Package 'phytools'

May 31, 2013

Version 0.2-80

Date 2013-5-31

Title Phylogenetic Tools for comparative biology (and other things)

Author Liam J. Revell

Maintainer Liam J. Revell <liam.revell@umb.edu>

Depends R (>= 2.6), ape (>= 3.0-7), mnormt, rgl

Imports calibrate, msm, numDeriv, phangorn (>= 1.6-3), plotrix

Suggests animation

Enhances extrafont

ZipData no

Description phytools provides various functions for phylogenetic
analysis, mostly relevant to comparative biology.

License GPL (>= 2)

URL http://www.phytools.org

Repository CRAN

Date/Publication 2013-05-31 15:21:47

NeedsCompilation no

R topics documented:

1

Index **[110](#)**

---

phytools-package *phytools: Phylogenetic Tools for comparative biology (and other things)*

Description

phytools provides functions for phylogenetic comparative biology; as well as several other functions for tree inference, manipulation, and analysis that are not implemented in other R packages.

The complete list of functions can be displayed with library(help = phytools).

More information on **phytools** can be found at <http://www.phytools.org> or http://blog.phytools.org.

Author(s)

Liam J. Revell

Maintainer: Liam J. Revell <liam.revell@umb.edu>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). Methods Ecol. Evol. *3*, 217-223. doi:10.1111/j.2041-210X.2011.00169.x

---

add.color.bar *Add color bar to a plot*

Description

This function adds a color bar to a plot created by [plotBranchbyTrait](#). The only way to add the color bar at present is by clicking on the desired location within your plot. It is also used internally by contMap(...,type="fan") and densityMap(...,type="fan").

Usage

```
add.color.bar(leg, cols, title=NULL, lims=c(0,1), digits=1, prompt=TRUE, ...)
```

Arguments

`leg`	numerical value for the length of the legend.
`cols`	colors for the legend.
`title`	text to plot above the bar.
`lims`	range for the bar.
`digits`	digits for plotted numbers.
`prompt`	logical value indicating whether the location of the legend should be obtained interactively.
`...`	optional arguments including: x x-coordinate of the legend (if `prompt=FALSE`); y y-coordinate of the legend.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[contMap](), [densityMap](), [plotBranchbyTrait]()

---

add.everywhere   *Add tip to all edges in a tree*

Description

This function adds a tip to all branches to the tree and returns a a list of trees as an object of class `"multiPhylo"`.

Usage

```
add.everywhere(tree, tip.name)
```

Arguments

`tree`	an object of class `"phylo"`.
`tip.name`	a string containing the name of the tip to add.

Value

A list of trees as an object of class `"multiPhylo"`. Since the tip can be added to any branch, the length of the list is equal to the number of edges in `tree`.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[allFurcTrees](), [exhaustiveMP]()

---

add.random   *Add tips at random to the tree*

Description

This function adds new tips at random to a tree with branch lengths. If no edge lengths are provided, and the tree is ultrametric, then edge lengths are assigned to keep the tree ultrametric. The probability that at new tip is added along any branch is directly proportional to the length of the branch.

Usage

```
add.random(tree, n=NULL, tips=NULL, edge.length=NULL, order=c("random","input"))
```

Arguments

tree	an object of class `"phylo"`.
n	a number of tips to add to the tree. If `NULL`, will use `length(tips)`.
tips	a set of tip names for the added tips. If `NULL`, will name as `paste("t",length(tree$tip)+1:n,sep="")`.
edge.length	terminal edge length for the added tips. If `NULL`, and `is.ultrametric(tree)==TRUE`, then edge lengths will be assigned to keep the tree ultrametric. Note that if edge lengths are assigned and n>1, then the asssigned terminal edge lengths are not guaranteed as subsequent random tip addition could occur along the new terminal edge.
order	addition order for the new tips.

Details

Note that sometimes the resultant tree plotted with [plot.phylo](#) or [plotSimmap](#) may display with branches crossing. If so, the tree can be 'untangled' using [untangle](#).

Value

An object of class `"phylo"`.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[allFurcTrees](#), [add.everywhere](#)

---

allFurcTrees   *Generate all bi- and multifurcating unrooted trees*

Description

This function creates all possible unrooted bi- and multifurcating trees and returns a list of trees as an object of class ″multiPhylo″.

Usage

```
allFurcTrees(n, tip.label=NULL, to.plot=TRUE)
```

Arguments

n   an integer giving the desired number of species.

tip.label   an optional vector of length n containing the tip names.

to.plot   an optional logical value indicating whether or not to plot the trees.

Details

This function should be used with caution for n greater than about 8, as in this case the number of possible trees is extremely large.

Value

A list of trees as an object of class ″multiPhylo″.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Felsenstein, J. 2004. *Inferring Phylogenies*. Sinauer.

See Also

[add.everywhere](#), [exhaustiveMP](#)

---

anc.Bayes   *Bayesian ancestral character estimation*

Description

This function uses Bayesian MCMC to sample from the posterior distribution for the states at internal nodes in the tree.

Usage

```
anc.Bayes(tree, x, ngen=10000, control=list())
```

Arguments

tree	an object of class "phylo".
x	a vector of tip values for species; names(x) should be the species names.
ngen	a integer indicating the number of generations for the MCMC.
control	a list of control parameters containing the following elements: sig2: starting value for σ^2 (BM rate); a: starting for the state at the root node; y: starting values for the states at all internal nodes excluding the root (should be labeled with node numbers); pr.mean: means for the prior distributions in the following order - sig2, a, y, note that the prior probability distribution is exponential for sig2 and normal for a and y; pr.var: variances on the prior distributions, same order as pr.mean (but the variance is not used for sig2); prop: variances on the normal proposal distributions in the same order as pr.mean; sample: sample frequency from the MCMC.

Value

A matrix with number of rows ngen/sample+1 containing the posterior sample and likelihoods. Matrix columns are labeled either sig2 or by the node number of the internal node.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[ace](), [anc.ML](), [anc.trend](), [evol.rate.mcmc](), [fastAnc]()

Examples

```
tree<-rtree(50)
x<-fastBM(tree,sig2=2) # simulate using fastBM
X<-anc.Bayes(tree,x,ngen=10000) # sample ancestral states
estimates<-colMeans(X[21:nrow(X),]) # get estimates, excluding burnin
```

---

anc.ML   *Ancestral character estimation using likelihood*

Description

This function estimates the evolutionary parameters and ancestral states for Brownian evolution using likelihood.

Usage

```
anc.ML(tree, x, maxit=2000)
```

Arguments

tree	an object of class "phylo".
x	a vector of tip values for species; names(x) should be the species names.
maxit	an optional integer value indicating the maximum number of iterations for optimization.

Details

Because this function relies on a high dimensional numerical optimization of the likelihood function, `fastAnc` should probably be preferred for most purposes. If using `anc.ML`, users should be cautious to ensure convergence. This has been ameliorated in phytools>=0.2-48 by seeding the ML optimization with the result from `fastAnc`.

Value

A list with the following components:

sig2	the variance of the BM process.
ace	a vector with the ancestral states.
logLik	the log-likelihood.
convergence	the value of $convergence returned by optim() (0 is good).

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

ace, anc.Bayes, fastAnc, optim

Examples

```
tree<-rtree(50)
x<-fastBM(tree) # simulate using fastBM
anc.ML(tree,x) # fit model & estimate ancestral states
```

---

anc.trend   *Ancestral character estimation with a trend*

Description

This function estimates the evolutionary parameters and ancestral states for Brownian evolution
with directional trend.

Usage

```
anc.trend(tree, x, maxit=2000)
```

Arguments

tree   an object of class `"phylo"`.

x   a vector of tip values for species; `names(x)` should be the species names.

maxit   an optional integer value indicating the maximum number of iterations for opti-
                mization.

Details

Note that this will generally only work and produce sensible results for a phylogeny with some non-
contemporary tips (i.e., a tree with some fossil species). The function uses [optim](#) with `method="L-BFGS-B"`;
however optimization is only constrained for the `sig2` which must be >0.

Value

A list with the following components:

ace   a vector with the ancestral states.

mu   a trend parameter per unit time.

sig2   the variance of the BM process.

logL   the log-likelihood.

convergence   the value of $convergence returned by `optim()` (0 is good).

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[ace](#), [anc.Bayes](#), [anc.ML](#), [optim](#)

Examples

```
tree<-rtree(30)
x<-fastBM(tree,mu=2) # simulate using fastBM with a trend (m!=0)
anc.trend(tree,x) # fit model & estimate ancestral states
```

---

ancThresh	*Ancestral character estimation under the threshold model using Bayesian MCMC*

Description

This function uses Bayesian MCMC to estimate ancestral states and thresholds for a discrete character under the threshold model from quantitative genetics (Felsenstein 2012).

Usage

```
ancThresh(tree, x, ngen=1000, sequence=NULL, method="mcmc", control=list(), ...)
```

Arguments

tree	phylogenetic tree.
x	a named vector containing discrete character states; or a matrix containing the tip species, in rows, and probabilities of being in each state, in columns.
ngen	number of generations to run the MCMC.
sequence	assumed ordering of the discrete character state. If not supplied and x is a vector then numerical/alphabetical order is assumed; if not supplied and x is a matrix, then the column order of x is used.
method	only method currently available is "mcmc".
control	list containing the following elements: sample, the sampling interval; propliab variance of the proposal distribution for liabilities; propthresh variance on the proposal distribution for the thresholds; pr.anc prior probability distribution on the ancestral states for each node, in a matrix - not all nodes need to be supplied; pr.th prior density on the thresholds; burnin number of generations to exclude for burn-in when plotting posterior probabilities on the tree; plot logical value indicating whether or not to plot the posterior probabilities; print logical value indicating whether or not to print the state of the MCMC; piecol colors for the posterior probabilities plotted as pie charts at internal nodes; and tipcol which indicates whether the tip colors should be based on the input data ("input") or sampled tip liabilities ("estimated"). These will only differ if there is uncertainty in the tip states.
...	additional arguments to be passed to [plotThresh](called internally).

Value

This function returns as list with four elements:

ace	posterior probabilities for each character state at each internal node, with the burn-in excluded.
mcmc	full posterior sample for the states.
par	full posterior sample for the thresholds and likelihood.
liab	full posterior sample of the liabilities at internal and tip nodes.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Felsenstein, J. (2012) A comparative method for both discrete and continuous characters using the threshold model. *American Naturalist*, **179**, 145-156.

See Also

anc.Bayes, threshBayes

---

anoletree   *Phylogeny of 100 Greater Antillean anole species with mapped discrete character*

Description

A phylogeny of 100 Greater Antillean anole species with a mapped discrete character - 'ecomorph class.' Data and tree are from Mahler et al. (2010).

Usage

```
data(anoletree)
```

Format

The data are stored as a modified object of class ″phylo″ with a mapped discrete character. (E.g., see read.simmap.)

Source

Mahler, D. L, L. J. Revell, R. E. Glor, and J. B. Losos. (2010) Ecological opportunity and the rate of morphological evolution in the diversification of Greater Antillean anoles. *Evolution*, **64**, 2731-2745.

---

applyBranchLengths   *Applies the branch lengths of a reference tree to a target*

Description

This function applies the set of branch lengths from a reference tree to a target tree while reconciling any mappings (as in [read.simmap](#)) with the new branch lengths.

Usage

```
applyBranchLengths(tree, edge.length)
```

Arguments

tree	target tree.
edge.length	number of digits for rounding. Passed to [round](#).

Value

A tree with branch lengths, or modified ″phylo″ object with a mapped discrete character.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

ave.rates   *Average the posterior rates*

Description

Primarily used internally by [posterior.evolrate](#).

Usage

```
ave.rates(tree, shift, tips, sig1, sig2, ave.shift, showTree=TRUE)
```

Arguments

tree	a tree.
shift	the shift point for this sample.
tips	tip names tipward of shift.
sig1	rate 1.
sig2	rate 2.
ave.shift	average shift from all samples.
showTree	logical value indicating whether to plot the rate-stretched tree.

Value

A list of the rates.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[evol.rate.mcmc](#), [minSplit](#), [posterior.evolrate](#)

---

bind.tip   *Attaches a new tip to a tree*

Description

Functions adds a new tip to the tree. If the tree is ultrametric and no branch length is specified, then edge.length is scaled so that the tree remains ultrametric after the new tip is added.

Usage

```
bind.tip(tree, tip.label, edge.length=NULL, where=NULL, position=0)
```

Arguments

tree	receptor tree.
tip.label	a string containing the species name for the new tip.
edge.length	edge length for the new tip (a scalar).
where	node number to attach new tip. If position>0 then then tip will be attached *below* the specified node. Node numbers can also be tips, in which case the new tip will be added along the terminal edge. To find out the tip number for given species with name *"species"* type: which(tree$tip.label=="species".
position	distance *below* node to add tip.

Details

Wrapper function for 'ape' [bind.tree](#).

Value

A tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

`bmPlot`	*Simulates and visualizes discrete-time Brownian evolution on a phylogeny*

Description

This function conducts discrete-time Brownian motion simulation on an input tree, plots the outcome, and returns the tip and internal node states to the user as a named vector. The function will first rescale and round the branch lengths to integer length, if they are not already in integer values. If integer branch lengths are provided, the user should also set ngen=max(nodeHeights(tree)). For type="threshold" the visualization is of the threshold model (Felsenstein 2012), in which the evolving character is liability and the segements of evolution are colored by their value for the threshold trait. If type="threshold" is used, the function requires at least one addition input: thresholds, a vector containing the ordered thresholds between states. The user can also provide the colors for plotting in cols. Note that one more color than threshold should be provided as one threshold implies two states; two thresholds, three states; etc. If no value for cols is provided, the function will recycle a set of four colors up to the number of times required by thresholds.

Usage

```
bmPlot(tree, type="BM", anc=0, sig2=1/1000, ngen=1000, ...)
```

Arguments

`tree`	a phylogenetic tree in "phylo" format.
`type`	the type of plot to create. See Description.
`anc`	the ancestral value for the root node.
`sig2`	the BM rate (variance of the Brownian evolution process).
`ngen`	number of generations for the simulation: will rescale the tree to this total length.
`...`	arguments to be passed to different methods.

Value

This function conducts and plots discrete time Brownian simulation and returns a vector containing the simulated states at internal nodes and tips of the tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Felsenstein, J. 2012. A comparative method for both discrete and continuous characters using the threshold model. *American Naturalist*, **179**, 145-156.

See Also

fastBM, pbtree, phenogram, threshBayes

Examples

```
# plot BM simulation on 20 taxon tree
tree<-pbtree(n=20)
x<-bmPlot(tree)

# plot simulation of a threshold character
tree<-pbtree(n=20)
x<-bmPlot(tree,type="threshold",thresholds=c(0,1,2))
```

---

branching.diffusion   *Animation of branching random diffusion*

Description

This function creates an animation of branching random diffusion (i.e., BM with speciation).

Usage

```
branching.diffusion(sig2=1, b=0.0023, time.stop=1000, ylim=NULL,
smooth=TRUE, pause=0.02, record=NULL, path=NULL)
```

Arguments

sig2	variance of BM process.
b	birthrate for branching process.
time.stop	number of generations to run.
ylim	y limits (for plotting).
smooth	if TRUE, smoother plotting (but slower).
pause	pause (in s) between generations.
record	filename for video file output (no video if NULL).
path	path to file for video rendering (by default is C:/Program Files/ffmpeg/bin/ffmpeg.exe).

