Package 'SigTree'

January 20, 2025

Type Package

Title Identify and Visualize Significantly Responsive Branches in a
Phylogenetic Tree

Version 1.10.6

Date 2017-09-29

Author John R. Stevens and Todd R. Jones

Maintainer John R. Stevens <john.r.stevens@usu.edu>

Description Provides tools to identify and visualize branches in a phylogenetic tree that are signifi-
cantly responsive to some intervention, taking as primary inputs a phyloge-
netic tree (of class phylo) and a data frame (or matrix) of corresponding tip (OTU) labels and p-
values.

Imports ape, phylobase, phyext2, RColorBrewer, phyloseq, graphics,
utils, stats, MASS, vegan

Repository CRAN

License GPL-3

NeedsCompilation yes

Date/Publication 2017-10-02 20:53:17 UTC

Contents

SigTree-package   *Determine significantly responsive branches in phylogenetic trees*

Description

SigTree is a package of functions to determine significant response of branches of phylogenetic trees and produce colored plots both in R and (via exported .tre file) FigTree. plotSigTree takes a phylogenetic tree (of class phylo) and a data frame (or matrix) of corresponding tip (OTU) labels and p-values and determines the significance of the branches (as families of p-values) and plots the tree with colored branches (corresponding to families) according to the level of significance of the branch. export.inherit produces a CSV file (or data frame) with the p-values for all branches as well as which tips belong to which branches. export.figtree exports a .tre file that can be opened in FigTree that produces a colored plot (with colors according to the significance of corresponding branches) with p-value annotations.

Details

Package:	SigTree
Type:	Package
Version:	1.10.6
Date:	2017-09-29
License:	GPL-3

For more information, see the documentation for plotSigTree, export.inherit, and export.figtree.

To access the tutorial document for this package, type in R: vignette("SigTree")

Author(s)

John R. Stevens and Todd R. Jones

Maintainer: John R. Stevens <john.r.stevens@usu.edu>

References

Stevens J.R., Jones T.R., Lefevre M., Ganesan B., and Weimer B.C. (2017) "SigTree: A Microbial Community Analysis Tool to Identify and Visualize Significantly Responsive Branches in a Phylogenetic Tree." Computational and Structural Biotechnology Journal 15:372-378.

Jones T.R. (2012) "SigTree: An Automated Meta-Analytic Approach to Find Significant Branches in a Phylogenetic Tree" (2012). MS Thesis, Utah State University, Department of Mathematics and Statistics. http://digitalcommons.usu.edu/etd/1314

FigTree is available at http://tree.bio.ed.ac.uk/software/figtree/.

| adonis.tree | *Function to perform adonis test of independence on p-values from tests of multiple OTUs.* |

Description

adonis.tree takes `tree` and `unsorted.pvalues` and computes a p-value corresponding to a test for significant differences among the p-values in `unsorted.pvalues` based on the between-OTU distances in the phylogenetic tree `tree`.

Usage

```
adonis.tree(tree, unsorted.pvalues, seed=1234, perms=10000, z=TRUE, make2sided=TRUE)
```

Arguments

tree	a phylogenetic tree of class `phylo`.
unsorted.pvalues	
	a data frame (or matrix) with tip labels in column 1 and p-values in column 2. The tip labels must correspond to the tip labels in `tree`.
seed	positive integer seed value, to force reproducibility of permutations.
perms	number of permutations to employ for adonis test
z	logical argument (TRUE or FALSE) indicating whether or not to convert p-values to corresponding standard normal (Z) variates, on which scale the adonis test would subsequently be performed.
make2sided	logical argument (TRUE or FLASE) indicating whether or not to convert p-values to two-sided; this should be TRUE whenever `unsorted.pvalues` are one-sided p-values.

Details

After converting p-values to corresponding standard normal (Z) variates (when `make2sided=TRUE`), and obtaining the distance matrix of between-OTU distances, this function employs the `adonis` function of the package vegan. This effectively results in a test of whether the OTU p-values are independent (the null hypothesis here), or whether differences among the OTU p-values are associated with between-OTU distances.

The "adonis" method was apparently originally called "anodis", for "analysis of dissimilarities". To more easily distinguish this method from ANOSIM ("analysis of similarities", which also handles dissimilarities), it was re-named "anodis". According to the help file for `adonis`, "Most anosim models could be analyzed with adonis, which seems to be a more robust alternative" because it is less sensitive to dispersion effects (Warton et al., 2012).

