat) {
 species2remove<-sample(1:nrow(mat),sample(1:5,1))
 mat<-mat[-species2remove,-species2remove]
 return(mat)
}
matrices.missing<-lapply(matrices, fun)
#check that all matrices have now different dimensions:
lapply(matrices.missing, dim)
# Impute data to get back to the same dimensions
matrices.ok<-impMean(matrices.missing)
lapply(matrices.ok, dim) #all dimensions are now identical. Missing data have been imputed.
```

---

medcouple	*A robust measure of skewness for univariate data*

Description

Computes the medcouple, a robust measure of skewness for univariate data. For multivariate data the medcouple is calculated on each column of the data matrix.

Usage

```
medcouple(x, do.reflect = NULL)
```

Arguments

x	An n by p data matrix.
do.reflect	Logical indicating whether the medcouple should also be computed on the reflected sample -x, with final result $(mc(\mathsf{x}) - mc(-\mathsf{x}))/2$. Defaults TRUE when $n <= 100$ and to FALSE otherwise.

Details

The medcouple is a robust measure of skewness yielding values between -1 and 1. For left- and right-skewed data the medcouple is negative and positive respectively.
The medcouple is defined as the median of the kernel function $h(x_i, x_j) = \frac{(x_j - med(x)) - (med(x) - x_i)}{x_j - x_i}$ evaluated over all couples (x_i, x_j) where x_i is smaller than the median of x and x_j larger than

the median of x. When there are multiple observations tied to the median, the kernel is defined separately as the denominator is not defined for these observations. Let $m_1 < ... < m_k$ denote the indices of the observations which are tied to the median. Then $h(x_{m_i}, x_{m_j})$ is defined to equal -1 if $i + j - 1 < k$, 0 when $i + j - 1 = k$ and $+1$ if $i + j - 1 > k$. To compute the medcouple an algorithm with time complexity $O(n log(n))$ is applied. For details, see https://en.wikipedia.org/wiki/Medcouple. For numerical accuracy it is advised, for small data sets, to compute the medcouple on both x and -x. The final value of the medcouple may then be obtained as a linear combination of both calculations. This procedure is warranted by the properties of the medcouple. Indeed the medcouple of the distribution X equals minus the medcouple of the reflected distribution $-X$. Moreover the medcouple is location and scale invariant. Note that missing values are not allowed.

Value

mc A p-vector containing the medcouple of each column of the data matrix x.

Note

This function is extracted from the package mrfDepth - 07/2023

Author(s)

P. Segaert with original code from M. Maechler and G. Brys.

References

Brys G., Hubert M., Struyf A. (2004). A robust measure of skewness. *Journal of Computational and Graphical Statistics*, **13**, 996–1017.

Examples

```
# Calculate the medcouple of univariate data sets.
# For 2000 normally distributed values
# the medcouple value is close to 0 because
# data are not skewed
x<-rnorm(2000)
medcouple(x)
# For 2000 values following a lognormal
# distribution (mean 0,sd 1), medcouple is close to 1
# because values are right-skewed
y<-rnorm(2000)
medcouple(y)
# Use the option do.reflect to increase expected accuracy.
medcouple(y, do.reflect = TRUE)
```

---

normalize   *Median normalization of 2D matrix by row or by colomn*

Description

This function normalizes the 2WR matrix (or any 2D matrix) according to the species (rows) or to the genes (columns), by dividing each row or each column by its median.

Usage

```
normalize(mat, what = "none")
```

Arguments

mat   A matrix

what   Character string indicating whether the matrix should be normalized and how. If
                what="none", the matrix is not normalized (the default), if what="species", the
                matrix is normalized so that the difference between species is increased, and if
                what="genes", the matrix is normalized so that the difference between genes is
                increased. Normalization consists in dividing either each row or each columns
                by its median.

Value

A normalized matrix

Examples

```
# random matrix
x<-matrix(rnorm(270), nrow=9, ncol=14)