Value

An animated plot and (optionally) a recorded video file.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

bmPlot, fastBM

---

brownie.lite   *Likelihood test for rate variation in a continuous trait*

Description

This function takes a modified "phylo" object with a mapped binary or multistate trait (see read.simmap) and data for a single continuously valued character. It then fits the Brownian rate variation ("non-censored") model of O'Meara et al. (2006; *Evolution*). This is also the basic model implemented in Brian O'Meara's "Brownie" program.

Usage

```
brownie.lite(tree, x, maxit=2000, test="chisq", nsim=100, se=NULL)
```

Arguments

tree	a phylogenetic tree in modified "phylo" format (see read.simmap, make.simmap, or paintSubTree).
x	a vector of tip values for species; names(x) should be the species names.
maxit	an optional integer value indicating the maximum number of iterations for optimization - may need to be increased for large trees.
test	an optional string indicating the method for hypothesis testing - options are "chisq" or "simulation".
nsim	number of simulations (only used if test="simulation").
se	a vector containing the standard errors for each estimated mean in x.

Details

Sampling error in the estimation of species means can also be accounted for by assigning the vector se with the species specific sampling errors for x.

Value

A list with the following components:

sig2.single	is the rate for a single rate model - this is usually the "null" model.
a.single	is the estimated state at the root node for the single rate model.
var.single	variance on the single rate estimator - obtained from the Hessian.
logL1	log-likelihood of the single-rate model.
k1	number of parameters in the single rate model (always 2).
sig2.multiple	is a length *p* (for *p* rates) vector of BM rates from the multi-rate model.
a.multiple	is the estimated state at the root node for the multi-rate model.
var.multiple	*p* x *p* variance-covariance matrix for the *p* rates - the square-roots of the diagonals should give the standard error for each rate.

logL.multiple	log-likelihood of the multi-rate model.
k2	number of parameters in the multi-rate model (*p*+1).
P.chisq	P-value for a likelihood ratio test against the χ^2 distribution; or
P.sim	P-value for a likelihood ratio test agains a simulated null distribution.
convergence	logical value indicating if the likelihood optimization converged.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

O'Meara, B. C., C. Ane, M. J. Sanderson, and P. C. Wainwright. (2006) Testing for different rates of continuous trait evolution using likelihood. *Evolution*, **60**, 922–933.

See Also

brownieREML, evol.vcv

---

brownieREML   *REML version of brownie.lite*

Description

This function takes a modified "phylo" object with a mapped binary or multistate trait (see read.simmap) and data for a single continuously valued character. It then uses restricted maximum likelihood (REML) to fit the Brownian rate variation ("noncensored") model of O'Meara et al. (2006; *Evolution*). This function is similar to brownie.lite but uses REML (which is faster and unbiased) instead of ML. REML optimization takes advantage of Felsenstein's (1985) contrasts algorithm.

Usage

```
brownieREML(tree, x, maxit=2000)
```

Arguments

tree	a phylogenetic tree in modified "phylo" format (see read.simmap and make.simmap).
x	a vector of tip values for species; names(x) should be the species names.
maxit	an optional integer value indicating the maximum number of iterations for optimization - may need to be increased for large trees.

Value

A list with the following components:

sig2.single   is the rate for a single rate model - this is usually the "null" model.

logL1   log-likelihood of the single-rate model.

sig2.multiple   is a length *p* (for *p* rates) vector of BM rates from the multi-rate model.

logL2   log-likelihood of the multi-rate model.

convergence   numerical value from optim.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Felsenstein, J. 1985. Phylogenies and the comparative method. *American Naturalist*, **125**, 1–15.

O'Meara, B. C., C. Ane, M. J. Sanderson, and P. C. Wainwright. 2006. Testing for different rates of continuous trait evolution using likelihood. *Evolution*, **60**, 922–933.

See Also

brownie.lite, evol.vcv, evol.rate.mcmc

---

contMap   *Map continuous trait evolution on the tree*

Description

Function plots a tree with a mapped continuous character. The mapping is accomplished by estimating states at internal nodes using ML with fastAnc, and the interpolating the states along each edge using equation [2] of Felsenstein (1985).

Usage

```
contMap(tree, x, res=100, fsize=NULL, ftype=NULL, lwd=4, legend=NULL,
lims=NULL, outline=TRUE, sig=3, type="phylogram", ...)
## S3 method for class 'contMap'
plot(x, ...)
```

Arguments

tree	object of class "phylo".
x	vector of phenotypic trait values for species. names(x) should contain the species names and match tree$tip.label. Or, for plot.contMap, an object of class "contMap".
res	resolution for gradient plotting. Larger numbers indicate a finer (smoother) gradient.
fsize	relative font size - can be a vector with the second element giving the font size for the legend.
ftype	font type - see options in [plotSimmap](). As with fsize, this can be a vector with the second element giving font type for the legend.
lwd	line width for branches.
legend	if FALSE no legend is plotted; if a numeric value, it gives the length of the legend in units of branch length. Default is 0.5 times the total tree length.
lims	range for the color map. By default, this will be c(min(x),max(x)), and should always include this range.
outline	logical value indicating whether or not to outline the branches of the tree in black.
sig	the number of decimal places to show on the legend limits.
type	type of plot desired. Options are "phylogram" for a rightward square phylogram; and "fan" for a circular phylogram.
...	optional arguments for plot.contMap which include all the arguments of contMap except for tree, x, res, and lims.

Value

Plots a tree. An object of class "contMap" is returned invisibly.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. *In press.* Two new graphical methods for mapping trait evolution on phylogenies. *Methods in Ecology and Evolution.*

Felsenstein, J. 1985. Phylogenies and the comparative method. *American Naturalist*, **125**, 1-15.

See Also

[densityMap](), [fastAnc](), [plotSimmap]()

---

countSimmap | *Counts the number of character changes on a SIMMAP style tree or set of trees*

Description

This function takes a tree or a set of trees with a mapped discrete character (SIMMAP style, e.g., see `make.simmap` or `read.simmap`), and computes the total number of character changes as well as the number of character changes between all states.

Usage

```
countSimmap(tree, states=NULL, message=TRUE)
```

Arguments

tree
: a single tree or a set of trees with a mapped discrete character (e.g, see `make.simmap` or `read.simmap`.

states
: optional argument with the states for the mapped character. If not provided, these will be computed from the tree. This is useful if averaging across many trees, some of which may lack certain states.

message
: optional logical argument indicating whether or not to return an informative message about the function output.

Value

A list with up to three elements: `N` is an integer value giving the total number of character changes on the tree; `Tr` gives the number of of transitions between row and column states (or a matrix containing both `N` and the transitions between states, in rows, for an object of class `"multiPhylo"`); and (optionally) `message` contains an explanatory message about the function output.

Author(s)

Liam Revell <liam.revell@umb.edu>

Examples

```
tree<-pbtree(n=100,scale=1)
Q<-matrix(c(-2,1,1,1,-2,1,1,1,-2),3,3)
colnames(Q)<-rownames(Q)<-c("A","B","C")
mtree<-sim.history(tree,Q)
countSimmap(mtree,states=rownames(Q))
```

---

densityMap   *Plot posterior density of stochastic mapping on a tree*

Description

Function plots a tree with the posterior density for a mapped character from stochastic character
mapping on the tree. Since the mapped value is the probability of being in state "1", only binary
[0,1] characters are allowed.

Usage

```
densityMap(trees, res=100, fsize=NULL, ftype=NULL, lwd=3, check=FALSE,
legend=NULL, outline=FALSE, type="phylogram", ...)
## S3 method for class 'densityMap'
plot(x, ...)
```

Arguments

trees	set of phylogenetic trees in a modified "multiPhylo" object. Values for a discrete character are mapped on the tree. See `make.simmap` and `read.simmap` for details.
res	resolution for gradient plotting. Larger numbers indicate a finer (smoother) gradient.
fsize	relative font size - can be a vector with the second element giving the font size for the legend.
ftype	font type - see options in `plotSimmap`. As with `fsize`, can be a vector with the second element giving font type for the legend.
lwd	line width for branches.
check	check to make sure that the topology and branch lengths of all phylogenies in `trees` are equal.
legend	if FALSE no legend is plotted; if a numeric value, it gives the length of the legend in units of branch length. Default is 0.5 times the total tree length.
outline	logical value indicating whether or not to outline the branches of the tree in black.
type	type of plot desired. Options are "phylogram" for a rightward square phylogram; and "fan" for a circular phylogram.
x	for plot.densityMap, an object of class "densityMap".
...	optional arguments for plot.densityMap. These include all the arguments of densityMap except trees and res.

Value

Plots a tree and returns an object of class "densityMap" invisibly.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. *In press.* Two new graphical methods for mapping trait evolution on phylogenies. *Methods in Ecology and Evolution.*

Huelsenbeck, J. P., R. Neilsen, and J. P. Bollback. 2003. Stochastic mapping of morphological characters. *Systematic Biology*, **52**, 131-138.

Bollback, J. P. 2006. Stochastic character mapping of discrete traits on phylogenies. *BMC Bioinformatics*, **7**, 88.

See Also

make.simmap, plotSimmap, read.simmap

---

describe.simmap *Summarizes a stochastic mapped tree or set of trees*

Description

This function summarizes the result of one or more stochastic maps.

Usage

```
describe.simmap(tree, ...)
```

Arguments

tree   a single tree or a set of trees with a mapped discrete character (e.g, see make.simmap or read.simmap.

...   optional arguments which include: plot, a logical value indicating whether or not to plot the posterior probabilities at nodes (default is plot=FALSE); check.equal, a logical value indicating whether or not to check if all trees are equal using all.equal.phylo (default is check.equal=FALSE); and message, a logical indicating whether or not to print an informative message to the screen (default is message=TRUE).

Value

A list with the following elements is returned (invisibly if message=TRUE):

count   a matrix containing the number and types of transitions for each tree, if class(tree)="multiPhylo".

times   a matrix containg the times spend in each state on each tree.

ace   the posterior probabilities of each node being in each state, if class(tree)="multiPhylo".

legend   a vector containing the plot legend, if plot=TRUE.

if class(tree)="phylo" then the function simply returns the results of [countSimmap](#) combined with the states at each node of the tree and a matrix containing the total and relative times spent in each state on the tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

drop.clade   *Drop a clade from a tree*

Description

Mostly internal function for [posterior.evolrate](#); function drops the clade containing the species in tip.

Usage

```
drop.clade(tree, tip)
```

Arguments

tree   object of class "phylo".

tip   set of tips in a clade.

Details

Probably should not use unless you know what you're doing.

Value

An object of class "phylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

---

drop.tip.simmap   *Drop tips or extract clade from tree with mapped discrete character*

Description

This function drops one or multiple tips from the modified ″phylo″ object with a mapped binary or multistate trait (see read.simmap) while maintaining the matrix $mapped.edge and list of mappings by branch maps. This function is equivalent to drop.tip but for a tree with a mapped discrete character.

extract.clade.simmap is functionally equivalent to extract.clade but preserves discrete character mappings on the tree.

Usage

```
drop.tip.simmap(tree, tip)
extract.clade.simmap(tree, node)
```

Arguments

tree   a modified object of class ″phylo″ (see read.simmap).

tip   name or names of species to be dropped.

node   node number for the root node of the clade to be extracted.

Value

A modified object of class ″phylo″ containing the elements maps and $mapped.edge with the time spent in each state along each edge of the tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

brownie.lite, drop.tip, extract.clade, make.simmap, read.simmap, sim.history

---

estDiversity   *Estimate diversity at each node of the tree*

Description

This function estimates the lineage density at each node in the tree based on a biogeographic model (based on Mahler et al. 2010).

Usage

```
estDiversity(tree, x, method=c("asr","simulation"), model="ER", ...)
```

Arguments

tree	is a phylogenetic tree in "phylo" format.
x	a vector containing the biogeographic area for each of the tip taxa.
method	method for reconstructing ancestral biogeography.
model	model for ancestral character estimation. In theory, any model from [ace](ace); however only symmetric models permitted for method="asr".
...	optional arguments. So far, this includes only nsim, the number of stochastic mappings to conduct using make.simmap for method="simulation".

Details

Two different methods are implemented in the current version. For method="asr" the state at the current node, and at each position along each co-extant internal edge, is computed as the marginal (empirical Bayesian) ancestral state reconstruction using the re-rooting method of Yang (2006). The lineage density is then computed as the sum of the marginal reconstructions (posterior probabilities) times the summed marginal ancestral reconstructions across co-extant edges. In method="simulation", stochastic character mapping is used to generate optional argument nsim stochastic maps of ancestral biogeography. Then the lineage density at each node is computed as the number of co-existing lineages with the same biogeography as the focal node, averaged acrossed stochastic maps. The importance of this distinction may depend on the degree to which reconstructions at internal nodes are independent, which relates to the distinction between marginal and joint reconstruction (e.g., see Yang 2006).

Value

A vector containing the estimated lineage density at each node

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Mahler, D. L, L. J. Revell, R. E. Glor, and J. B. Losos. (2010) Ecological opportunity and the rate of morphological evolution in the diversification of Greater Antillean anoles. *Evolution*, **64**, 2731-2745.

Yang, Z. (2006) *Computational Molecular Evolution*. Oxford University Press.

See Also

fitDiversityModel

---

evol.rate.mcmc	*Bayesian MCMC method for identifying exceptional phenotypic diversification in a phylogeny*

Description

This function takes a phylogenetic tree and data for a single continuously valued character and uses a Bayesian MCMC approach to identify the phylogenetic location of a shift in the evolutionary rate through time.

Usage

```
evol.rate.mcmc(tree, x, ngen=10000, control=list())
```

Arguments

tree	a phylogenetic tree in "phylo" format.
x	a vector of tip values for species; names(x) should be the species names.
ngen	an optional integer value indicating the number of generations for the MCMC.
control	a list of control parameters containing the following elements: sig1: starting value for $\sigma(1)^2$; sig2: starting value for $\sigma(2)^2$; a: starting value for a; sd1: standard deviation for the normal proposal distribution for $\sigma(1)^2$; sd2: standard deviation for the normal proposal distribution for $\sigma(2)^2$; kloc: scaling parameter for tree move proposals - $1/\lambda$ for the reflected exponential distribution; sdlnr: standard deviation on the log-normal prior on $\sigma(1)^2/\sigma(2)^2$; rand.shift: probability of proposing a random shift in the tree (improves mixing); print: print frequency for the MCMC; sample: sample frequency.

Details

Default values of control are given in Revell et al. (2012).

Value

A list with the following components:

mcmc   results from the MCMC run.

tips   list of stips in rate $\sigma(1)^2$ for each sampled generation of MCMC (to polarize the
                rate shift).

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J., D. L. Mahler, P. Peres-Neto, and B. D. Redelings. (2012) A new method for identifying
exceptional phenotypic diversification. *Evolution*, **66**, 135-146.

See Also

[anc.Bayes](), [brownie.lite](), [evol.vcv](), [minSplit](), [posterior.evolrate]()

---

evol.vcv   *Likelihood test for variation in the evolutionary VCV matrix*

Description

This function takes a modified "phylo" object with a mapped binary or multistate trait and data
for an arbitrary number of continuously valued character. It then fits the multiple evolutionary
variance-covariance matrix (rate matrix) model of Revell & Collar (2009; *Evolution*).