To access the tutorial document for this package (including this function), type in R: `vignette("SigTree")`

Value

This function returns a single numeric value, corresponding to a p-value of null: "p-values for OTUs are independent" vs. alternative: "OTU p-value differences are associated with pairwise OTU distances".

Author(s)

John R. Stevens

References

Stevens J.R., Jones T.R., Lefevre M., Ganesan B., and Weimer B.C. (2017) "SigTree: A Microbial Community Analysis Tool to Identify and Visualize Significantly Responsive Branches in a Phylogenetic Tree." Computational and Structural Biotechnology Journal 15:372-378.

Jones T.R. (2012) "SigTree: An Automated Meta-Analytic Approach to Find Significant Branches in a Phylogenetic Tree" (2012). MS Thesis, Utah State University, Department of Mathematics and Statistics. http://digitalcommons.usu.edu/etd/1314

Anderson, M.J. (2001) "A new method for non-parametric multivariate analysis of variance." Austral Ecology, 26: 32-46.

Reiss P.T., Stevens M.H.H., Shehzad Z., Petkova E., and Milham M.P. (2010) "On Distance-Based Permutation Tests for Between-Group Comparisons." Biometrics 66:636-643.

Warton, D.I., Wright, T.W., Wang, Y. (2012) "Distance-based multivariate analyses confound location and dispersion effects." Methods in Ecology and Evolution, 3, 89-101.

Examples

```
### To access the tutorial document for this package, type in R (not run here):
# vignette('SigTree')

### Create tree, then data frame, then use plotSigTree to plot the tree
### Code for random tree and data frame
node.size <- 10
seed <- 109
# Create tree
set.seed(seed);
library(ape)
r.tree <- rtree(node.size)
# Create p-values data frame
set.seed(seed)
r.pval <- rbeta(node.size, .1, .1)
# Randomize the order of the tip labels
# (just to emphasize that labels need not be sorted)
set.seed(seed)
r.tip.label <- sample(r.tree$tip.label, size=length(r.tree$tip.label))
r.pvalues <- data.frame(label=r.tip.label, pval=r.pval)

# Check for dependence among p-values; lack of significance here
# indicates default test="Stouffer" would be appropriate in other
# main SigTree package functions (plotSigTree, export.figtree,
```

```
# and export.inherit); otherwise, test="Hartung" would be more
# appropriate.
adonis.tree(r.tree,r.pvalues)
```

| export.figtree | *Function to export a* NEXUS *file that can be opened in* FigTree *to pro-duce a plot of the phylogenetic tree with branches colored according to significance of families of p-values* |

Description

export.figtree takes tree and unsorted.pvalues and produces a NEXUS file that can FigTree can subsequently open. The p-values for each branch (family of tips) are computed and the branches are colored accordingly. It computes the p-values based on arguments involving p-value adjustment (for multiple hypothesis testing) and either Stouffer's or Fisher's p-value combination method. There are arguments that allow for the customization of the p-value cutoff ranges as well as the colors to be used in the coloring of the branches. There is also an option to include annotations for each edge that contain the p-value for the corresponding branch.

Usage

```
export.figtree(tree, unsorted.pvalues, adjust=TRUE, side=1,
method="hommel", p.cutoffs=ifelse(rep(side==1, ifelse(side==1, 6, 3)),
c(.01, .05, .1, .9, .95, .99), c(.01, .05, .1)), file="",
pal=ifelse(rep(side==1, ifelse(side==1, 1, length(p.cutoffs)+1)),
"RdBu", rev(brewer.pal(length(p.cutoffs)+1,"Reds"))),
test = "Stouffer", edge.label=TRUE, ignore.edge.length=FALSE,
branch="edge")
```

Arguments

tree	a phylogenetic tree of class phylo.
unsorted.pvalues	
	a data frame (or matrix) with tip labels in column 1 and p-values in column 2. The tip labels must correspond to the tip labels in tree.
adjust	a logical argument that controls whether there is p-value adjustment performed (TRUE) or not (FALSE).
side	a numerical argument that takes values 1 and 2, depending on whether the p-values in unsorted.pvalues are 1-sided or 2-sided, respectively. Only used in p-value adjustment if adjust = TRUE.
method	one of the p-value adjustment methods (used for multiple-hypothesis testing) found in p.adjust.methods ("holm", "hochberg", "hommel", "BH", "bonferroni", "BY", "fdr", and "none"). See help for p.adjust for more details on these methods. method is only used if adjust = TRUE.