# normalize by row
x1<-normalize(x, "genes")

# normalize by column
x2<-normalize(x, "species")
```

---

phylter   *Filter phylogenomics datasets*

Description

Detection and filtering out of outliers in a list of trees or a list of distance matrices.

Usage

```
phylter(
  X,
  bvalue = 0,
  distance = "patristic",
  k = 3,
  k2 = k,
  Norm = "median",
  Norm.cutoff = 0.001,
  gene.names = NULL,
  test.island = TRUE,
  verbose = TRUE,
  stop.criteria = 1e-05,
  InitialOnly = FALSE,
  normalizeby = "row",
  parallel = TRUE
)
```

Arguments

X	A list of phylogenetic trees (phylo object) or a list of distance matrices. Trees can have different number of leaves and matrices can have different dimensions. If this is the case, missing values are imputed.
bvalue	If X is a list of trees, nodes with a support below 'bvalue' will be collapsed prior to the outlier detection.
distance	If X is a list of trees, type of distance used to compute the pairwise matrices for each tree. Can be "patristic" (sum of branch lengths separating tips, the default) or nodal (number of nodes separating tips). The "nodal" option should only be used if all species are present in all genes.
k	Strength of outlier detection. The higher this value the less outliers detected (see details).
k2	Same as k for complete gene outlier detection. To preserve complete genes from being discarded, k2 can be increased. By default, k2 = k. (see above)
Norm	Should the matrices be normalized prior to the complete analysis and how. If "median" (the default), matrices are divided by their median, if "mean" they are divided by their mean, if "none", no normalization if performed. Normalizing ensures that fast-evolving (and slow-evolving) genes are not treated as outliers. Normalization by median is a better choice as it is less sensitive to outlier values.

Norm.cutoff	Value of the median (if Norm="median") or the mean (if Norm="mean") of phylogenetic distance matrices below which genes are simply discarded from the analysis. This prevents dividing by 0, and allows getting rid of genes that contain mostly branches of length 0 and are therefore uninformative anyway. Discarded genes, if any, are listed in the output (out$DiscardedGenes).
gene.names	List of gene names used to rename elements in X. If NULL (the default), 0 elements are named 1,2,..,length(X).
test.island	If TRUE (the default), only the highest value in an 'island' of outliers is considered an outlier. This prevents non-outliers hitchhiked by outliers to be considered outliers themselves.
verbose	If TRUE (the default), messages are written during the filtering process to get information of what is happening
stop.criteria	The optimisation stops when the gain in concordance between matrices between round n and round n+1 is smaller than this value. Default to 1e-5.
InitialOnly	Logical. If TRUE, only the Initial state of the data is computed.
normalizeby	Should the gene x species matrix be normalized prior to outlier detection, and how.
parallel	Should the computations be parallelized when possible? Default to TRUE. Note that the number of threads cannot be set by the user when 'parallel=TRUE'. It uses all available cores on the machine.

Value

A list of class 'phylter' with the 'Initial' (before filtering) and 'Final' (after filtering) states, or a list of class 'phylterinitial' only, if InitialOnly=TRUE. The function also returns the list of DiscardedGenes, if any.

Examples

```
data(carnivora)

# using default paramaters
res <- phylter(carnivora, parallel = FALSE) # perform the phylter analysis
res # brief summary of the analysis
res$DiscardedGenes # list of genes discarded prior to the analysis
res$Initial # See all elements prior to the analysis
res$Final # See all elements at the end of the analysis
res$Final$Outliers # Print all outliers detected


# Change the call to phylter to use nodal distances, instead of patristic:
res <- phylter(carnivora, distance = "nodal")
```

---

plot.phylter	*Plot phylter objects*

Description

These functions take objects of class phylter as input and display various plots summarizing the results obtained (see details).

Usage

```
## S3 method for class 'phylter'
plot(x, what = "all", layout = 1, sorted = TRUE, ...)