Usage

```
evol.vcv(tree, X, maxit=2000, vars=FALSE)
```

Arguments

tree   a phylogenetic tree in modified "phylo" format (see [read.simmap]()).

X   an n x m matrix of tip values for m continuously valued traits in n species - row
                names should be species names.

maxit   an optional integer value indicating the maximum number of iterations for opti-
                mization - may need to be increased for large trees.

vars   an optional logical value indicating whether or not to estimate the variances of
                the parameter estimates from the Hessian matrix.

Details

This function performs optimization by first optimizing the likelihood with respect to the Cholesky matrices using optim. Optimization is by method="Nelder-Mead". Using box constraints does not make sense here as they would be applied to the Cholesky matrix rather than the target parameters. May have to increase maxit for large trees and more than 2 traits.

Value

A list with the following components:

R.single	vcv matrix for the single rate matrix model.
vars.single	optionally, a matrix containing the variances of the elements of R.single.
logL1	log-likelihood for single matrix model.
k1	number of parameters in the single marix model.
R.multiple	m x m x p array containing the p estimated vcv matrices for the p regimes painted on the tree.
vars.multiple	optionally, an array containing the variances of the parameter estimates in R.multiple.
logL.multiple	log-likelihood of the multi-matrix model.
k2	number of parameters estimated in this model.
P.chisq	P-value of the χ^2 test on the likelihood ratio.
convergence	logical value indicating whether or not the optimization has converged.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J., and D. C. Collar (2009) Phylogenetic analysis of the evolutionary correlation using likelihood. *Evolution*, **63**, 1090-1100.

See Also

evol.rate.mcmc, brownie.lite

---

evolvcv.lite	*Likelihood test for a shift in the evolutionary correlation between traits*

Description

This function takes a modified "phylo" object with a mapped binary or multistate trait and data for two and only two continuously valued character. It then fits four different evolutionary models: common rates and correlation; different rates, common correlation; different correlations, common rates; no common structure.

Usage

```
evolvcv.lite(tree, X, maxit=2000, tol=1e-10)
```

Arguments

tree	a phylogenetic tree in modified "phylo" format (see [read.simmap](#)).
X	an n x m matrix of tip values for m continuously valued traits in n species - row names should be species names.
maxit	an optional integer value indicating the maximum number of iterations for optimization - may need to be increased for large trees.
tol	tolerance value for "L-BFGS-B" optimization.

Value

A list with the results summarized for each model.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J., and D. C. Collar (2009) Phylogenetic analysis of the evolutionary correlation using likelihood. *Evolution*, **63**, 1090-1100.

See Also

[brownie.lite](#), [evol.vcv](#)

---

exhaustiveMP	*Exhaustive and branch & bound MP optimization*

Description

This function does exhaustive and branch & bound MP searches.

Usage

```
exhaustiveMP(data, tree=NULL, method="branch.and.bound")
```

Arguments

data	is a [phyDat](#) (Schliep 2011) object containing DNA or other data.
tree	an optional input tree (used only with method="branch.and.bound").
method	an optional string indicatingn method to use: "branch.and.bound" or "exhaustive".

Details

Should probably not be used for more than about 8 species (and definitely not more than 10 species). Performs parsimony calculations using `parsimony` in the "phangorn" package (Schliep 2011).

Value

A `"phylo"` or `"multiPhylo"` object that is the MP tree or set of MP trees. It also returns the parsimony scores in attr(trees,`"pscore"`) or attr(trees[[i]],`"pscore"`) for the ith tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Felsenstein, J. (2004) *Inferring Phylogenies*. Sinauer.

Schliep, K. P. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**, 592-593.

See Also

mrp.supertree, optim.parsimony, pratchet

---

expm *Matrix exponential*

Description

Wrapper for `MatrixExp` that retains row/column names.

Usage

```
expm(Y)
```

Arguments

Y   a matrix.

Value

A matrix.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

export.as.xml   *Export trees & data in XML format*

Description

This function exports trees & character data in XML format.

Usage

```
export.as.xml(file, trees, X)
```

Arguments

file	filename for export.
trees	a phylogenetic tree or trees in "phylo" or "multiPhylo" format.
X	a matrix of class "DNAbin" or a matrix with discretely valued non-DNA character data.

Details

Can be used to create input file for the program SIMMAP v1.5 (Bollback 2006), also see: `http://www.simmap.org`.

Value

A file.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Bollback, J. P. (2006) Stochastic character mapping of discrete traits on phylogenies. *BMC Bioinformatics*, **7**, 88.

See Also

`make.simmap`, `read.nexus`, `read.simmap`, `write.simmap`

---

fancyTree   *Plots special types of phylogenetic trees*

Description

This function plots different types of phylogenetic trees. If type="extinction" (or any unambiguous abbreviation) the function will plot a tree in which branches preceding the MRCA of all extant taxa and branches leading only to extnct lineages are plotted with dashed red lines. If type="traitgram3d" the function will plot a three dimensional traitgram (that is, a projection of the tree into three dimensional morphospace where two dimensions are the phenotypic trait and the third axis is time since the root). In this case, the additional argument X, a matrix containing the tip values of all species (with species IDs as row names) should be supplied. Optionally, the user can also supply the matrix A, which contains the ancestral states in the tree with rows labeled by node number. If type="droptip" the function will create a two panel figure in which the first panel is the tree with lineages to be pruned highlighted; and the second panel is the pruned tree. In this case, the additional argument tip, the tip name or vector of tip names to be dropped, must be supplied. If type="xkcd", an xkcd.com style tree will be plotted. (See xkcdTree for additional optional arguments if type="xkcd".) If type="densitymap", a posterior probability density "heat-map" is created based on a set of trees in a "multiPhylo" object containing a binary [0,1] mapped character. (See densityMap for additional optional arguments if type="densitymap".) If type="contmap", reconstructed continuous trait evolution is mapped on the tree. Again, see contMap for additional arguments if type="contmap". Finally, if type="phenogram95" a 95-percent phenogram is plotted using transparency to visualize uncertainty at ancestral nodes and along branches. Most of the options of phenogram are available. Presently only type="traitgram3d" uses the list control which can be supplied the same set of control parameters as phylomorphospace3d, as well as the control parameter maxit which will be passed to anc.ML.

Usage

```
fancyTree(tree, type=c("extinction","traitgram3d","droptip","xkcd","densitymap",
"contmap","phenogram95"), ..., control=list())
```

Arguments

tree	a phylogenetic tree in "phylo" format.
type	the type of special plot to create. See Description.
...	arguments to be passed to different methods.
control	a list of control parameters, depending on type.

Value

This function plots different types of phylogenetic trees. For type="droptip" the function also returns the pruned tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

contMap, densityMap, drop.tip, phenogram, phylomorphospace3d, plot.phylo, plotSimmap, xkcdTree

Examples

```
# plot tree with extinction
set.seed(10)
tree<-rlineage(1,0.5,4)
fancyTree(tree,type="extinction")

## Not run:
# plot 3D traitgram
tree<-pbtree(n=50,scale=10)
Y<-sim.corrs(tree,vcv=matrix(c(1,0.75,0.75,1),2,2))
fancyTree(tree,type="traitgram3d",X=Y,control=list(spin=FALSE))

# plot with internal nodes from simulation
Y<-sim.corrs(tree,vcv=matrix(c(1,0.75,0.75,1),2,2),internal=TRUE)
B<-Y[length(tree$tip)+1:tree$Nnode,]; Y<-Y[1:length(tree$tip),]
fancyTree(tree,type="traitgram3d",X=Y,A=B,control=list(simple.axes=TRUE,spin=FALSE))

## End(Not run)

# plot with dropped tips
tree<-pbtree(n=30)
tips<-sample(tree$tip.label)[1:10]
pruned<-fancyTree(tree,type="droptip",tip=tips)

## Not run:
# plot 95-percent CI phenogram
tree<-pbtree(n=30)
x<-fastBM(tree)
fancyTree(tree,type="phenogram95",x=x)

## End(Not run)
```

---

fastAnc   *Fast estimation of ML ancestral states*

Description

This function performs fast estimation of the ML ancestral states for a continuous trait by taking advantage of the fact that the state computed for the root node of the tree during Felsenstein's (1985) contrasts algorithm is also the MLE of the root node. Thus, the function reroots the tree at all internal nodes and computes the contrasts state at the root each time. The function can also (optionally) compute variances or 95-percent confidence intervals on the estimates.

Usage

```
fastAnc(tree, x, vars=FALSE, CI=FALSE)
```

Arguments

tree	an object of class "phylo".
x	a vector of tip values for species; names(x) should be the species names.
vars	a logical value indicating whether or not to compute variances on the ancestral state estimates. Variances are based on Equation (6) of Rohlf (2001).
CI	a logical value indicating whether or not to compute 95-percent confidence intervals on state estimates.

Value

A named vector containing the states at internal nodes - names are node numbers; or a list containing ancestral state estimates (ace), variances on the estimates (var), and/or 95-percent confidence intervals (CI95).

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[ace](#), [anc.Bayes](#), [anc.ML](#), [pic](#)

Examples

```
tree<-rtree(50)
x<-fastBM(tree) # simulate using fastBM
fastAnc(tree,x) # estimate states
```

---

fastBM   *Fast Brownian simulation*

Description

This function conducts fast quantitative trait simulation on a phylogeny under several different models: Brownian motion (default), BM with a trend (for mu!=0), and bounds (for bounds!=c(-Inf,Inf)).

Usage

```
fastBM(tree, a=0, mu=0, sig2=1, bounds=c(-Inf,Inf), internal=FALSE, nsim=1)
```

Arguments

tree	is a phylogenetic tree in "phylo" format.
a	a value for ancestral state at the root node.
mu	an optional value for the mean of random normal changes along branches of the tree - can be used to simulate a trend if mu!=0.
sig2	instantaneous variance of the BM process.
bounds	a vector with the lower and upper bounds (respectively) for bounded Brownian simulation - by default simulation is unbounded.
internal	logical value indicating whether or not to return states for internal nodes.
nsim	number of simulations.

Value

A vector (for nsim=1) or matrix containing the tip states for the n species in the tree, and (optionally) the ancestral states for internal nodes.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

branching.diffusion

Examples

```
tree<-pbtree(n=1000)
x<-fastBM(tree,sig2=0.1) # Brownian motion
y<-fastBM(tree,mu=1) # with a trend
```

---

fastMRCA	*Get the MRCA of a pair of tip taxa*

Description

This function returns the most recent common ancestor (node number) for a pair of taxa.

Usage

```
fastMRCA(tree, sp1, sp2)
```

Arguments

tree	a phylogenetic tree as an object of class "phylo".
sp1	species name.
sp2	species name.

Details

This function is mostly redundant with `findMRCA` but for very large trees will be considerably faster. Function uses `Ancestors` in the phangorn package.

Value

The node number of the MRCA.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

`Ancestors`, `findMRCA`, `mrca`

Examples

```
tree<-pbtree(n=2000)
anc<-fastMRCA(tree,"t1","t15")
```

---

findMRCA   *Get the MRCA of a set of taxa*

Description

This function returns the most recent common ancestor (node number) for a set of taxa.

Usage

```
findMRCA(tree, tips=NULL)
```

Arguments

tree	a phylogenetic tree as an object of class "phylo".
tips	a vector containing a set of tip labels.

Details

If `tips==NULL` will return the result of a normal function call to `mrca`.

Value

The node number of the MRCA, or a matrix of node numbers (if `tips==NULL`).

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

mrca

Examples

```
tree<-pbtree(n=20)
anc<-findMRCA(tree,c("t1","t10","t15"))
```

---

fitBayes	*Evolutionary model fitting with intraspecific variability using Bayesian MCMC*

Description

This function uses Bayesian MCMC to sample terminal states (species means) as well as evolutionary parameters.

Usage

```
fitBayes(tree, x, ngen=10000, model="BM", method="reduced", control=list())
```

Arguments

tree	an object of class "phylo".
x	a vector of phenotypic values for individuals; names(x) should contain the species names (not individual IDs).
ngen	a integer indicating the number of generations for the MCMC.
model	an evolutionary model: either "BM" or "lambda".
method	a method: either "reduced" or "full".
control	a list of control parameters containing the following elements: sig2: starting value for σ^2 (BM rate); lambda: starting value for the λ parameter; a: starting for the state at the root node; xbar: starting values for the states at the tips; intV: starting value for the intraspecific variance (reduced method); or v: starting value for the vector of intraspecific variances for all species (full method); pr.mean: means for the prior distributions in the following order - sig2, lambda (if applicable), a, xbar, intV or v (if applicable), note that the prior probability distribution is exponential for sig2 and normal for a and y; pr.var: variances on the prior distributions, same order as pr.mean.

Value

A matrix with number of rows ngen/control$sample+1 containing the posterior sample and likelihoods. Matrix columns are labeled by species (for species means and variances), or by the corresponding evolutionary parameter.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. and R. G. Reynolds. (2012) A new Bayesian method for fitting evolutionary models to comparative data with intraspecific variation. *Evolution*, 66, 2697-2707.

See Also

[anc.Bayes](), [brownie.lite](), [evol.rate.mcmc]()

---

fitDiversityModel   *Fit diversity-dependent phenotypic evolution model*

Description

This function fits a diversity-dependent phenotypic evolution model (based on Mahler et al. 2010).

Usage

```
fitDiversityModel(tree, x, d=NULL, showTree=TRUE, tol=1e-6)
```

Arguments

tree	an object of class ″phylo″.
x	a vector with tip values for a continuously distributed trait.
d	a vector containing the inferred historical diversity at each node in the tree - if d=NULL (the default) function will treat the diversification as if it occurred in a single geographic area.
showTree	optional logical value indicating whether to plot the tree transformation implied by the model.
tol	some small value by which d is incremented during rescaling of psi for optimization. If R thinks your matrices are singular during optimization, try increasing tol slightly.

Value

A list with the following components:

logL	log-likelihood of the fitted model.
sig0	estimated starting value for the rate at the root of the tree.
psi	the estimated rate of change in the rate associated with the addition of a lineage.
vcv	a matrix with the variances and covariance of the estimated parameters (from the Hessian).

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Mahler, D. L, L. J. Revell, R. E. Glor, and J. B. Losos. 2010. Ecological opportunity and the rate of morphological evolution in the diversification of Greater Antillean anoles. *Evolution*, **64**, 2731-2745.

See Also

[brownie.lite](), [estDiversity](), [evol.rate.mcmc]()

---

gammatest   *Gamma test of Pybus & Harvey (2000)*

Description

Conducts γ-test of Pybus & Harvey (2000).

Usage

```
gammatest(x)
```

Arguments

x   list from function call of [ltt]() in which ltt(...,gamma=F).

Details

Do not use for object returned by ltt(...,gamma=T).

Value

A list containing:

gamma   optionally, a value for the γ-statistic.
p   two-tailed P-value for the γ-test.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Pybus, O. G., and P. H. Harvey (2000) Testing macro-evolutionary models using incomplete molecular phylogenies. *Proc. R. Soc. Lond. B*, **267**, 2267-2272.