p.cutoffs   a vector of increasing p-value cutoffs (excluding 0 and 1) to determine the ranges
                   of p-values used in the coloring of the branches.

file   the file path that the NEXUS file that FigTree can open is written to.

pal   one of the palettes from the RColorBrewer package (see brewer.pal.info for
                   a list) or a vector of hexadecimal colors. These are the colors that are used to
                   color the branches. The first color corresponds to the first range in p.cutoffs and
                   so on.

test   a character string taking on "Hartung", "Stouffer", or "Fisher". This is
                   the p-value combination method that will be used. In most cases, "Stouffer"
                   will be most appropriate, unless adonis.tree indicates significant evidence of
                   dependence among p-values, in which case "Hartung" is preferred.

edge.label   a logical argument that, when TRUE, produces edge label annotations in FigTree.
                   These annotations are the p-values of the edge's parent node when branch="node".
                   They are accessible via highlighting an edge, hitting the Annotate button, and
                   selecting P-value.

ignore.edge.length
                   a logical parameter. When TRUE, the edge lengths from tree are ignored and
                   FigTree creates its own edge lengths. ignore.edge.length has no effect if
                   the edges in tree have no defined length.

branch   a character controlling branch definition: "edge" and "node" are the only op-
                   tions. This does not affect statistical methods, only the colors used in edge
                   coloring. Prior to package version 1.2, only branch="node" was implemented.

Details

The tip labels of tree (accessed via tree$tip.label) must have the same names (and the same
length) as the tip labels in unsorted.pvalues, but may be in a different order. The p-values in
column 2 of unsorted.pvalues obviously must be in the [0, 1] range. p.cutoffs takes values
in the (0, 1) range. The default value for p.cutoffs is c(0.01, 0.05, 0.1, 0.9, 0.95, 0.99) if
side is 1 and c(0.01, 0.05, 0.1) if side is 2. Thus, the ranges (when side is 1) are: [0, .01], (.01,
.05], ..., (.99, 1]. These ranges correspond to the colors specified in pal. P-values in the [0, .01]
range correspond to the left-most color if pal is a palette (view this via display.brewer.pal(x,
pal) - where x is the number of colors to be used) or the first value in the vector if pal is a vector
of colors. If pal is a vector of colors, then the length of pal should be one greater than the length of
p.cutoffs. In other words, its length must be the same as the number of p-value ranges. In addition,
each color in this vector of colors needs to be in hexadecimal format, for example, "#B2182B".
Formats of colors other than hexadecimal will likely give unwanted results in the edges of the tree
produced in *FigTree*, such as all-black edges or the edges being colored in a meaningless way. This
is because the color conversion assumes hexadecimal colors. The default value of pal is "RdBu"
(a divergent palette of reds and blues, with reds corresponding to small p-values) if side is 1 and
the reverse of "Reds" (a sequential palette) if side is 2. The sequential palettes in RColorBrewer
go from light to dark, so "Reds" is reversed so that the dark red corresponds to small p-values. It
probably makes more sense to use a divergent palette when using 1-sided p-values and a sequential
palette (reversed) when using 2-sided p-values. To create a vector of reversed colors from a palette
with x number of colors and "PaletteName" as the name of the palette, use rev(brewer.pal(x,
"PaletteName")). ignore.edge.length may be useful to get a more uniformly-shaped tree.

`export.figtree` assumes that each internal node has exactly two descendants. It also assumes that each internal node has a lower number than each of its ancestors (excluding tips).

The `branch` argument controls whether edge coloring corresponds to the combined p-value of the tips below the edge (″edge″) or of the tips below the edge's leading (away from the tips) node (″node″). Note that if branch=″node″ is used, then both edges leaving a node will necessarily be colored the same.

To access the tutorial document for this package (including this function), type in R: `vignette("SigTree")`

Value

This function creates a `NEXUS` file that can be opened by the program `FigTree`.

Author(s)

John R. Stevens and Todd R. Jones

References

Stevens J.R., Jones T.R., Lefevre M., Ganesan B., and Weimer B.C. (2017) "SigTree: A Microbial Community Analysis Tool to Identify and Visualize Significantly Responsive Branches in a Phylogenetic Tree." Computational and Structural Biotechnology Journal 15:372-378.