plot2WR(
  x,
  show.missing = TRUE,
  show.outliers = TRUE,
  transpose = FALSE,
  clust = FALSE
)

plotDispersion(x)

plotRV(x, what = "Initial", labelnames = TRUE, clust = FALSE)

plotopti(x)
```

Arguments

x	The object returned by the 'phylter()' function.
what	Specifies what to plot. If "species", a barplot will show how many genes each species is in, and what proportion of thoses were detected as outliers. If "genes", a barplot shows how many species contains each gene and how many of them has been detected as outliers. If "all" (the defaut), the two plots described above are displayed one after the other, prompting the user to press ENTER to display the next one.
layout	What layout to use. Do not change if you don't know what it is.
sorted	Logical. Should bars be sorted by the number of outliers detected. Default to TRUE
...	Additional arguments to be passed to plot and print functions.
show.missing	Logical. Should missing data be represented on the heatmap. If TRUE (the default), white dots show were these missing entries are in both the initial and final 2WR matrices.
show.outliers	Logical. Should outliers be represented on the heatmap. If TRUE (the default), yellow dots indicate outliers on the final 2WR matrix.

transpose	Logical. If TRUE, the two matrices are piled up instaed of being displayed side by side. Default to FALSE.
clust	Logical. Should the rows or/and columns of the matrices that are plotted be reorderd prior to plotting. Reordering is based on a hierarchical clustering. Default to FALSE. This is conveninent when the names of the genes are very long for instance.
labelnames	Logical. If TRUE, the names of labels are indicated on the heatmap. If FALSE they are removed. This is conveninent when the names of the genes are very long for instance.

Details

- `plot(x)` and `plot.phylter(x)` plot the genes found in each species or species found in each gene as barplots, highlighting the outliers detected.

- `plot2WR(x)` plots side by side the initial and the final gene x species (unreable for large datasets) matrices (the 2WR matrices), highlighting missing data and detected outliers.

- `plotDispersion(x)` plots dispersion of data before and after phylter, on a 2D space. Each dot represents a gene-species association.

- `plotRV(x)` plots the RV coefficient matrix that descibes all agains all correlations between gene matrices

- `plotopti(x)` plots the compromise matrix score at each step of the optimization.

Value

The desired plots are returned. Note that you might want to call the pdf(), png(), jpeg(), or tiff() function first if you want to save the plot(s) to an external file. You can also use the function `write.phylter(pdfreport=TRUE)` to write all possible plots in a single pdf file.

Examples

```
data(carnivora)

# perform phylter analysis
res <- phylter(carnivora, parallel = FALSE)

# plot for each gene the number of outlier species
plot(res, "genes")


# plot for each species the number genes where it is outlier
plot(res, "species")

# plot the dispersion of data before and after the use of phylter
plotDispersion(res)
```

PreparePhylterData　　　　　*Prepare data for phylter analysis*

Description

Prepare datasets for the `phylter` function. Detects possible issues, discards genes if necessary, imputes missing data if any, and reorders row- and col-names. For internal usage mostly.

Usage

```
PreparePhylterData(
  X,
  bvalue = 0,
  distance = "patristic",
  Norm = "median",
  Norm.cutoff = 0.001,
  gene.names = NULL,
  verbose = TRUE
)
```

Arguments

X	A list of phylogenetic trees (phylo object) or a list of distance matrices. Trees can have different number of leaves and matrices can have different dimensions. If this is the case, missing values are imputed.
bvalue	If X is a list of trees, nodes with a support below 'bvalue' will be collapsed prior to the outlier detection.
distance	If X is a list of trees, type of distance used to compute the pairwise matrices for each tree. Can be "patristic" (sum of branch lengths separating tips, the default) or nodal (number of nodes separating tips).
Norm	Should the matrices be normalized and how. If "median", matrices are divided by their median, if "mean" they are divided by their mean, if "none", no normalization if performed. Normalizing ensures that fast-evolving (and slow-evolving) genes are not treated as outliers. Normalization by median is less sensitive to outlier values but can lead to errors if some matrices have a median value of 0. are not considered outliers.
Norm.cutoff	Value of the median (if Norm="median") or the mean (if Norm="mean") below which matrices are simply discarded from the analysis. This prevents dividing by 0, and getting rid of genes that contain mostly branches of length 0 and are therefore uninformative anyway.
gene.names	List of gene names used to rename elements in X. If NULL (the default), elements are named 1,2,..,length(X).
verbose	If TRUE (the default), messages are written during the filtering process to get information of what is happening

Value

A list of class 'phylter' with the 'Initial' (before filtering) and 'Final' (after filtering) states, or a list of class 'phylterinitial' only, if InitialOnly=TRUE.

Examples