See Also

[ltt](ltt)

Examples

```
tree<-pbtree(n=200)
z<-ltt(tree,gamma=FALSE)
g<-gammatest(z)
```

---

getCladesofSize   *Get all subtrees larger than or equal to a specified size*

Description

This function gets all subtrees that cannot be further subdivided into two reciprocally monophyletic subtrees of size >= `clade.size`.

Usage

```
getCladesofSize(tree, clade.size=2)
```

Arguments

tree   is an object of class ″phylo″.

clade.size   subtree size.

Value

An object of class ″multiPhylo″.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

extract.clade, getDescendants

---

getDescendants   *Get descendant node numbers*

Description

This function returns the set of node & tip numbers descended from node.

Usage

```
getDescendants(tree, node, curr=NULL)
```

Arguments

tree	a phylogenetic tree as an object of class "phylo".
node	an integer specifying a node number in the tree.
curr	the set of previously stored node numbers - used in recursive function calls.

Value

The set of node and tip numbers for the nodes and tips descended from node in a vector.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

paintSubTree

---

getExtant   *Returns a list of the extant or extinct lineages in a tree containing non-contemporaneous tips*

Description

The function getExtant takes a tree as input and returns a vector containing the names of all the tips that have a height above the root that is equal (to a degree of numerical precision determined by tol) to the height of the highest tip. getExtinct returns the complement.

Usage

```
getExtant(tree, tol=1e-8)
getExtinct(tree, tol=1e-8)
```

Arguments

tree	a phylogeny stored as an object of class "phylo" with some tips that are non-contemporaneous (i.e., end before the present).
tol	a tolerance value to account for numerical imprecision.

Value

A vector with the tip names of extant or extinct species in the tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[nodeHeights](#)

---

getSisters	*Get the sister node number, label, or set of nodes for a node or tip*

Description

This function takes a tree and node or tip number of label and returns the number or label of the sister or sisters to that node or tip.

Usage

```
getSisters(tree, node, mode=c("number","label"))
```

Arguments

tree	object of class "phylo".
node	a node number, tip number, node label, or tip label.
mode	an optional string indicating whether to return the node or tip number(s) or the node or tip label(s), if available.

Value

If mode="number" this function returns an integer or vector containing the node number of numbers of the sister node or tip. If mode="label" that this function returns a list containing up to two vectors: one for the node numbers of labels of sister nodes (if available); and the other containing the tip labels of the sister tips.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

getStates   *Get the states at nodes or tips from a mapped tree*

Description

This function gets the states from the nodes or tips of a mapped tree (e.g., `make.simmap`).

Usage

```
getStates(tree, type=c("nodes","tips"))
```

Arguments

tree	is a modified object of class `"phylo"` or `"multiPhylo"`.
type	mode indicating whether to get states at the nodes (type=`"nodes"`) or the tips (type=`"tips"`) of the tree.

Value

A named vector (for `"phylo"`) or matrix (for `"multiPhylo"`).

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

describe.simmap, make.simmap, read.simmap, sim.history

---

lambda.transform   *Lambda transformation of matrix*

Description

Function multiplies the off-diagonals of a square matrix by `lambda`. Used internally in `phyl.pca` and other functions.

Usage

```
lambda.transform(lambda, C)
```

Arguments

lambda	scalar, usually (but not necessarily) on the interval 0,1.
C	matrix probably returned by `vcv.phylo`.

Details

Do not use unless you know what you're doing.

Value

A matrix.

Author(s)

Liam Revell <liam.revell@umb.edu>

Examples

```
tree<-pbtree(n=100)
V.lambda<-lambda.transform(0.7,vcv.phylo(tree))
```

---

likMlambda   *Likelihood for joint lambda*

Description

Computes the likelihood.

Usage

```
likMlambda(lambda, X, C)
```

Arguments

lambda	scalar, usually on the interval 0,1.
X	data, in a matrix.
C	matrix probably returned by `vcv.phylo`.

Details

Do not use unless you know what you're doing.

Value

A scalar.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

ls.tree   *Least squares branch lengths for a given tree*

Description

Computes the least squares branch lengths conditioned on a topology and distance matrix. Internal function for `optim.phylo.ls`.

Usage

```
ls.tree(tree, D)
```

Arguments

tree   phylogeny.

D   distance matrix.

Details

Do not use unless you know what you're doing.

Value

A tree with branch lengths.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

ltt   *Creates lineage-through-time plot (including extinct lineages)*

Description

This function computes LTT plot with extant and extinct lineages, and optionally conducts γ-test of Pybus & Harvey (2000).

Usage

```
ltt(tree, plot=TRUE, drop.extinct=FALSE, log.lineages=TRUE, gamma=TRUE)
```

Arguments

tree	is a phylogenetic tree in "phylo" format, or an object of class "multiPhylo" containing a list of phylogenetic trees.
plot	a logical value indicating whether or not to create LTT plot.
drop.extinct	logical value indicating whether or not to drop extinct tips from the tree.
log.lineages	logical value indicating whether LTT plot should be on log-linear (default) or linear-linear scale.
gamma	logical value indicating whether or not to compute eqngamma from Pybus & Harvey (2000; *Proc. Roy. Soc. B*).

Details

It's unclear how to interpret the γ-statistic if not all the tips in the tree are contemporaneous.

Value

A list with the following components:

times	a vector of branching times.
ltt	a vector of linages.
gamma	optionally, a value for the gamma-statistic.
p	two-tailed P-value for the gamma-test.

If tree is an object of class "multiPhylo", a list of the above list will be returned.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Pybus, O. G., and P. H. Harvey (2000) Testing macro-evolutionary models using incomplete molecular phylogenies. *Proc. R. Soc. Lond. B*, **267**, 2267-2272.

See Also

[gammatest](), [ltt95]()

Examples

```
trees<-pbtree(n=100,scale=100,nsim=10)
ltt(trees,log.lineages=FALSE)
tree<-rlineage(birth=1,death=0.25,Tmax=4)
ltt(tree,gamma=FALSE)
```

---

ltt95   *Creates a (1-α)-percent CI for a set of LTTs*

Description

This function computes LTT plots for a set of trees & plots a (1-α)-percent CI by various methods.

Usage

```
ltt95(trees, alpha=0.05, log=FALSE, method=c("lineages","times"),
mode=c("median","mean"), ...)
```

Arguments

trees	is an object of class "multiPhylo" containing a list of phylogenetic trees.
alpha	confidence level.
log	logical value indicating whether or not to plot on the log-scale.
method	plot the CI on the number of lineages given time ("lineages"); or on times given a number of lineages ("times").
mode	plot the median or mean LTT.
...	optional arguments. So far, res gives the number of time-steps (defaults to res=100.

Details

This function creates a plot and returns a matrix with the results.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[ltt](ltt)

---

make.era.map *Create "era" map on a phylogenetic tree*

Description

This function creates a temporal map on the tree based on limits provided by the user.

Usage

```
make.era.map(tree, limits)
```

Arguments

tree   a phylogenetic tree as an object of class "phylo".

limits   a vector containing the temporal limits, in time since the root node of the tree, for the mappings.

Value

A modified phylogenetic tree of class "phylo" with the following additional elements:

maps   a list of named vectors containing the times spent in each state on each branch, in the order in which they occur.

mapped.edge   a matrix containing the total time spent in each state along each edge of the tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[make.simmap](#), [read.simmap](#), [plotSimmap](#)

Examples

```
tree<-pbtree(n=1000,scale=100)
tree<-make.era.map(tree,c(0,25,50,75))
plotSimmap(tree,pts=FALSE,ftype="off")
```

Description

This function performs stochastic mapping using several methods.

For Q="empirical", it first fits a continuous-time reversible Markov model for the evolution of x and then simulates stochastic character histories using that model and the tip states on the tree. This is the same procedure that is described in Bollback (2006), except that simulation is performed using a fixed value of the transition matrix, Q, instead of by sampling Q from its posterior distribution.

For Q="mcmc", it first samples Q nsim times from the posterior probability distribution of Q using MCMC, then it simulates nsim stochastic maps conditioned on each sampled value of Q.

For Q set to a matrix, it samples stochastic mappings conditioned on the fixed input matrix.

Usage

```
make.simmap(tree, x, model="SYM", nsim=1, ...)
```

Arguments

tree	a phylogenetic tree as an object of class "phylo", or a list of trees as an object of class "multiPhylo".
x	a vector containing the tip states for a discretely valued character, or a matrix containing the prior probabilities of tip states in rows.
model	a character string containing the model - options as in [ace].
nsim	number of simulations. If tree is an object of class "multiPhylo", then nsim simulations will be conducted per tree.
...	optional arguments. So far, pi gives the prior distribution on the root node of the tree - options are "equal", "estimated", or a vector with the frequencies. If pi="estimated" then the stationary distribution is estimated by numerically solving pi*Q=0 for pi, and this is used as a prior on the root. Defaults to pi="equal" which results in the root node being sampled from the conditional scaled likelihood distribution at the root. message tells whether or not to print a message containing the rate matrix, *Q* and state frequencies. Defaults to message=TRUE. For optional argument Q="mcmc" the mean value of Q from the posterior sample is printed. tol gives the tolerance for zero elements in Q. (Elements less then tol will be reset to tol). Q can be a string ("empirical" or "mcmc"), or a fixed value of the transition matrix, Q. If "empirical" than a single value of Q, the most likely value, is used for all simulations. If "mcmc", then nsim values of Q are first obtained from the posterior distribution for Q using Bayesian MCMC, then a simulated stochastic character map is generated for each value of Q. vQ a single numeric value or a vector containing the (normal) sampling variances for the MCMC. The order of vQ is assumed to be in the order of the index.matrix in [ace] for the chosen model. prior a list containing alpha and beta parameters for the gamma prior distribution on the transition

rates in Q. Note that alpha and beta can be single values or vectors, if different priors are desired for each value in Q. As for vQ, the order of prior is assumed to be the order of index.matrix in ace. prior can also be given the optional logical value use.empirical which tells the function whether or not to give the prior distribution the empirical mean for Q. If TRUE then only prior$beta is used and prior$alpha is set equal to prior$beta times the empirical mean of Q. burnin and samplefreq are burn-in and sample frequency for the MCMC, respectively.

Details

Uses code modified from ace (by Paradis et al.) to perform Felsenstein's pruning algorithm & compute the likelihood.

As of phytools>=0.2-33 x can be a vector of states or a matrix containing the prior probabilities of tip states in rows. In this case the column names of x should contain the states, and the row names should contain the tip names.

Note that there was a small (but potentially significant) bug in how node states were simulated by make.simmap in versions of phytools<=0.2-26. Between phytools 0.2-26 and 0.2-36 there was also a bug for asymmetric models of character change (e.g., model="ARD"). Finally, between phytools 0.2-33 and phytools 0.2-47 there was an error in use of the conditional likelihoods for the root node, which caused the root node of the tree to be sampled incorrectly. All of these issues should be fixed in the present version.

Q="mcmc" and Q set to a fixed value were introduced to phytools >= 0.2-53. As of the present version of phytools, this method is still somewhat experimental & should be used with caution.

If tree is an object of class "multiPhylo" then nsim stochastic maps are generated for each input tree.

Value

A modified phylogenetic tree of class "phylo" (or a modified "multiPhylo" object, for nsim > 1) with the following additional elements:

maps	a list of named vectors containing the times spent in each state on each branch, in the order in which they occur.
mapped.edge	a matrix containing the total time spent in each state along each edge of the tree.
Q	the assumed or sampled value of Q.
logL	the log-likelihood of the assumed or sampled Q.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

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

Huelsenbeck, J. P., R. Neilsen, and J. P. Bollback (2003) Stochastic mapping of morphological characters. *Systematic Biology*, **52**, 131-138.

Bollback, J. P. (2006) Stochastic character mapping of discrete traits on phylogenies. *BMC Bioinformatics*, **7**, 88.

See Also

[brownie.lite](), [brownieREML](), [countSimmap](), [describe.simmap](), [evol.vcv](), [plotSimmap](), [read.simmap](), [write.simmap]()

---

map.overlap	*Proportional overlap between two mapped character histories on a tree*

Description

This function computes the fraction of a stochastic character mapping that is shared between two differently mapped trees.

Usage

```
map.overlap(tree1, tree2, tol=1e-6)
```

Arguments

tree1	a modified "phylo" object (see [read.simmap]().
tree2	a modified "phylo" object with the same topology and branch lengths, but different map from tree1.
tol	an optional tolerance value.

Value

A numerical value on the interval 0-1.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[make.simmap](), [read.simmap]()

matchNodes   *Matches nodes between two trees*

Description

This function returns a matrix in which the first column contains all the internal nodes for tr1 and the second column contains the matching nodes from tr2, inasmuch as they can be identified. For method="descendants", pairs of matching nodes are defined by sharing all descendant leaves in common. for method="distances", nodes are considered to matched if the share the same set of distances (or proportional distances, for optional argument corr=TRUE) to all tips.

Usage

```
matchNodes(tr1, tr2, method=c("descendants","distances"), ...)
```

Arguments

tr1	first tree.
tr2	second tree.
method	method to use to match nodes between trees. "descendants" uses the tip species descended from each node; "distances" uses the distances from the nodes to the tips. Any umambiguous shortening of "descendants" or "distances" is allowed.
...	optional arguments which may or may not be used depending on method. tol is a tolerance value for the difference from exact matching that is allowed for method="distances". corr, which is FALSE by default, indicates whether to match nodes under method="distances" using the correlation (corr=TRUE) or the absolute similarity of distances.

Details

Primarily designed to be used internally by [fastAnc](#).

Value

A matrix in which the first column contains the nodes of tr1 with the second column containing matching nodes in tr2, with the criterion for matching defined by method.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

mergeMappedStates   *Merge two or more mapped states into one state*

Description

This function merges two or mapped states on the tree to get one new state. For instance, one could merge the states ″C″, ″G″, and ″T″ to get the state ″not-A″.

Usage

```
mergeMappedStates(tree, old.states, new.state)
```

Arguments

tree	a modified object of class ″phylo″ or ″multiPhylo″ with a mapped discrete character.
old.states	states to merge.
new.state	name for new state.

Value

A modified object of class ″phylo″ or ″multiPhylo″.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[make.simmap](), [read.simmap]()

---

minSplit   *Finding the minimum (median) split in the posterior sample*

Description

This function takes a phylogenetic tree and a list of splits and identifies the split with the smallest summed or summed squared distances to all the other splits. Used to be called min.split() but was changed to avoid conflict with the generic [min]().

Usage

```
minSplit(tree, split.list, method="sum", printD=FALSE)
```

Arguments

tree	a phylogeny stored as an object of class "phylo".
split.list	either a matrix with two named columns, "node" and "bp"; a $mcmc matrix from evol.rate.mcmc(); or the entire raw output from evol.rate.mcmc().
method	an optional string indicating the criterion to minimize: options are "sum" and "sumsq".
printD	logical specifying whether to print distances to screen (FALSE by default).

Value

A list with the following components:

node	node for the minimum split.
bp	location on the branch leading to node of the minimum split.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J., D. L. Mahler, P. Peres-Neto, and B. D. Redelings (2012) A new method for identifying exceptional phenotypic diversification. *Evolution*, **66**, 135-146.

See Also

[evol.rate.mcmc](), [posterior.evolrate]()

---

mrp.supertree	*Matrix representation parsimony supertree estimation*

Description

This function estimates the MRP (matrix representation parsimony) supertree from a set of trees.