Jones T.R. (2012) "SigTree: An Automated Meta-Analytic Approach to Find Significant Branches in a Phylogenetic Tree" (2012). MS Thesis, Utah State University, Department of Mathematics and Statistics. http://digitalcommons.usu.edu/etd/1314

FigTree is available at `http://tree.bio.ed.ac.uk/software/figtree/`.

Examples

```
### To access the tutorial document for this package, type in R (not run here):
# vignette("SigTree")

### Create tree, then data frame, then use plotSigTree to plot the tree
### Code for random tree and data frame
node.size <- 10
seed <- 109
# Create tree
set.seed(seed)
library(ape)
r.tree <- rtree(node.size)
# Create p-values data frame
set.seed(seed)
r.pval <- rbeta(node.size, .1, .1)
# Randomize the order of the tip labels
# (just to emphasize that labels need not be sorted)
set.seed(seed)
r.tip.label <- sample(r.tree$tip.label, size=length(r.tree$tip.label))
r.pvalues <- data.frame(label=r.tip.label, pval=r.pval)

# Check for dependence among p-values; lack of significance here
# indicates default test="Stouffer" is appropriate;
```

```
# otherwise, test="Hartung" would be more appropriate.
adonis.tree(r.tree,r.pvalues)

# Export "ExportFigtree1.tre" file that can be opened in FigTree
library(phyext2)
export.figtree(r.tree, r.pvalues, test="Stouffer", file="ExportFigtree1.tre")
```

---

export.inherit   *Function to produce a CSV file with the p-value for each branch as*
                                   well as a list of all of the tips in each branch's family

---

Description

export.inherit takes tree and unsorted.pvalues and produces a CSV file (or data frame) with
p-values for each branch (including tips) as well as a list of all of the tips that belong to each branch's
family (i.e., all of the tips that are descendants of the branch). The p-values are computed based
on arguments involving p-value adjustment (for multiple hypothesis testing) and either Stouffer's
or Fisher's p-value combination method.

Usage

```
export.inherit(tree, unsorted.pvalues, adjust = TRUE, side = 1, method = "hommel",
file = "", test = "Stouffer", frame = FALSE, branch="edge")
```

Arguments

tree	a phylogenetic tree of class phylo.
unsorted.pvalues	
	a data frame (or matrix) with tip labels in column 1 and p-values in column 2. The tip labels must correspond to the tip labels in tree.
adjust	a logical argument that controls whether there is p-value adjustment performed (TRUE) or not (FALSE).
side	a numerical argument that takes values 1 and 2, depending on whether the p-values in unsorted.pvalues are 1-sided or 2-sided, respectively. Only used in p-value adjustment if adjust = TRUE.
method	one of the p-value adjustment methods (used for multiple-hypothesis testing) found in p.adjust.methods ("holm", "hochberg", "hommel", "BH", "bonferroni", "BY", "fdr", and "none"). See help for p.adjust for more details on these methods. method is only used if adjust = TRUE.
file	the file path for the CSV file to be written to. If frame=TRUE, no CSV file is created. If frame=FALSE, a CSV file will only be created if file is specified.
test	a character string taking on "Hartung", "Stouffer", or "Fisher". This is the p-value combination method that will be used. In most cases, "Stouffer" will be most appropriate, unless adonis.tree indicates significant evidence of dependence among p-values, in which case "Hartung" is preferred.

frame	a logical argument that controls whether or not to return (in R) the resulting data.frame object. If FALSE, file must be specified.
branch	a character controlling branch definition: "edge" and "node" are the only options. This does not affect statistical methods, only the colors used in edge coloring. Prior to package version 1.2, only branch="node" was implemented. The edge and node numbering is internal to class phylo, and are not necessarily sequential numbers.

Details

The tip labels of tree (accessed via tree$tip.label) must have the same names (and the same length) as the tip labels in unsorted.pvalues, but may be in a different order. The p-values in column 2 of unsorted.pvalues obviously must be in the [0, 1] range. export.inherit assumes that each internal node has exactly two descendants. It also assumes that each internal node has a lower number than each of its ancestors (excluding tips).

To access the tutorial document for this package (including this function), type in R: vignette("SigTree")

Value

This function produces a CSV file; alternatively, if frame=TRUE, this function will return a data.frame object.