```
data(carnivora)
# transform trees to a named list of matrices with same dimensions
# and identical row and column orders and names
carnivora.clean<-PreparePhylterData(carnivora)
```

---

print.phylter   *Print phylter objects*

Description

Print on screen a simple description of the content of phylter objects

Usage

```
## S3 method for class 'phylter'
print(x, ...)
```

Arguments

x	Object returned by function 'phylter()'.
...	Additional arguments.

Value

Print formatting

Examples

```
data(carnivora)
res <- phylter(carnivora, parallel = FALSE)
print(res)
```

---

print.phylterfinal   *print objects of class phylterfinal*

Description

Print on screen a simple description of the content of objects of class `phylterfinal`.

Usage

```
## S3 method for class 'phylterfinal'
print(x, ...)
```

Arguments

x   Object returned by function 'phylter()'.

...   Additional arguments.

Value

NA

Examples

```
data(carnivora)
res <- phylter(carnivora, parallel = FALSE)
print(res$Final)
```

---

print.phylterinitial   *print objects of class phylterinitial*

Description

Print on screen a simple description of the content of objects of class `phylterinitial`.

Usage

```
## S3 method for class 'phylterinitial'
print(x, ...)
```

Arguments

x   Object present in the `$initial` element of the object returned by function `phylter()`.

...   Additional arguments.

Value

NA

Examples

```
data(carnivora)
res <- phylter(carnivora, parallel = FALSE)
print(res$Initial)
```

---

print.summary.phylter   *print summary of phylter objects*

Description

Prints on screen the summary of an object of class summmary.phylter as returned by the summary.phylter()
function.

Usage

```
## S3 method for class 'summary.phylter'
print(x, ...)
```

Arguments

x   Object returned by function 'summary.phylter()'.

...   Additional arguments.

Value

NA

Examples

```
data(carnivora)
res <- phylter(carnivora, parallel = FALSE)
summary <- summary(res)
print(summary)
```

---

rename.genes   *Name or rename a list of gene trees or matrices*

Description

Name or rename a list of gene trees or gene matrices genes. For internal use mostly.

Usage

```
rename.genes(X, gene.names = NULL)
```

Arguments

X   A list of trees or matrices

gene.names   List of names to assign to the elements of X. Must be of the same length as
                     length(X). If NULL (the default) the object are numbered 1,2,...,length(X).

Value

X with name assigned to each element.

Examples

```
data(carnivora)

# names before renaming
names(carnivora)

carnivora.renamed<-rename.genes(carnivora, gene.names=as.character(1:length(carnivora)))

# names after renaming
names(carnivora.renamed)
```

---

simtrees   *Simplistic simulation of gene trees with outliers*

Description

Simple (simplistic) simulation of trees with outliers.

Usage