Usage

```
mrp.supertree(trees, method=c("pratchet","optim.parsimony"), ...)
```

Arguments

trees	an object of class "multiPhylo" (i.e., a list of trees).
method	an argument specifying whether to optimize the tree using [pratchet]() or [optim.parsimony]().
...	optional arguments - mostly to be passed to [pratchet]() or [optim.parsimony]().

Details

Function uses pratchet or optim.parsimony from the "phangorn" package (Schliep 2011) and prop.part from the "ape" package (Paradis et al. 2004). See pratchet or optim.parsimony for optional arguments, which vary slightly depending on the method. All optional arguments of these methods are available to the user with one exception. The argument tree in optim.parsimony is supplied instead as start. In addition to being an object of class "phylo", start can also be assigned the string values of "NJ" or "random", in which case either a neighbor-joining or random tree will be used as the starting tree for optimization.

Value

A "phylo" or "multiPhylo" object that is the MP or set of MP MRP trees.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Felsenstein, J. (2004) *Inferring Phylogenies*. Sinauer.

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

Schliep, K. P. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**, 592-593.

See Also

exhaustiveMP, optim.parsimony, pratchet

---

multi.mantel	*Multiple matrix regression (partial Mantel test)*

Description

This function conducting a multiple matrix regression (partial Mantel test) and uses Mantel (1967) permutations to test the significance of the model and individual coefficients. It also returns the residual and predicted matrices.

Usage

```
multi.mantel(Y, X, nperm=1000)
```

Arguments

Y	single "dependent" square matrix. Can be either a symmetric matrix of class "matrix" or a distance matrix of class "dist".
X	a single independent matrix or multiple independent matrices in a list. As with Y can be a object of class "matrix" or class "dist".
nperm	number of Mantel permutations.

Value

A list with the following components:

r.squared	multiple R-squared.
coefficients	model coefficients, including intercept.
tstatistic	t-statistics for model coefficients.
fstatistic	F-statistic for the overall model.
probt	vector of probabilities, based on permutations, for tstatistic.
probF	probability of F, based on Mantel permutations.
residuals	matrix of residuals.
predicted	matrix of predicted values.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Mantel, N. (1967) The detection of disease clustering and a generalized regression approach. *Cancer Research*, **27**, 209–220.

---

multiC	*Returns a list with phylogenetic VCV matrix for each mapped state*

Description

This function takes a modified "phylo" object as input and returns a set of so-called phylogenetic covariance matrices as a list: one for each mapped state.

Usage

```
multiC(tree)
```

Arguments

tree	a phylogeny with mapped discrete state in a modified object of class "phylo" (e.g., see read.simmap.

Value

A list of matrices.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[evolvcv.lite](), [read.simmap](), [vcvPhylo](), [vcv.phylo]()

---

nodeHeights *Compute the heights above the root of each node*

Description

This function computes node heights.

Usage

```
nodeHeights(tree)
```

Arguments

tree   a phylogeny as an object of class "phylo".

Details

The function also gives a handy way to get the total length of the tree from the root to the heighest tip which will be given by max(nodeHeights(tree)).

Value

A matrix of the same dimensions as tree$edge containing the height above the root of each node in edge.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[estDiversity](), [fitDiversityModel]()

Examples

```
tree<-rtree(10)
X<-nodeHeights(tree)
```

---

optim.phylo.ls *Phylogeny inference using the least squares method*

Description

This function performs phylogeny inference using least-squares.

Usage

```
optim.phylo.ls(D, stree=NULL, set.neg.to.zero=TRUE, fixed=FALSE,
tol=1e-10, collapse=TRUE)
```

Arguments

D   a distance matrix.

stree   an optional starting tree for the optimization.

set.neg.to.zero
                    a logical value indicating whether to set negative branch lengths to zero (default
                    TRUE).

fixed   a logical value indicating whether to estimate the topology - if TRUE only the
                    branch lengths will be computed.

tol   a tolerance value used to assess whether the optimization has converged.

collapse   a logical indicating whether to collapse branches with zero length.

Details

Function uses nni from the "phangorn" package (Schliep 2011) to conduct NNIs for topology estimation. Since topology optimization is performed using NNIs, converge to the true least-squares topology is not guaranteed. It is consequently probably wise to start with a very good tree - such as a NJ tree.

Value

An objec of class "phylo" that (may be) the least-squares tree with branch lengths; also returns the sum of squares in attr(tree,"Q-score").

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Cavalli-Sforza, L. L., and A. W. F. Edwards. (1967) Phylogenetic analysis: Modesl and estimation procedures. *American Journal of Human Genetics*, **19**, 233-257.

Felsenstein, J. (2004) *Inferring Phylogenies*. Sinauer.

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

Schliep, K. P. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**, 592-593.

See Also

[exhaustiveMP](), [nni]()

---

orderMappedEdge   *Order the columns of mapped.edge to match across trees*

Description

This function takes a modified object of class `"multiPhylo"` with a mapped discrete character (e.g., see [read.simmap]() and sorts the columns of each `tree$mapped.edge` to have the same state ordering. Also works for a single tree.

Usage

```
orderMappedEdge(trees, ordering=NULL)
```

Arguments

trees   object of class `"phylo"` or `"multiPhylo"`.

ordering   ordering for the columns of `$mapped.edge`. If NULL, then an alphabetical order is assumed. Options are `"alphabetical"`, `"numerical"`, or any specific ordering of the mapped traits (e.g., `c("A","B","C")`).

Value

A modified object of class `"phylo"` or `"multiPhylo"`.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

paintSubTree　　　　　　　*Paint sub-trees with a discrete character*

Description

This function maps or "paints" an arbitrary, i.e., user-specified, discrete character history on the tree.

Usage

```
paintSubTree(tree, node, state, anc.state="1", stem=FALSE)
```

Arguments

tree	a phylogenetic tree as an object of class ″phylo″ or a modified object with mapped character traits.
node	an integer specifying the node number tipward of which the function should paint the derived state.
state	a string (or numeric value) specifying the state to paint on the tree tipward of node.
anc.state	the ancestral state to use; will only be applied if there are presently no character values mapped on the tree.
stem	logical or numeric value indicating whether to use the derived state on the stem leading to node (or not, if stem=FALSE), or, alternatively, what fraction of the stem should be assigned to the derived clade. Note that for tip clades stem=FALSE is not allowed.

Value

A modified phylogenetic tree of class ″phylo″ with the following additional elements:

maps	a list of named vectors containing the times spent in each state on each branch, in the order in which they occur.
mapped.edge	a matrix containing the total time spent in each state along each edge of the tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[make.simmap](), [read.simmap](), [plotSimmap](), [sim.history]()

---

paste.tree   *Paste two trees together*

Description

Primarily internal function for `posterior.evolrate`; can be used to graft a clade into a receptor tree, at the "sticky tip" labeled with `"NA"`.

Usage

```
paste.tree(tr1, tr2)
```

Arguments

tr1   receptor tree.

tr2   donor clade.

Details

The donor clade needs to have a root edge, even if it is zero length.

Value

A tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

Examples

```
tr1<-rtree(10)
tr2<-rtree(10)
tr1$tip.label[1]<-"NA"
tr2$root.edge<-0
tr3<-paste.tree(tr1,tr2)
```

---

pbtree   *Simulate pure-birth or birth-death stochastic tree or trees*

Description

This function simulates stochastic birth-death trees. Simulation can be performed conditioning on n, on t, or on both simultaneously. If the both, then (for optional argument method="rejection") rejection sampling is performed whereby trees are simulated given b and t until a tree containing n taxa is found. The giving-up point can be set using the optional argument max.count. Simulations can also be performed in continuous time (the default) or discrete time; the difference being that wait times in the continous-time simulation come from the exponential distribution; whereas waiting times in discrete-time simulations come from the geometric distribution. In addition, discrete-time simulations allow for the possibility that multiple speciation events can occur at (exactly) the same time, so long as they are on separate branches. Finally, sometimes for stopping criterion n in discrete-time there will be a number of tips different from n. This indicates that the last event contained more than one speciation event, and a warning is printed.

method="direct" is presently experimental. It does not really perform direct sampling; however waiting times & birth or death events are sampled first - with only wait-times consistent with n and t being retained. This rejection sampling occurs one layer earlier than for method="rejection". This results in a significant (several-fold) speed-up of the code and enables sampling conditioned on n and t simultaneously for much higher b and d. At the present time, extant.only=TRUE does not work for this mode, nor does type="discrete".

Note that if ape=FALSE, then the function will run faster, and the tree is theoretically compatible with the ape "phylo" standard; however some downstream errors with functions such as [bind.tree](#) have been observed.

Usage

```
pbtree(b=1, d=0, n=NULL, t=NULL, scale=NULL, nsim=1,
type=c("continuous","discrete"), ...)
```

Arguments

b	birth rate or speciation rate for type="continuous"; the probability of speciating per time-step for type="discrete".
d	death rate or extinction rate for type="continuous"; the probability of going extinct per time-step for type="discrete".
n	desired number of species (i.e., taxa-stop criterion).
t	total time for simulation (i.e., time-stop criterion).
scale	if set, rescales tree to have total length scale.
nsim	number of simulated trees to return.
type	string to indicate whether to simulate trees in continuous or discrete time. If the former, then wait times between speciation events are drawn from an exponential distribution; whereas if the latter then wait times comes from a geometric distribution.

...   optional arguments including ape, a logical value indicating whether to return nodes in a 'ape' compatible ordering (default is TRUE); extant.only a logical value indicating whether or not to return only extant species (defaults to FALSE); max.count a numeric value indicating the maximum number of iterations to run is sampling conditioned on both n and t (defaults to 1e5); method gives the method used form simultaneously conditioning on n and t - options are "rejection" and "direct"; and, finally, quiet, a logical value indicating whether or not to suppress certain message (defaults to FALSE).

Value

A tree or list of trees as an object of class "phylo" or "multiPhylo", respectively.

Author(s)

Liam Revell <liam.revell@umb.edu>

Examples

```
# simulate a pure-birth tree with 1000 tips, scaled to a length of 1.0
tree<-pbtree(n=1000,scale=1)
# simulate a pure-birth tree conditioning on n & t
tt<-log(50)-log(2)
tree<-pbtree(n=50,t=tt)
```

---

pgls.Ives   *Phylogenetic regression with intraspecific sampling error*

Description

This function fits the phylogenetic regression model with within-species sampling error following Ives et al. (2007).

Usage

```
pgls.Ives(tree, X, y, Vx=NULL, Vy=NULL, Cxy=NULL, lower=c(1e-8,1e-8))
```

Arguments

tree   a phylogeny as an object of class "phylo".

X   a named vector containing a *single* independent variable (multiple independent variables to be added in future). X can contain the species means, or a single long vector containing the sample of values for each species. In the latter case the names(X) will be repeating - all samples from the same species should have the same name.

y   vector the dependent variable. Can be species means or individual values, as for X.

Vx	sampling variances for X. If NULL, then the within-species variance is computed from the data assuming that individual samples, not species means, have been provided in X.
Vy	sampling variances for y. If NULL, then the within-species variance is computed from the data assuming that individual samples, not species means, have been provided in y.
Cxy	sampling covariances between X and y. This will also be computed from the data if Cxy==NULL. Note than in this case - but not for the calculation of Vx and Vy, the same number of observations and the same ordering must be provided for X and y. If this is not the case, then it is assumed that different individuals have been sampled for X and y and thus Cxy is assumed to be zero for all species.
lower	vector specifying the lower bounds for estimation for sig2x and sig2y, respectively. (Must be >0.)

Details

Presently only the bivariate regression model is implemented. Uses optim(...,method="L-BFGS-B") for optimization.

Value

A list with the following components:

beta	a vector or matrix of regression coefficients.
sig2x	fitted BM rate for X.
sig2y	fitted BM rate for y.
a	fitted ancestral states for X and y.
logL	log-likelihood.
convergence	a value for convergence. convergence=0 is good; see [optim](#) for more details.
message	a message for convergence.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Ives, A. R., P. E. Midford, and T. Garland Jr. (2007) Within-species measurement error in phylogenetic comparative methods. *Systematic Biology*, **56**, 252–270.

See Also

[brownie.lite](#), [phylosig](#), [phyl.resid](#)

---

phenogram	*Plot phenogram (traitgram)*

Description

Function plots a traitgram, that is, a projection of the phylogenetic tree in a space defined by phenotype (on the *y* axis) and time (on the *x*). If a discrete character is mapped on the tree this will also be plotted.

Usage

```
phenogram(tree, x, fsize=1.0, ftype="reg", colors=NULL, axes=list(),
add=FALSE, ...)
```

Arguments

tree	an object of class "phylo", with or without a mapped discrete character.
x	a vector containing the states at the tips *or* the states at all the tips and the internal nodes of the tree.
fsize	relative font size for tip labels.
ftype	font type - options are "reg", "i" (italics), "b" (bold), or "bi" (bold-italics).
colors	colors for plotting the mapped character (if available) in tree. If no character is mapped on the tree, then a single color for all the branches of the tree can be provided.
axes	list of axis dimensions. Items are time and trait.
add	optional logical value indicating whether to add to an open plot. If TRUE, then new axes will not be plotted.
...	optional arguments including xlim, ylim, log, main, sub, xlab, ylab, asp, type, lty, lwd, offset, digits, and digits are as in plot.default or [par](par). Note that axes overrides xlim and ylim. Finally, spread.labels is a logical value indicating whether or not to minimize tip label overlap (default is FALSE); spread.cost is a numeric vector indicating the relative penalty to be used for label overlap and deviance if spread.labels=TRUE; finally, link is a numeric value by which to offset the tip labels, linking them to the tips with a dashed line (default is 0 if spread.labels=FALSE or 0.1*max(nodeHeights(tree)) otherwise). The optional argument offsetFudge "fudges" the computation of label offset in scaling xlim. It is 1.37, which is the correct fudge in the Windows R GUI, but this may need to be changed in other systems.

Details

For spread.labels=TRUE numerical optimization is performed to optimize the distribution of the labels vertically, where the solution depends on the vector spread.cost containing the cost of overlap (first) and the cost of deviation from the vertical position of the tip. Because this is done via numerical optimization, plotting may hang briefly while the best solution is found (especially for large trees).

Value

Plots a traitgram, optionally with a mapped discrete character.

Author(s)

Liam Revell <liam.revell@umb.edu>

Examples

```
tree<-pbtree(n=20,scale=2)
x<-fastBM(tree)
phenogram(tree,x)
# or, simulate a discrete character history
tree<-sim.history(tree,Q=matrix(c(-1,1,1,-1),2,2),anc="1")
# simulate in which the rate depends on the state
x<-sim.rates(tree,c(1,10))
phenogram(tree,x)
# now use spread.labels
tree<-pbtree(n=40)
x<-fastBM(tree)
phenogram(tree,x,spread.labels=TRUE,spread.cost=c(1,0))
```

---

phyl.cca   *Phylogenetic canonical correlation analysis*

Description

This function performs phylogenetic canonical correlation analysis (e.g., Revell & Harrison 2008; *Bioinformatics*).

Usage

```
phyl.cca(tree, X, Y, lambda=1.0, fixed=TRUE)
```

Arguments

tree	a phylogenetic tree in "phylo" format.
X	a data matrix with traits in columns.
Y	data matrix with traits in columns, to be correlated with X.
lambda	optionally, a (fixed) value for lambda.
fixed	optionally, a logical value indicating whether or not to estimate lambda using likelihood.