Author(s)

John R. Stevens and Todd R. Jones

References

Stevens J.R., Jones T.R., Lefevre M., Ganesan B., and Weimer B.C. (2017) "SigTree: A Microbial Community Analysis Tool to Identify and Visualize Significantly Responsive Branches in a Phylogenetic Tree." Computational and Structural Biotechnology Journal 15:372-378.

Jones T.R. (2012) "SigTree: An Automated Meta-Analytic Approach to Find Significant Branches in a Phylogenetic Tree" (2012). MS Thesis, Utah State University, Department of Mathematics and Statistics. http://digitalcommons.usu.edu/etd/1314

Examples

```
### To access the tutorial document for this package, type in R (not run here):
# vignette("SigTree")

### Create tree, then data frame, then use plotSigTree to plot the tree
### Code for random tree and data frame
node.size <- 10
seed <- 109
# Create tree
set.seed(seed);
library(ape)
r.tree <- rtree(node.size)
# Create p-values data frame
set.seed(seed)
```

```
r.pval <- rbeta(node.size, .1, .1)
# Randomize the order of the tip labels
# (just to emphasize that labels need not be sorted)
set.seed(seed)
r.tip.label <- sample(r.tree$tip.label, size=length(r.tree$tip.label))
r.pvalues <- data.frame(label=r.tip.label, pval=r.pval)

# Check for dependence among p-values; lack of significance here
# indicates default test="Stouffer" is appropriate;
# otherwise, test="Hartung" would be more appropriate.
adonis.tree(r.tree,r.pvalues)

# Create CSV file called "ExportInherit1.csv"
export.inherit(r.tree, r.pvalues, test="Stouffers", file="ExportInherit1.csv")

# Look at resulting file in R -- see package vignette
f <- export.inherit(r.tree, r.pvalues, test="Stouffers", frame=TRUE)
f
```

---

p2.p1   *Function to convert two-tailed p-values to one-tailed, for use by other*
                               SigTree functions.

Description

p2.p1 takes vectors p (representing two-sided p-values of null: Mean2=Mean1) and diff (representing Mean2-Mean1) and computes one-tailed p-values. One-tailed p-values are used by other SigTree functions, primarily plotSigTree, export.figtree, and export.inherit.

Usage

```
p2.p1(p,diff)
```

Arguments

p   vector of two-tailed p-values, corresponding to a test of null: Mean2=Mean1.

diff   vector of differences Mean2-Mean1, or a vector of the signs of the Mean2-
                Mean1 differences.

Details

This function has application when multiple tests (as at multiple OTUs) of some intervention have been performed, such as comparing the mean of a treatment 2 with the mean of a treatment 1. The resulting two-sided p-values can be converted to one-sided p-values, so that the tools of the SigTree package are applicable.

To access the tutorial document for this package (including this function), type in R: vignette("SigTree")

Value

This function produces a vector of one-sided p-values, corresponding to a test of null: Mean2=Mean1 vs. alternative: Mean2>Mean1.

Author(s)

John R. Stevens and Todd R. Jones

References

Stevens J.R., Jones T.R., Lefevre M., Ganesan B., and Weimer B.C. (2017) "SigTree: A Microbial Community Analysis Tool to Identify and Visualize Significantly Responsive Branches in a Phylogenetic Tree." Computational and Structural Biotechnology Journal 15:372-378.

Jones T.R. (2012) "SigTree: An Automated Meta-Analytic Approach to Find Significant Branches in a Phylogenetic Tree" (2012). MS Thesis, Utah State University, Department of Mathematics and Statistics. http://digitalcommons.usu.edu/etd/1314

Examples

```
### To access the tutorial document for this package, type in R (not run here):
# vignette('SigTree')

## Assume 10 OTUs are measured in each of
## 20 subjects receiving treatment 2, and
## 15 subjects receiving treatment 1.
## For each OTU, test null: Mean2=Mean1
## using a Wilcoxon Rank Sum test.
## Simulate data, and obtain p-values and differences
set.seed(1234)
library(MASS)
X2 <- mvrnorm(n=20, mu=runif(10), Sigma=diag(10))
X1 <- mvrnorm(n=15, mu=runif(10), Sigma=diag(10))
p1.orig <- p2 <- diff <- rep(NA,10)
for(i in 1:10)
 {
    p1.orig[i] <- wilcox.test(X1[,i],X2[,i],
       alt='less', exact=FALSE)$p.value
    p2[i] <- wilcox.test(X1[,i],X2[,i],
       exact=FALSE)$p.value
    diff[i] <- mean(X2[,i]) - mean(X1[,i])
  }

## Convert two-sided p-values to one-sided
p1.new <- p2.p1(p2,diff)

## Compare with 'original' one-sided p-values
plot(p1.new,p1.orig); abline(0,1)
```