```
simtrees(Ngn, Nsp, Nsp.out = 0, Ngn.out = 0, Nb.cell.outlier = 0,
brlen.sd = 0, out.type="topology",bl.mult=2)
```

Arguments

Ngn	Number of gene trees to simulate.
Nsp	Number of species (tips) per tree.
Nsp.out	Number of outlier species (also called rogue taxa). 0 = none.
Ngn.out	Number of outlier genes. 0 = none.
Nb.cell.outlier	
	Number of times one species in one tree is an outlier. 0 = none. The type of outlier is set by out.type=.
brlen.sd	Heterogeneity of branch lengths in trees. A value with mean 0 and standard deviation equal to brlen.sd is added to each branch length. If the resulting branch length has a negative value, its absolute value is taken.
out.type	The type of cell outlier. Can be "topology" (the default), where outlier species are simulated by branching them elsewhere (randomly) in the tree, "brlength", where terminal branches of outlier species are multiplied by bl.mult (see after) or "both" where half of the outliers are "topology" and the other half are "brlength". In the latter case, if the number of outliers (Nb.cell.outlier) id odd, there is one more topology outlier than brlength outlier simulated.
bl.mult	Multiplier of terminal branches of outlier species when out.type="topology" or out.type="both". Ignored otherwise.

Details

The simulation process is as follows: a first tree is generated with the rtree() function and is then duplicated and modified according to the parameters chosen by the user.

Value

A list X containing a list of trees in multiPhylo format (X$trees) and a list of outliers (X$outl).

Examples

```
# Very basic simulator, for debugging purpose mainly.
# examples: 30 genes, 120 species, 2 outlier species, 3 outlier genes
# 4 gene/species outliers of type "topology", branch length variance = 0.6.
# The branch length multiplier is set to 2 but this is
# ignored if out.type="topology"
simu<-simtrees(30,120,2,3,4,0.6, "topology",2)
trees<-simu$trees #list of trees
outl<-simu$outl #list of simulated outliers and their type
```

---

summary.phylter   *Get summary for phylter objects*

Description

Get the summary of an object of class `phylter`.

Usage

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

Arguments

object	Object returned by function 'phylter()'.
...	Additional arguments affecting the summary produced.

Value

On object of class `summmary.phylter`

Examples

```
data(carnivora)
res <- phylter(carnivora, parallel = FALSE)
summary <- summary(res)
class(summary)
```

---

trees2matrices   *Convert phylogenetic trees to distance matrices*

Description

Transform a list of phylogenetic trees into a list of phylogenetic distance matrices.

Usage

```
trees2matrices(trees, distance = "patristic", bvalue = 0)
```

Arguments

`trees`	A list of gene trees in format "multiphylo".
`distance`	A method to generate distance matrices. It could be "nodal" to establish that the distance between two species is the number of nodes that separate them. Or "patristic" (default) if the distance between two species is the sum of branch lengths separating them. The "nodal" option should only be used if all species are present in all trees (no missing data).
`bvalue`	This argument is only used if trees contain bootstrap values. It determines under what bootstrap values the nodes should be collapsed. Value 0 (the default) means that no nodes are collapsed.

Value

return a list of distance matrices

Examples

```
data(carnivora)
matrices<-trees2matrices(carnivora)
```

`write.phylter`	*Write summary of phyter analysis to file(s)*

Description

Write the summary of the phylter analysis in a txt file or as a pdf report.

Usage

```
write.phylter(
  x,
  file = "",
  include.discarded = TRUE,
  pdfreport = FALSE,
  pdfreport.file = "report.pdf"
)
```

Arguments

`x`	The object returned by the 'phylter()' function.
`file`	Name of the file where to write the text summary of the phylter output. If `""` (the default), `write.phylyer()` prints to the standard output.
`include.discarded`	
	Logical. If TRUE (the default) the elements discarded before the analysis are still in the list of Outliers in the output. Useful for cleaning datasets after a phylter analysis.

pdfreport   Logical. Should a full report of the phylter analysis

pdfreport.file If report=TRUE, name of the pdf file where the report is written. Default to
                 report.pdf

Value

No return value

Examples

```
data(carnivora)
res <- phylter(carnivora, parallel = FALSE)
# write a full report to the standard output
write.phylter(res)
```

Index