Details

(Optional) joint optimization of λ is performed using [optimize](#) on the interval (0,1).

Value

A list with the following components:

cor	canonical correlations.
xcoef	coefficients for the canonical variables for X.
ycoef	coefficients for the canonical variables for Y.
xscores	matrix with the canonical scores for X.
yscores	matrix with the canonical scores for Y.
chisq	vector of χ^2 values.
p	P-values for the hypothesis test that the *i*th and all subsequent correlations are zero.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. & A. S. Harrison. (2008) PCCA: A program for phylogenetic canonical correlation analysis. *Bioinformatics*, **24**, 1018–1020.

See Also

phyl.pca

---

phyl.pairedttest   *Phylogenetic paired t-test*

Description

This function conducts a phylogenetic paired t-test, roughly following Lindenfors et al. (2010; *J. Evol. Biol.*).

Usage

```
phyl.pairedttest(tree, x1, x2=NULL, se1=NULL, se2=NULL, lambda=1.0, h0=0.0,
fixed=FALSE)
```

Arguments

tree	a phylogeny as an object of class "phylo".
x1	data vector for first trait, or matrix with two traits in columns.
x2	data vector for second trait (or null if x1 is a matrix).
se1	standard errors for x1.
se2	standard errors for x2.
lambda	starting value for Pagel's lambda (or fixed value, if fixed=TRUE).
h0	null hypothesis (to be tested) for the mean difference between x1 and x2.
fixed	logical value specifying whether or not to optimize lambda.

Details

Likelihood optimization is performed using [optim](#) with method="L-BFGS-B" with box constraints on lambda (0,1).

Value

A list with the following components:

dbar	phylogenetic mean differenc.
se	standard error of dbar.
sig2	estimated evolutionary variance (of the difference).
lambda	fitted (or fixed) value of lambda.
logL	log-likelihood of the fitted model.
t.dbar	t-value ((dbar-h0)/se where se is computed from the Hessian).
P.dbar	P-value.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Lindenfors, P., L. J. Revell, and C. L. Nunn (2010) Sexual dimorphism in primate aerobic capacity: A phylogenetic test. *J. Evol. Biol.*, **23**, 1183–1194.

| phyl.pca | *Phylogenetic principal components analysis* |

Description

This function performs phylogenetic PCA (e.g., Revell 2009; *Evolution*).

Usage

```
phyl.pca(tree, Y, method="BM", mode="cov")
```

Arguments

tree	phylogeny as an object of class "phylo".
Y	data matrix with traits in columns.
method	method to obtain the correlation structure: can be "BM" or "lambda".
mode	is the mode for the PCA: can be "cov" or "corr".

Details

If method="lambda" λ is optimized on the interval (0,1) using `optimize`.

Value

A list with the following components:

Eval	diagonal matrix of eigenvalues.
Evec	matrix with eigenvectors in columns.
S	matrix with scores.
L	matrix with loadings.
lambda	fitted value of lambda (method="lambda" only).
logL	log-likelihood for lambda model (method="logL" only).

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2009) Size-correction and principal components for interspecific comparative studies. *Evolution*, **63**, 3258–3268.

See Also

[phyl.cca](), [phyl.resid]()

---

phyl.resid   *Phylogenetic size-correction via GLS regression*

Description

This function fits one or multiple phylogenetic regressions (depending on the number of columns in Y) and computes the residuals. Designed for phylogenetic size correction using GLS regression (e.g., Revell 2009; *Evolution*).

Usage

```
phyl.resid(tree, x, Y, method="BM")
```

Arguments

tree	a phylogenetic tree in "phylo" format.
x	vector containing the single independent variable (e.g., size), or matrix with multiple independent variables in columns.
Y	vector or matrix with one or multiple dependent variables in columns.
method	method to obtain the correlation structure: can be "BM" or "lambda".

Details

Optionally fits λ for each regression model. Likelihood optimization of λ is performed for method="lambda" using optimize on the interval (0,1). This function is redundant with residuals(gls(...,correlation=corPagel(...))) but some users may find this method simpler, and it provides a good way to cross-check your results & make sure that you are using gls correctly.

Value

A list with the following components:

beta	a vector or matrix of regression coefficients.
resid	a vector or matrix of residuals for species.
lambda	a vector of lambda values (method="lambda" only).
logL	a vector of log-likelihoods (method="lambda" only).

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2009) Size-correction and principal components for interspecific comparative studies. *Evolution*, **63**,3258–3268.

Revell, L. J. (2010) Phylogenetic signal and linear regression on species data. *Methods in Ecology and Evolution*, **1**, 319–329.

See Also

[phyl.pca](), [gls]()

---

phyl.RMA   *Phylogenetic reduced major axis (RMA) regression*

Description

This function performs phylogenetic RMA regression.

Usage

```
phyl.RMA(x, y, tree, method="BM", lambda=NULL, fixed=FALSE, h0=1.0)
```

Arguments

x	vector with names.
y	vector with names.
tree	a phylogenetic tree in "phylo" format.
method	method to obtain the correlation structure: can be "BM" or "lambda".
lambda	value of lambda for fixed λ.
fixed	logical value indicating whether or not λ should be optimized using likelihood.
h0	null hypothesis for beta. Defaults to 1.0. Note that a null hypothesis of 0.0 is not allowed.

Details

Optionally jointly estimates lambda if method="lambda". Likelihood optimization of lambda is performed using [optimize]() on the interval (0,1).

Value

A list with the following components:

RMA.beta	a vector of RMA regression coefficients.
V	a VCV matrix for the traits.
lambda	fitted value of lambda (method="lambda" only).
logL	log-likelihood (method="lambda" only).
test	a vector containing results for hypothesis tests on beta.
resid	a vector of residuals for y given x.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[phyl.cca](), [phyl.pca](), [phyl.resid]()

| phyl.vcv | *Compute evolutionary VCV matrix for a tree & dataset* |

Description

Primarily an internal function for [phyl.pca](); this can be used to compute the phylogenetic trait variance-covariance matrix given a phylogenetic VCV, lambda, and a data matrix.

Usage

```
phyl.vcv(X, C, lambda)
```

Arguments

lambda	value for λ transformation.
X	data matrix.
C	matrix containing the height above the root of each pair of species in the tree. Typically this will have been produced by calling [vcv.phylo]().

Details

Do not use unless you know what you're doing.

Value

A matrix.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

phylANOVA   *Phylogenetic ANOVA and post-hoc tests*

Description

This function performs the simulation-based phylogenetic ANOVA of Garland et al. (1993) and (optionally) conducts all posthoc comparisons of means among groups (also obtaining the P-values by phylogenetic simulation).

Usage

```
phylANOVA(tree, x, y, nsim=1000, posthoc=TRUE, p.adj="holm")
```

Arguments

tree	a phylogenetic tree in "phylo" format.
x	a vector containing the groups.
y	a vector containing the response variable (continuously valued).
nsim	an integer specifying the number of simulations (including the observed data).
posthoc	a logical value indicating whether or not to conduct posthoc tests to compare the mean among groups.
p.adj	method to adjust P-values for the posthoc tests to account for multiple testing. Options same as p.adjust.

Details

Uses a little bit of code from phy.anova in the "geiger" package as well as pairwise.t.test.

Value

A list containing the following elements:

F	F from observed data.
Pf	P-value for F from simulation.
T	matrix of t-values.
Pt	matrix of multiple test corrected P-values from posthoc t-tests.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Garland, T., Jr., A. W. Dickerman, C. M. Janis, & J. A. Jones. (1993) Phylogenetic analysis of covariance by computer simulation. *Systematic Biology*, **42**, 265-292.

Harmon, L. J., J. T. Weir, C. D. Brock, R. E. Glor, W. Challenger. (2008) GEIGER: investigating evolutionary radiations. *Bioinformatics*, 24, 129-131.

See Also

[anova](), [pairwise.t.test]()

---

phyloDesign	*Compute design matrix for least squares analyses*

Description

Primarily an internal function for [optim.phylo.ls](), thisi function creates a design matrix for least squares phylogenetic analysis.

Usage

```
phyloDesign(tree)
```

Arguments

tree   phylogenetic tree.

Details

This function returns a matrix containing the edges in the tree (in columns) and pairs of tip node numbers (in rows). Values in the matrix are either 1 if the edge is on the shortest path between the two tips; and 0 otherwise. Probably do not use unless you know what you're doing.

Value

A matrix.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

phylomorphospace	*Creates phylomorphospace plot*

Description

This function creates a phylomorphospace plot (a projection of the tree into morphospace) for two characters following Sidlauskas (2008; *Evolution*). It will also plot a discrete character mapped on tree.

Usage

```
phylomorphospace(tree, X, A=NULL, label=TRUE, control=list(), ...)
```

Arguments

tree	a phylogenetic tree in ″phylo″ format, or a modified ″phylo″ object with a mapped discrete character.
X	an n x 2 matrix of tip values for two characters in n species.
A	an optional m x 2 matrix (for m nodes) of values for two taits at internal nodes in the tree - if not supplied, these values will be estimated using fastAnc.
label	logical value indicating whether to print tip labels next to terminal nodes in the plot.
control	a list containing the following optional control parameters: col.edge: a vector of edge colors; and col.node: a vector of node colors.
...	optional arguments for plotting, including xlim, ylim, xlab, ylab, lwd, colors, fsize, and node.by.map. colors is only used when there is a mapped discrete character on the tree, in which case control$col.edge is ignored. fsize is relative to the default, which is textxy(...,cx=0.75). node.by.map is a logical value (defaults to FALSE which tells the function whether or not to plot the node colors using the colors of the mapped discrete character. Setting this option to TRUE will cause control$col.node to be ignored.

Value

This function creates a phylomorphospace plot

Author(s)

Liam Revell <liam.revell@umb.edu>

References

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

Sidlauskas, B. (2008) Continuous and arrested morphological diversification in sister clades of characiform fishes: A phylomorphospace approach. *Evolution*, **62**, 3135–3156.

Examples

```
tree<-pbtree(n=25)
X<-fastBM(tree,nsim=2)
phylomorphospace(tree,X,xlab="trait 1",ylab="trait 2")
```

---

phylomorphospace3d   *Creates tree-dimensional phylomorphospace plot*

Description

This function creates a phylomorphospace plot for three characters using the 3D visualization package, 'rgl'.

Usage

```
phylomorphospace3d(tree, X, A=NULL, label=TRUE, control=list())
```

Arguments

tree	a phylogenetic tree in ″phylo″ format.
X	an n x 3 matrix of tip values for two characters in n species.
A	an optional m x 3 matrix (for m nodes) of values for two taits at internal nodes in the tree - if not supplied, these values will be estimated using anc.ML.
label	logical value indicating whether to print tip labels next to terminal nodes in the plot (presently doesn't do anything, but labels can be dropped using control). spin=TRUE,axes=TRUE,box=TRUE,simple.axes=FALSE,lwd=1,ftype="reg"
control	a list containing the following optional control parameters: spin: a logical value indicating whether to animate the plot when created; axes: a logical indicating whether to plot the axes; box: a logical value indicating whether to plot in box; simple.axes: logical value indicating whether to replace box and axes with simpler axes; lwd: line widths; ftype: font type ("off" turns off labels altogether).

Value

This function creates a three dimensional phylomorphospace plot

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Sidlauskas, B. (2008) Continuous and arrested morphological diversification in sister clades of characiform fishes: A phylomorphospace approach. *Evolution*, **62**, 3135–3156.

See Also

fancyTree, phenogram, phylomorphospace

Examples

```
tree<-pbtree(n=25)
X<-fastBM(tree,nsim=3)
## Not run:
phylomorphospace3d(tree,X,control=list(spin=FALSE))

## End(Not run)
```

---

phylosig   *Compute phylogenetic signal with two methods*

Description

This function computes phylogenetic signal using two different methods. It can also conduct the hypothesis tests for significant phylogenetic signal, and estimate phylogenetic signal incorporating sampling error following Ives et al. (2007).

Usage

```
phylosig(tree, x, method="K", test=FALSE, nsim=1000, se=NULL, start=NULL,
control=list())
```

Arguments

tree	a phylogenetic tree in "phylo" format.
x	vector containing values for a single continuously distributed trait.
method	method to compute signal: can be "K" or "lambda".
test	logical indicating whether or not to conduct a hypothesis test of "K" or "lambda".
nsim	for method="K", number of simulations in randomization test.
se	named vector containing the standard errors for each species.
start	vector of starting values for optimization of (respectively) sigma^2 and lambda. Only used in method="lambda" and se!=NULL.
control	list of control parameters for multidimensional optimization, implemented in optim. Only used in method="lambda" and se!=NULL.

Details

λ optimization is performed using optimize with the range of lambda set between 0 and the theoretical upper limit of lambda (determined by the relative height of the most recent internal node on the tree).

Value

If (`method="K"`), a list with the following components:

K	value of the K-statistic.
sig2	rate of evolution, σ^2, for estimation with sampling error.
logL	log-likelihood, for estimation with sampling error.
P	optionally, the P-value from the randomization test.

If (`method="lambda"`), a list with the following components:

lambda	fitted value of lambda.
sig2	rate of evolution, for estimation with sampling error.
logL	log-likelihood.
logL0	log-likelihood for `lambda=0.0`.
P	P-value of the likelihood ratio test.
convergence	value for convergence, for estimation with sampling error only. (See `optim`).
message	message from `optim`, for estimation with sampling error only.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Pagel, M. (1999) Inferring the historical patterns of biological evolution. *Nature*, **401**, 877–884.

Blomberg, S. P., T. Garland Jr., A. R. Ives (2003) Testing for phylogenetic signal in comparative data: Behavioral traits are more labile. *Evolution*, **57**, 717–745.

Ives, A. R., P. E. Midford, T. Garland Jr. (2007) Within-species variation and measurement error in phylogenetic comparative biology. *Systematic Biology*, **56**, 252-270.

Examples

```
tree<-pbtree(n=100)
x<-fastBM(tree)
phylosig(tree,x,method="lambda",test=TRUE)
```

---

plotBranchbyTrait *Plot branch colors by a quantitative trait or value*

Description

Function plots a tree with branches colored by the value for a quantitative trait or probability, by various methods. Unlike most other tree plotting functions in phytools, this function calls [plot.phylo](not plotSimmap) internally.

Usage

```
plotBranchbyTrait(tree, x, mode=c("edges","tips","nodes"), palette="rainbow",
legend=TRUE, xlims=NULL, ...)
```

Arguments

tree	an object of class "phylo".
x	either a vector of states for the edges, tips, or nodes of the tree (for mode="edges", "tips", and "nodes", respectively).
mode	string indicating plotting mode. mode="edges", the default, requires that the mapping state of each edge in the tree should be provided. mode="tips" takes the tip values and estimates the state at each internal node. The mapped character value along each branch is the average of the nodes subtending that branch. mode="nodes" similar to "tips", except that the node values are provided instead of estimated.
palette	color palette to translate character values to color. Options are presently "rainbow" (the default), "heat.colors", and "gray".
legend	can be a logical value (TRUE or FALSE) or a numeric value greater than 0. In the latter case the numeric value gives the length of the plotted legend, which also acts as a scale bar for the branch lengths of the tree.
xlims	range for the translation map between trait values and the color map. Should be inclusive of all the values in x.
...	other optional arguments to be passed to [plot.phylo]().