---

plotSigTree	*Function to plot the phylogenetic tree in R with branches (tips) colored according to significance of families of p-values (tips' p-values)*

Description

plotSigTree takes tree and unsorted.pvalues and computes p-values for each branch (family of tips) and colors the corresponding descendant branches. It computes the p-values based on arguments involving p-value adjustment (for multiple hypothesis testing) and either Hartung's, Stouffer's, or Fisher's p-value combination method. There are arguments that allow for the customization of the p-value cutoff ranges as well as the colors to be used in the coloring of the branches.

Usage

```
plotSigTree(tree, unsorted.pvalues, adjust=TRUE, side=1,
method="hommel", p.cutoffs=ifelse(rep(side==1, ifelse(side==1, 6, 3)),
c(.01, .05, .1, .9, .95, .99), c(.01, .05, .1)),
pal=ifelse(rep(side==1, ifelse(side==1, 1, length(p.cutoffs)+1)),
"RdBu", rev(brewer.pal(length(p.cutoffs)+1,"Reds"))),
test="Stouffer", branch.label=FALSE, tip.color=TRUE, edge.color=TRUE,
tip.label.size=1, branch.label.size=1,  type="fan",
use.edge.length=TRUE, edge.width=1, branch="edge",
root.edge=ifelse(type=="fan",FALSE,TRUE),
branch.label.frame="none")
```

Arguments

tree	a phylogenetic tree of class phylo.
unsorted.pvalues	
	a data frame (or matrix) with tip labels in column 1 and p-values in column 2. The tip labels must correspond to the tip labels in tree.
adjust	a logical argument that controls whether there is p-value adjustment performed (TRUE) or not (FALSE).
side	a numerical argument that takes values 1 and 2, depending on whether the p-values in unsorted.pvalues are 1-sided or 2-sided, respectively. Only used in p-value adjustment if adjust = TRUE.
method	one of the p-value adjustment methods (used for multiple-hypothesis testing) found in p.adjust.methods ("holm", "hochberg", "hommel", "BH", "bonferroni", "BY", "fdr", and "none"). See help for p.adjust for more details on these methods. method is only used if adjust = TRUE.
p.cutoffs	a vector of increasing p-value cutoffs (excluding 0 and 1) to determine the ranges of p-values used in the coloring of the branches.
pal	one of the palettes from the RColorBrewer package (see brewer.pal.info for a list) or a vector of hexadecimal colors (or other valid R colors). These are the colors that are used to color the branches. The first color corresponds to the first range in p.cutoffs and so on.

test	a character string taking on "Hartung", "Stouffer", or "Fisher". This is the p-value combination method that will be used. In most cases, "Stouffer" will be most appropriate, unless adonis.tree indicates significant evidence of dependence among p-values, in which case "Hartung" is preferred.
branch.label	a logical argument that controls whether the branches are labeled (TRUE) or not (FALSE). This results in either edges or nodes being labeled, depending on the branch argument. The edge and node labeling is internal to class phylo, and are not necessarily sequential numbers. When branch="edge" is used, the tip edges are not labeled (to avoid clutter). Branch labels match those returned by export.inherit.
tip.color	a logical argument that controls whether the tips are colored (TRUE) or not (FALSE).
edge.color	a logical argument that controls whether the edges are colored (TRUE) or not (FALSE).
tip.label.size	a numerical argument that controls the (cex) size of the text of the tip labels.
branch.label.size	
	a numerical argument that controls the (cex) size of the text of the branch labels (see branch.label argument).
type	a character string that controls which type of plot will be produced. Possible values are "phylogram", "cladogram", "fan", "unrooted", and "radial". See plot.phylo.
use.edge.length	
	a logical argument that uses the original edge lengths from tree (TRUE) or not (FALSE). This has no effect if tree does not have edge lengths defined to begin with. Can be affected by root.edge, depending on type (see root.edge below).
edge.width	a numeric vector controlling width of plotted edges. This is passed to (plot.phylo).
branch	a character controlling branch definition: "edge" and "node" are the only options. This does not affect statistical methods, only the colors used in edge coloring. Prior to package version 1.2, only branch="node" was implemented.
root.edge	a logical argument that controls whether the root edge is plotted (TRUE) or not (FALSE). Note that root.edge=TRUE forces use.edge.length=FALSE when type is "phylogram", "cladogram", "fan", or "unrooted".
branch.label.frame	
	a character controlling the frame around the branch labels (only used when branch.label=TRUE). Only options "none", "circ", and "rect" are supported.

Details

The tip labels of tree (accessed via tree$tip.label) must have the same names (and the same length) as the tip labels in unsorted.pvalues, but may be in a different order. The p-values in column 2 of unsorted.pvalues obviously must be in the [0, 1] range. p.cutoffs takes values in the (0, 1) range. The default value for p.cutoffs is c(0.01, 0.05, 0.1, 0.9, 0.95, 0.99) if side is 1 and c(0.01, 0.05, 0.1) if side is 2. Thus, the ranges (when side is 1) are: [0, .01], (.01, .05], ..., (.99, 1]. These ranges correspond to the colors specified in pal. P-values in the [0, .01] range correspond to the left-most color if pal is a palette (view this via display.brewer.pal(x,

pal) - where x is the number of colors to be used) or the first value in the vector if `pal` is a vector of colors. If `pal` is a vector of colors, then the length of `pal` should be one greater than the length of `p.cutoffs`. In other words, its length must be the same as the number of p-value ranges. An example of a color in hexadecimal format is "#B2182B". The default value of `pal` is "RdBu" (a divergent palette of reds and blues, with reds corresponding to small p-values) if `side` is 1 and the reverse of "Reds" (a sequential palette) if `side` is 2. The sequential palettes in `RColorBrewer` go from light to dark, so "Reds" is reversed so that the dark red corresponds to small p-values. It probably makes more sense to use a divergent palette when using 1-sided p-values and a sequential palette (reversed) when using 2-sided p-values. To create a vector of reversed colors from a palette with x number of colors and "PaletteName" as the name of the palette, use `rev(brewer.pal(x, "PaletteName"))`. `use.edge.length` may be useful to get a more uniformly-shaped tree. `plotSigTree` assumes that each internal node has exactly two descendants. It also assumes that each internal node has a lower number than each of its ancestors (excluding tips).

The `branch` argument controls whether edge coloring corresponds to the combined p-value of the tips below the edge ("edge") or of the tips below the edge's leading (away from the tips) node ("node"). Note that if `branch="node"` is used, then both edges leaving a node will necessarily be colored the same.

To access the tutorial document for this package (including this function), type in R: `vignette("SigTree")`

Value

This function produces a phylogenetic tree plot.

Note

Extensive discussion of methods developed for this package are available in Jones (2012). In that reference, (and prior to package version number 1.1), this `plotSigTree` function was named `plot.color`; the name change was made to resolve S3 class issues.

For purposes of acknowledgments, it is worth noting here that the plotting done by `plotSigTree` relies internally on tools of the ape package (Paradis et al., 2004 Bioinformatics 20:289-290). To accomodate edge-specific coloring (as with the `branch="edge"` option), some of these ape package tools were adapted and re-named in the `SigTree` package. Specifically, see `?plotphylo2` and `?circularplot2`.

Author(s)

John R. Stevens and Todd R. Jones

References

Stevens J.R., Jones T.R., Lefevre M., Ganesan B., and Weimer B.C. (2017) "SigTree: A Microbial Community Analysis Tool to Identify and Visualize Significantly Responsive Branches in a Phylogenetic Tree." Computational and Structural Biotechnology Journal 15:372-378.