Details

Note that the functin will prompt for the position of the legend. For some reason, the position is required *before* the prompting message!

Value

Plots a phylogeny.

Author(s)

Liam Revell <liam.revell@umb.edu>

`plotSimmap`	*Plot stochastic character mapped tree*

Description

Function plots one or multiple stochastic character mapped trees.

Usage

```
plotSimmap(tree, colors=NULL, fsize=1.0, ftype="reg", lwd=2, pts=TRUE,
node.numbers=FALSE, mar=NULL, add=FALSE, offset=NULL, direction="rightwards",
type="phylogram")
```

Arguments

tree	a modified object of class "phylo" or "multiPhylo" containing a stochastic mapping or set of mappings (e.g., see [read.simmap](#) & [make.simmap](#)).
colors	a vector with names translating the mapped states to colors - see Examples.
fsize	relative font size for tip labels.
ftype	font type - options are ″reg″, ″i″ (italics), ″b″ (bold), or ″bi″ (bold-italics).
lwd	line width for plotting.
pts	logical value indicating whether or not to plot filled circles at each vertex of the tree, as well as at transition points between mapped states. Default is TRUE.
node.numbers	a logical value indicating whether or not node numbers should be plotted.
mar	vector containing the margins for the plot to be passed to [par](#). If not specified, the default margins are [0.1,0.1,0.1,0.1].
add	a logical value indicating whether or not to add the plotted tree to the current plot (TRUE) or create a new plot (FALSE, the default).
offset	offset for the tip labels. Primarily to be used internally by [densityMap](#).
direction	plotting direction. Options are ″rightwards″ (the default) and ″leftwards″. Note that for some reason that is not totally clear, node.numbers=TRUE does not work for direction="leftwards".
type	plot type. Can be ″phylogram″ or ″fan″. Only a subset of options are presently available for type="fan".

Details

The underscore character ″_″ is automatically swapped for a space in tip labels, as in [plot.phylo](#).

Value

Plots a tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Huelsenbeck, J. P., R. Neilsen, and J. P. Bollback. (2003) Stochastic mapping of morphological characters. *Systematic Biology*, **52**, 131-138.

Bollback, J. P. (2006) Stochastic character mapping of discrete traits on phylogenies. *BMC Bioinformatics*, **7**, 88.

See Also

densityMap, make.simmap, read.simmap

Examples

```
# simulate a mapped tree
Q<-matrix(c(-2,1,1,1,-2,1,1,1,-2),3,3)
rownames(Q)<-colnames(Q)<-letters[1:3]
tree<-sim.history(pbtree(n=100,scale=1),Q)
cols<-setNames(c("blue","red","green"),letters[1:3])
# plot the mapping
plotSimmap(tree,cols,ftype="i",fsize=0.7)
```

---

plotThresh   *Tree plotting with posterior probabilities of ancestral states from the threshold model*

Description

This function uses the object returned by ancThresh to plot the posterior probabilities of ancestral states under the threshold model. It is also called internally by ancThresh.

Usage

```
plotThresh(tree, x, mcmc, burnin=NULL, piecol, tipcol="input", ...)
```

Arguments

tree	phylogenetic tree.
x	a named vector containing discrete character states; or a matrix containing the tip species, in rows, and probabilities of being in each state, in columns.
mcmc	list object returned by ancThresh.
burnin	number of generations (not samples) to exclude as burn in; if NULL then 20 percent of generations are excluded as burn-in.
piecol	a named vector containing the colors for the posterior probabilities plotted as pie charts at internal nodes.

tipcol	a string indicating whether the tip colors should be based on the input data ("input") or sampled tip liabilities ("estimated"). These will only differ if there is uncertainty in the tip states.
...	other arguments to be passed to `plot.phylo` - `label.offset` should be >0 so that tip labels and species names do not overlap.

Value

Plots a tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

`ancThresh`, `plot.phylo`

---

plotTree   *Plots rooted phylogenetic tree*

Description

This function plots a rooted phylogram. Arguments in ... are passed to `plotSimmap`, with the exception of optional argument `color` which is used to determine the plotted color of the branch lengths of the tree. Note that the optional argument pts, which is by default pts=TRUE in `plotSimmap` has the default value of pts=FALSE in `plotTree`.

Usage

```
plotTree(tree, ...)
```

Arguments

tree	a phylogenetic tree in "phylo" format; or multiple trees as an object of class "multiPhylo".
...	optional arguments.

Value

This function plots a rooted phylogram.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[plot.phylo](), [plotSimmap]()

Examples

```
tree<-pbtree(n=25)
plotTree(tree,color="blue",ftype="i")
```

---

posterior.evolrate   *Analysis of the posterior sample from evol.rate.mcmc*

Description

This fucntion takes a phylogenetic tree, an average split position, and a raw MCMC output from
evol.rate.mcmc and returns a posterior sample of evolutionary rates rootward ($\sigma(1)^2$) and tipward
($\sigma(2)^2$) from the average split.

Usage

```
posterior.evolrate(tree, ave.shift, mcmc, tips, showTree=FALSE)
```

Arguments

tree	a phylogenetic tree in "phylo" format.
ave.shift	mean or median shift-point from the posterior sample (see [minSplit]().
mcmc	matrix $mcmc from evol.rate.mcmc (probably with burnin excluded).
tips	list of stips in state sig(1)^2 for each sampled generation of MCMC.
showTree	optional logical value indicating whether or not to plot the stretched and shrunken tree generated by the pre-processing algorithm implemented in this function (default is FALSE).

Value

A matrix containing the posterior sample of evolutionary rates and shift-points between rates.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J., D. L. Mahler, P. Peres-Neto, and B. D. Redelings (2012) A new method for identifying
exceptional phenotypic diversification. *Evolution*, **66**, 135-146.

See Also

[evol.rate.mcmc](), [minSplit]()

| ratebystate | *Method for investigating the rate of one trait as a function of the state of another* |

Description

This function attempts to ask if the rate of a continuous character, y, depends on the state of a separate continuous trait, x. This is accomplished by regressing the squared contrasts in y on the branch or node ancestral estimates of x.

Usage

```
ratebystate(tree, x, y, nsim=100, corr=c("pearson","spearman"), ...)
```

Arguments

tree	phylogenetic tree.
x	a continuous character - the dependent variable in the model.
y	a second continuous trait - the response variable.
nsim	number of simulations for hypothesis testing.
corr	correlation method to use. Same as in [cor]().
...	optional arguments which include sim.method ("fastBM" or "sim.corrs"; see [fastBM]() and [sim.corrs]()); method ("by.node" or "by.branch" indicating whether to assume the rate varies as a function of the node state or the mean branch state); message - a logical value indicating whether or not to return corr and method; finally logarithm - indicating whether or not to fit a model in which the variance of Brownian evolution in y changes as a multiplicative function of x. The default is logarithm=FALSE.

Value

This function returns a list with up to the following four elements:

beta	value of the regression coefficient for square of the contrasts in y regressed on the ancestral or branch-wise estimated states for x.
r	correlation coefficient for corr=corr.
corr	string giving the value of corr.
method	string giving the value of method.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[fastAnc](), [pic]()

---

read.newick   *Read Newick style tree*

Description

This function reads a Newick style tree from file.

Usage

```
read.newick(file="", text)
```

Arguments

file	name of text file with single Newick style tree.
text	character string containing tree.

Details

Note, this function does not do anything that `read.tree` doesn't do better - and is included mostly for development purposes.

Value

An object of class `"phylo"`

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

`read.tree`, `read.nexus`

Examples

```
tree<-"((Human,Chimp),Gorilla),Monkey);"
phy<-read.newick(text=tree)
```

---

read.simmap	*Read SIMMAP style trees from file*

Description

This reads one or multiple SIMMAP style trees from file.

Usage

```
read.simmap(file="", text, format="nexus", rev.order=TRUE, version=1)
```

Arguments

file	name of text file with one or multiple SIMMAP v1.0 or v1.5 style trees.
text	character string containing the tree. If version=1.5 this argument is ignored. (This format tree can only be read from file in the present version.)
format	format of the trees: either ″phylip″ or ″nexus″ - the latter is the default output from SIMMAP. If version=1.5 this argument is ignored.
rev.order	a logical value indicating whether the states and times along each branch is given (from root to tip) in right-to-left order (if TRUE) or in left-to-right order. If version=1.5 this argument is ignored.
version	version of SIMMAP for input tree. If the tree(s) was/were simulated in SIMMAP v1.0 or written to file by link{make.simmap} then version=1.0; if the tree(s) was/were simulated using SIMMAP v1.5 then version=1.5.

Details

This function now accepts trees in both SIMMAP v1.0 and SIMMAP v1.5 format. In addition, it can read a more flexible format than is produced by SIMMAP (for instance, multi-character mapped states and more than 7 mapped states). Uses some modified code from read.nexus from the "ape" package to read the NEXUS block created by SIMMAP. Also creates the attribute ″map.order″ which indicates whether the stochastic map was read in from left to right or right to left. This attribute is used by default by write.simmap to write the tree in the same order.

Value

A modified object of class ″phylo″ (or list of class ″multiPhylo″) with the following additional elements:

maps	a list of named vectors containing the times spent in each state on each branch, in the order in which they occur.
mapped.edge	a matrix containing the total time spent in each state along each edge of the tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

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

Bollback, J. P. (2006) Stochastic character mapping of discrete traits on phylogenies. *BMC Bioinformatics*, **7**, 88.

See Also

[brownie.lite](), [evol.vcv](), [read.tree](), [read.nexus]()

---

reorderSimmap   *Reorder edges of a simmap tree*

Description

Function returns a reordered modified "phylo" object by using reorder.phylo but then sorting the additional elements $mapped.edge and $maps to have the same order as $edge.

Usage

```
reorderSimmap(tree, order="cladewise")
```

Arguments

tree   a modified object of class "phylo".

order   either "cladewise" or "pruningwise" (see [reorder.phylo]()).

Value

A modified object of class "phylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[reorder.phylo](), [plotSimmap]()

---

repPhylo | *Replicate a tree into a list of trees*

Description

This function is functionally equivalent to [rep](), except that it works for phylogenies. Will create an object of class "multiPhylo".

Usage

```
repPhylo(tree, times)
```

Arguments

tree   object of class "phylo".

times   number of times to replicate tree.

Value

An object of class "multiPhylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[c.phylo](), [rep]()

Examples

```
tree<-pbtree(n=100)
trees<-repPhylo(tree,100)
```

---

reroot | *Re-root a tree along an edge*

Description

This function re-roots a phylogenetic tree at an arbitrary position along an edge.

Usage

```
reroot(tree, node.number, position)
```

Arguments

tree   a phylogenetic tree in ″phylo″ format.

node.number   number of the node descending from the target branch in tree$edge - this can
                     also be a tip in which case the node number is the index number of the tip in
                     tree$tip.label.

position   position along the target edge at which to re-root the tree..

Details

This function had an error for rootings along edges descended from the root node for phytools<=0.2-
47. This should be fixed in the present version. Now uses paste.tree, root, and splitTree
internally.

Value

A phylogenetic tree in ″phylo″ format.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

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

See Also

splitTree, paste.tree, root

---

rerootingMethod   *Get marginal ancestral state reconstructions by re-rooting*

Description

This function uses the re-rooting method of Yang et al. (1995) to get the marginal ancestral state
estimates for each internal node of the tree using likelihood. This method get the conditional scaled
likelihoods for the root node (which is the same as the marginal ancestral state reconstruction for
that node) and successively moves the root to each node in the tree. The function can also return
the posterior probabilities for the tip nodes of the tree.

Usage

```
rerootingMethod(tree, x, model=c("ER","SYM"), ...)
```

Arguments

tree	an object of class "phylo".
x	a vector of tip values for species, or a matrix containing the prior probability that the tip is in each state. If x is a vector, then names(x) should be the species names. If x is a matrix of prior probabilities, then rownames should be species names, column names should be states for the discrete character, and rows of the matrix should sum to 1.0.
model	any revsersible model. model=c("ER","SYM") recommended.
...	optional arguments. Presently the logical argument tips. If tips=TRUE, then the function will also compute the empirical Bayes posterior probabilities of the tips following Yang (2006).

Details

This function calls code modified from ace in the (Paradis et al. 2004) internally in the calculation of normalized conditional likelihoods.

Value

A list containing the following elements:

loglik	the log-likelihood.
Q	the fitted transition matrix between states.
marginal.anc	the marginal ancestral state reconstructions for each node (and, optionally, each tip).

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

ace, make.simmap

---

rescaleSimmap *Rescale SIMMAP style tree*

Description

This function scales a tree with a mapped discrete character to an arbitrary total height, preserving the relative time spent in each state along each edge.

Usage

```
rescaleSimmap(tree, totalDepth=1.0)
```

Arguments

tree   a phylogenetic tree in modified "phylo" format with a discrete character mapping (e.g., see read.simmap or make.simmap).

totalDepth   a new desired value for the total depth (height) of the tree. If the tree is not ultrametric this is the maximum height.

Details

Replaces rescaleTree in the 'geiger' package for SIMMAP style trees.

Value

A phylogenetic tree in modified "phylo" format.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

make.simmap, read.simmap

---

roundBranches   *Rounds the branch lengths of a tree*

Description

This function rounds the branch lengths of a tree and reconciles any mappings (as in read.simmap) with the rounded branch lengths.

Usage

```
roundBranches(tree, digits)
```

Arguments

tree   tree.

digits   number of digits for rounding. Passed to round.

Value

A tree with branch lengths, or modified "phylo" object with a mapped discrete character.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

rstate | *Pick a random state according to a vector of probabilities*

Description

Primarily an internal function for `make.simmap`.

Usage

```
rstate(y)
```

Arguments

y   vector of probabilities. Must have names & should probably add to `1.0`.

Details

This function picks a random element in a vector according to the probability assigned that element. It returns the name. Uses `rmultinom`.

Value

A character or string.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

sampleFrom | *Sample from a set of distributions*

Description

Function samples from a set of normal distributions with parameters given in xbar and xvar.

Usage

```
sampleFrom(xbar=0, xvar=1, n=1, randn=NULL, type="norm")
```

Arguments

xbar   a named vector of means.

xvar   a named vector of variances.

n   a vector containing the sample sizes of each species.

randn   a range of sample sizes are to be random.

type   `"norm"` is the only distribution implemented so far.

Value

A vector, with labels.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

sim.corrs   *Multivariate Brownian simulation with multiple correlations and rates*

Description

This function conducts BM simulation on a tree with multiple rates and/or multiple evolutionary correlations between characters. If vcv is a single matrix, instead of a list of matrices, sim.corrs will simulate multivariate BM with a single rate matrix.

Usage

```
sim.corrs(tree, vcv, anc=NULL, internal=FALSE)
```

Arguments

tree	is a phylogenetic tree in 'phylo' format; or a modified 'phylo' tree with a mapped discrete character.
vcv	is a square covariance matrix or named list of matrices (one for each mapped state on the tree).
anc	optional vector of values for the root state.
internal	logical value indicating whether to return states at internal nodes.

Value

A matrix containing the multivariate tip states for the n species in the tree (and nodes if internal=TRUE).