Jones T.R. (2012) "SigTree: An Automated Meta-Analytic Approach to Find Significant Branches in a Phylogenetic Tree" (2012). MS Thesis, Utah State University, Department of Mathematics and Statistics. http://digitalcommons.usu.edu/etd/1314

Examples

```
### To access the tutorial document for this package, type in R (not run here):
# vignette('SigTree')

### Create tree, then data frame, then use plotSigTree to plot the tree
### Code for random tree and data frame
node.size <- 10
seed <- 109
# Create tree
set.seed(seed);
library(ape)
r.tree <- rtree(node.size)
# Create p-values data frame
set.seed(seed)
r.pval <- rbeta(node.size, .1, .1)
# Randomize the order of the tip labels
# (just to emphasize that labels need not be sorted)
set.seed(seed)
r.tip.label <- sample(r.tree$tip.label, size=length(r.tree$tip.label))
r.pvalues <- data.frame(label=r.tip.label, pval=r.pval)

# Check for dependence among p-values; lack of significance here
# indicates default test="Stouffer" is appropriate;
# otherwise, test="Hartung" would be more appropriate.
adonis.tree(r.tree,r.pvalues)

# Plot tree in default 'fan' type, with branches labeled
plotSigTree(r.tree, r.pvalues, edge.width=4, branch.label=TRUE)

# Plot tree in 'phylogram' type, with branch labels circled
plotSigTree(r.tree, r.pvalues, edge.width=4, branch.label=TRUE,
  type='phylo', branch.label.frame='circ')

# Plot tree in 'phylogram' type, with branch labels circled,
# and assuming original p-values were for 2-sided test
plotSigTree(r.tree, r.pvalues, edge.width=4, branch.label=TRUE,
  type='phylo', branch.label.frame='circ', side=2)

# Plot tree in 'phylogram' type, with branch labels boxed;
# also give custom significance thresholds, and use
# a Purple-Orange palette (dark purple for low p-vals
# to dark orange for high p-vals)
plotSigTree(r.tree, r.pvalues, edge.width=4, branch.label=TRUE,
  type='phylo', branch.label.frame='rect',
  p.cutoffs=c(.01,.025,.975,.99), pal='PuOr')
```

SigTree.other   *SigTree Other Functions*

Description

Internal functions used by the main functions of SigTree (plotSigTree, export.figtree, and export.inherit):

num.edges	determine the number of edges in tree
num.tips	determine the number of tips in tree
num.internal.nodes	determine the number of nodes in tree
num.total.nodes	determine the number of total nodes (internal + tips) in tree
srt.pvalues	sort unsorted.pvalues by tip labels (column 1) to be in same order as tip labels in tree
stouffers	perform Stouffer's Method on a vector of p-values; return one p-value
fishers	perform Fisher's Method on a vector of p-values; return one p-value
index.matrix	create matrix to identify the descendants/tips (rows) belonging to each node/family (column)
p.p2.ADJ.p1	convert 1-sided p-values to 2-sided, perform p-value adjustment (for multiple-hypothesis testing), and convert back to 1-sided
result	calculate p-values for each node/edge branch
tip.colors	determine coloring of each tip
edge.colors	determine coloring of each edge
plotphylo2	(based on ape package's plot.phylo function); plots tree while allowing for different edge coloring (root edge when type="fan", and different colors for each half of the "perpendicular-to-the-root" edges). Prior to package version 1.2, plot.phylo was used instead. Beginning in package version 1.3 (to attain CRAN compatibility), includes .C calls to copies of four ape .C functions (copied with credit under ape's GPL license).
circularplot2	(based on ape package's circular.plot function) called by phyloplot2 when type="fan"
hartung	perform Hartung's Method on a vector of p-values; return one p-value

Details

It is assumed that each internal node has exactly two descendants. It is also assumed that each internal node has a lower number than each of its ancestors (excluding tips).

To access the tutorial document for this package (including this function), type in R: vignette("SigTree")

Note

Extensive discussion of methods developed for this package are available in Jones (2012). In that reference, (and prior to package version number 1.1), the srt.pvalues function was named sort.pvalues (the name change was made to resolve S3 class issues), and plotphylo2 was not available.

Author(s)

John R. Stevens and Todd R. Jones

References

Stevens J.R., Jones T.R., Lefevre M., Ganesan B., and Weimer B.C. (2017) "SigTree: A Microbial Community Analysis Tool to Identify and Visualize Significantly Responsive Branches in a

Phylogenetic Tree." Computational and Structural Biotechnology Journal 15:372-378.

Jones T.R. (2012) "SigTree: An Automated Meta-Analytic Approach to Find Significant Branches in a Phylogenetic Tree" MS Thesis, Utah State University, Department of Mathematics and Statistics. http://digitalcommons.usu.edu/etd/1314

Index