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[fastBM](), [make.simmap](), [read.simmap](), [sim.history](), [sim.rates]()

---

sim.history   *Simulate stochastic character history under some model*

Description

This function simulates a stochastic character history for a discretely valued character trait on the tree. The resultant tree is stored as a modified ″phylo″ object in stochastic character map (e.g., `make.simmap`) format.

Usage

```
sim.history(tree, Q, anc=NULL, nsim=1)
```

Arguments

tree	a phylogenetic tree as an object of class ″phylo″.
Q	a matrix containing the instantaneous transition rates between states.
anc	an optional value for the state at the root node; if NULL, a random state will be assigned.
nsim	number of simulations.

Value

A modified phylogenetic tree of class ″phylo″ (or a modified ″multiPhylo″ object, for nsim > 1) with the following additional elements:

maps	a list of named vectors containing the times spent in each state on each branch, in the order in which they occur.
mapped.edge	a matrix containing the total time spent in each state along each edge of the tree.
states	a vector containing the tip states.
node.states	a matrix containing the states at internal & terminal nodes (according to the dimensions of edge).

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

`make.simmap`, `read.simmap`, `plotSimmap`, `sim.rates`

---

sim.ratebystate   *Conduct simulation of state dependent rate variation*

Description

This function attempts to simulate two characters under a model in which the rate of evolution for the second (y) depends on the states for the first x. See [ratebystate](#) for more details.

Usage

```
sim.ratebystate(tree, sig2x=1, sig2y=1, beta=c(0,1), ...)
```

Arguments

tree   phylogenetic tree.

sig2x   variance of the Brownian process of evolution for x.

sig2y   variance of the Brownian process of evolution for y when x-min(x)==1 (for
              logarithm=FALSE) or x==0 (for logarithm=TRUE).

beta   intercept and slope of the relationship between the value of x and the Brownian
              rate in y.

...   optional arguments which include method ("by.node" or "by.branch" indi-
              cating whether to assume the rate varies as a function of the node state or the
              mean branch state); plot, a logical value indicating whether or not to plot a
              phenogram with the branches used for simulation of y after rescaling by the state
              of x; and logarithm, a logical value indicating whether or not simulate changes
              in the variance of Brownian evolution for y as an additive logarithm=FALSE or
              multiplicative function of x. The default is logarithm=FALSE.

Value

This function returns a matrix.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[fastBM](#), [ratebystate](#)

---

sim.rates | *Brownian simulation with multiple evolutionary rates*

Description

This function conducts BM simulation on a tree with multiple rates.

Usage

```
sim.rates(mtree, sig2, anc=0, nsim=1, internal=F, plot=F)
```

Arguments

mtree   is a stochastic map format phylogenetic tree in modified ″phylo″ format (e.g.,
            see make.simmap).

sig2   a named vector containing the rates for each state; names should be states in
            mtree.

anc   optional value for the root state.

nsim   number of simulations.

internal   logical value indicating whether to return states at internal nodes.

plot   logical value indicating whether or not to visual the rate heterogeneity (default
            value is FALSE.

Value

A vector (for nsim=1) or matrix containing the tip states for the n species in the tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

fastBM, make.simmap, read.simmap, sim.history

---

splitplotTree   *Plots a phylogeny in two columns*

Description

Function plots a tree in two columns or windows.

Usage

```
splitplotTree(tree, fsize=1.0, ftype="reg", lwd=2, split=NULL, new.window=FALSE)
```

Arguments

tree	an object of class "phylo".
fsize	relative font size for tip labels.
ftype	font type - options are "reg", "i" (italics), "b" (bold), or "bi" (bold-italics).
lwd	line width for plotting.
split	relative vertical position for splitting the tree (between 0 & 1).
new.window	whether or not to plot the split tree in a new window. If FALSE then the tree will be plotted in two columns within the same plotting window.

Value

Plots a tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[plotTree](), [plotSimmap]()

---

splitTree   *Split tree at a point*

Description

Primarily an internal function for [posterior.evolrate](), this function splits the tree at a given point, and returns the two subtrees as an object of class "multiPhylo".

Usage

```
splitTree(tree, split)
```

Arguments

`tree`	phylogenetic tree.
`split`	split encoded as a list with two elements: node: the node number tipward of the split; and bp: the position along the branch to break the tree, measured from the rootward end of the edge.

Details

Probably do not use this unless you can figure out what you are doing.

Value

Two trees in a list.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

starTree	*Create star phylogeny*

Description

This function creates a star phylogeny.

Usage

```
starTree(species, branch.lengths=NULL)
```

Arguments

`species`	a list of species.
`branch.lengths`	an optional list of branch lengths in the same order as `species`.

Details

Creates a star phylogeny with (optionally) user specified branch lengths.

Value

An object of class ″phylo″.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

stree

---

strahlerNumber   *Computes Strahler number for trees and nodes*

Description

The function strahlerNumber computes the Strahler number of all nodes and tips in the tree. For more information about Strahler numbers see http://en.wikipedia.org/wiki/Strahler_number. The function extract.strahlerNumber extracts all of the most inclusive clades of Strahler number i.

Usage

```
strahlerNumber(tree, plot=TRUE)
extract.strahlerNumber(tree, i, plot=TRUE)
```

Arguments

tree	an object of class "phylo".
i	order of Strahler number to extract for extract.strahlerNumber.
plot	logical value indicating whether to plot the tree with Strahler numbers for node labels.

Value

Either a vector with the Strahler number for each tip and internal node; or (for extract.strahlerNumber the set of (most inclusive) subtrees with Strahler number i as an object of class "multiPhylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

---

threshBayes   *Threshold model using Bayesian MCMC*

Description

This function uses Bayesian MCMC to fit the quantitative genetics threshold model (Felsenstein 2012) to data for two discrete characters or one discrete and one continuous character.

Usage

```
threshBayes(tree, X, types=NULL, ngen=1000, control=list())
```

Arguments

`tree`	an object of class `"phylo"`.
`X`	a numeric matrix containing values for a numerically coded discrete character and a continuous character; or two discrete characters. The row names of `X` should be species names.
`types`	a vector of length `ncol(X)` containing the data types for each column of `X`, for instance `c("discrete","continuous")`.
`ngen`	a integer indicating the number of generations for the MCMC.
`control`	a list of control parameters for the MCMC. Control parameters include: `sample`, the sampling interval for the MCMC; `propvar`, a vector containing (in this order) proposal variances for the two rates (if the type is `"discrete"` this will be ignored), the two ancestral states, and the correlation; `propliab`, a single proposal variance for the liabilities; `pr.mean`, a vector for the mean of the prior probability distributions for each parameter, in the same order as `propvar`; `pr.liab`, currently ignored; `pr.var`, a vector with variances for the prior densities for each parameter, in the same order as `pr.mean` - note that for the rates we use an exponential distribution so the first two means are currently ignored; and `pr.vliab` currently ignored.

Value

This function returns a list with two elements: `par` a matrix containing the posterior sample for the model parameters (evolutionary rates, ancestral states, and correlation); `liab` a matrix containing the posterior sample of the liabilities. For continuous characters, the liabilities are treated as known and so the posterior samples are just the observed values.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[anc.Bayes](), [bmPlot](), [evol.rate.mcmc]()

---

threshDIC	*Deviance Information Criterion from the threshold model*

Description

This function computes the Deviance Information Criterion from the MCMC object returned by [ancThresh]().

Usage

```
threshDIC(tree, x, mcmc, burnin=NULL, sequence=NULL, method="pD")
```

Arguments

tree	phylogenetic tree.
x	a named vector containing discrete character states; or a matrix containing the tip species, in rows, and probabilities of being in each state, in columns.
mcmc	list object returned by [ancThresh](#).
burnin	number of generations (not samples) to exclude as burn in; if not supplied then 20 percent of generations are excluded.
sequence	assumed ordering of the discrete character state. If not supplied and x is a vector then numerical-alphabetical order is assumed; if not supplied and x is a matrix, then the column order of x is used.
method	method for computing the effective number of parameters (options are ″pD″ and ″pV″).

Value

A vector containing the mean deviance and deviance for the parameter means, the effective number of parameters, and the DIC.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[ancThresh](#)

---

threshState	*Computes value for a threshold character from a liability and thresholds*

Description

Primarily to be used internally by [ancThresh](#); can also be used to simulate threshold traits.

Usage

```
threshState(x, thresholds)
```

Arguments

x	liability.
thresholds	a named vector containing the thresholds.

Value

A discrete character value.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[ancThresh](), [threshDIC]()

---

to.matrix	*Convert a character vector to a binary matrix*

Description

This function takes a vector of characters and computes a binary matrix. Primarily to be used internally by [make.simmap]() and [rerootingMethod]().

Usage

```
to.matrix(x, seq)
```

Arguments

x	a vector of characters.
seq	the sequence for the columns in the output matrix.

Value

A binary matrix of dimensions length(x) by length(seq).

Author(s)

Liam Revell <liam.revell@umb.edu>

---

treeSlice	*Slices the tree at a particular point and returns all subtrees*

Description

This function slices a tree at a particular height above the root and returns all subtrees or all non-trivial subtrees (i.e., subtrees with more than 1 taxon). Uses [extract.clade]() in the "ape" package.

Usage

```
treeSlice(tree, slice, trivial=FALSE)
```

Arguments

tree	is a phylogenetic tree in "phylo" format.
slice	a real number indicating the height above the root at which to slice the tree.
trivial	a logical value indicating whether or not to return subtrees with a number of tips less than two (default is FALSE).

Value

An object of class "phylo" or "multiPhylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[extract.clade](#)

---

untangle	*Attempts to untangle crossing branches for plotting*

Description

This function attempts to untangle the branches of a tree that are tangled in plotting with [plot.phylo](#), [plotTree](#), or [plotSimmap](#). Note that method="read.tree" does not presently work for SIMMAP style trees

Usage

```
untangle(tree, method=c("reorder","read.tree"))
```

Arguments

tree	tree as an object of class "phylo". Can be a SIMMAP style tree (e.g., [read.simmap](#).
method	method to use to attempt to untangle branches. method="reorder" uses two calls of [reorder.phylo](#) or [reorderSimmap](#); method="read.tree" writes the tree to a text string and then reads it back into memory using [read.tree](#).

Value

A tree with branch lengths, or modified "phylo" object with a mapped discrete character.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

vcvPhylo   *Calculates cophenetic (i.e., phylogenetic VCV) matrix*

Description

This function returns a so-called *phylogenetic variance covariance matrix* (e.g., see `vcv.phylo`), but (optionally) including ancestral nodes.

Usage

```
vcvPhylo(tree, anc.nodes=TRUE)
```

Arguments

tree	object of class `"phylo"`.
anc.nodes	logical value indicating whether or not to include ancestral nodes.

Value

A matrix.

Author(s)

Liam Revell <liam.revell@umb.edu>

---

write.simmap   *Write a stochastic character mapped tree to file*

Description

This function writes stochastic character mapped trees to file using the Newick style format of SIMMAP v1.0 (Bollback 2006). Note, can only write one tree at a time to file (hence the append option).

Usage

```
write.simmap(tree, file=NULL, append=FALSE, map.order=NULL)
```

Arguments

tree	a phylogenetic tree as a modified object of class `"phylo"`. See `make.simmap` and `read.simmap`.
file	an optional filename.
append	a logical value indicating whether to append to file.
map.order	a optional value specifying whether to write the map in left-to-right or right-to-left order. Acceptable values are "left-to-right" or "right-to-left" or some abbreviation of either. If not provided, `write.simmap` will use `attr(tree,"map.order")` if available.

Value

A file or string (if `file=NULL`).

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Huelsenbeck, J. P., R. Neilsen, and J. P. Bollback (2003) Stochastic mapping of morphological characters. *Systematic Biology*, **52**, 131-138.

Bollback, J. P. (2006) Stochastic character mapping of discrete traits on phylogenies. *BMC Bioinformatics*, **7**, 88.

See Also

[make.simmap](#), [read.simmap](#), [plotSimmap](#)

Examples

```
# simulate a tree & data
tree<-sim.history(pbtree(n=100,scale=1),Q=matrix(c(-1,1,1,-1),2,2))
# generate stochastic character maps
mtrees<-make.simmap(tree,tree$states,nsim=20)
# write them to file
## Not run:
lapply(mmtrees,write.simmap,file="treefile.tre",append=TRUE)

## End(Not run)
```

---

writeAncestors	*Write a tree to file with ancestral states and (optionally) CIs at nodes*

Description

This function writes a tree to file with ancestral character states and (optionally) 95-percent confidence intervals stored as node value..

Usage

```
writeAncestors(tree, Anc=NULL, file="", digits=6, format=c("phylip","nexus"), ...)
```

Arguments

tree	a phylogenetic tree or set of trees as an object of class "phylo" or "multiPhylo".
Anc	a vector of ancestral states, a list containing the ancestral states and 95-percent confidence intervals (as from [fastAnc](#) or [ace](#), or a list of such results.
file	an optional string with the filename for output.
digits	an integer indicating the number of digits to print for branch lengths and ancestral character values.
format	a string indicating whether to output the result in simple Newick (i.e., "phylip") or Nexus format.
...	additional arguments including x: a vector of character values, in which case ancestral states are estimated internally using fastAnc; and CI: a logical value indicating whether or not to estimate 95-percent confidence intervals.

Value

A file, string, or vector of strings.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[ace](#), [fastAnc](#), [write.tree](#)

---

writeNexus	*Write a tree to file in Nexus format*

Description

This function writes one or multiple phylogenetic trees to file in NEXUS format. Redundant with ape::[write.nexus](#).

Usage

```
writeNexus(tree, file="")
```

Arguments

tree	object of class "phylo" or "multiPhylo".
file	file name for output.

Value

Trees written to file.

Author(s)

Liam Revell <liam.revell@umb.edu>

See Also

[write.simmap](), [write.nexus]()

---

xkcdTree   *Plot xkcd style tree*

Description

Function plots a tree in the style of http://xkcd.com (for instance, http://xkcd.com/867/).

Usage

```
xkcdTree(tree, file=NULL, gsPath=NULL, fsize=2, lwd=7, color="blue",
    dim=c(8.5,11), jitter=0.001, waver=c(0.1,0.1), tilt=0, right=TRUE)
```

Arguments

tree	an object of class "phylo".
file	an output file name. If the ending ".pdf" is not provided, it will be automatically appended.
gsPath	for Windows, the full path to the Ghostscript executable. If not provided, the default for gs9.06 is used.
fsize	font size.
lwd	line width.
color	color for lines.
dim	vector of horizontal and vertical dimensions for the output PDF.
jitter	amount of stochasticity ("jitter") for plotting - higher is more jittery.
waver	a vector with approximate horizontal and vertical wavelength for plotting - lower is more wavery.
tilt	declination from horizontal (or vertical, if right=FALSE for tip labels, in degrees.
right	logical indicating whether to plot the tree with rightward orientation (if TRUE) or upwards.

Details

Uses package "extrafont" and font "xkcd.ttf". Before using, install package "extrafont" and add font "xkcd.ttf". Hopefully, if you have not done that, a sensible error will still be returned.

Value

plots a tree to PDF.

Author(s)

Liam Revell <liam.revell@umb.edu>; an anonymous user also contributed much of the code for internally called function xkcdLine.

See Also

[fancyTree](#), [plot.phylo](#)

Index