Package ‘phytools’

September 1, 2022

Version 1.2-0

Date 2022-09-01

Title Phylogenetic Tools for Comparative Biology (and Other Things)

Author Liam J. Revell

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

Depends R (>= 3.5.0), ape (>= 4.0), maps

Imports clusterGeneration, coda, combinat, expm, graphics, grDevices, MASS, methods, mnormt, nlme, numDeriv, optimParallel, parallel, phangorn (>= 2.3.1), plotrix, scatterplot3d, stats, utils

Suggests animation, geiger, RColorBrewer, rgl

ZipData no

Description A wide range of functions for phylogenetic analysis - concentrated in phylogenetic comparative biology, but also including numerous methods for visualizing, manipulating, reading or writing, and even inferring phylogenetic trees. Included among the functions in phylogenetic comparative biology are various for ancestral state reconstruction, model-fitting, and simulation of phylogenies and data, for continuous, discrete, and multivariate characters. A broad range of plotting methods for phylogenies and comparative data include, but are not restricted to, methods for mapping trait evolution on trees, for projecting trees into phenotypic space or a geographic map, and for visualizing correlated speciation between trees. Finally, numerous functions are designed for reading, writing, analyzing, inferring, simulating, and manipulating phylogenetic trees and comparative data. For instance, there are functions for randomly or non-randomly attaching species or clades to a phylogeny, for computing consensus phylogenies from a set, for simulating trees and phylogenetic data under a range of models, and for a wide variety of other manipulations and analyses that phylogenetic biologists might find useful in their research.

URL `https://github.com/liamrevell/phytools`

Repository CRAN

Date/Publication 2022-09-01 17:30:02 UTC

NeedsCompilation no

R topics documented:

`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)`.

The *phytools* development page is `https://github.com/liamrevell/phytools/`. More information on *phytools* can also be found at `http://blog.phytools.org` or `http://www.phytools.org`.

If you use *phytools* (or other packages that depend on *phytools*) in a publication, please *cite it*. The appropriate citation for *phytools* is listed below or can be obtained using `citation("phytools")` with the package installed.

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.

`add.arrow` *Add an arrow pointing to a tip or node on the tree*

Description

This function adds an arrow or a set of arrows to a plotted tree.

Usage

```
add.arrow(tree=NULL, tip, ...)
```

Arguments

`tree`	an object of class `"phylo"`, `"contMap"`, or `"densityMap"`. If not supplied, the function will obtain the last plotted phylogeny from the environmental variable `last_plot.phylo`.
`tip`	label of tip or tip or node number; or vector of such values. If `tree=NULL` then the tip label(s) or node number(s) must be supplied.
`...`	optional arguments to control the shape and size of the arrow including: its length (`arrl`) in the units of the plot; the length of the arrowhead (`hedl`); the total angle between the wings in the arrowhead (`angle`); the line width for the plotted lines (`lwd`); the offset from the tip or end of tip label, in character widths (`offset`); and the color (`col`).

Details

This function presently works for radial (`type="fan"`) and right facing square phylograms (`type="phylogram"`). Trees can be plotted using phytools function `plotTree`, `plotSimmap`, `contMap`, `densityMap`, and ape S3 method `plot.phylo`.

Author(s)

Liam 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.

See Also

`nodelabels`

Examples

```
## show arrows with a black outline
data(anoletree)
plotTree(anoletree,type="fan",fsize=0.7,ftype="i")
add.arrow(anoletree,tip=c("cuvieri","krugi",
    "pulchellus","poncensis","stratulus",
    "evermanni","cooki","cristatellus",
    "gundlachi","occultus"),lwd=5,arrl=1)
add.arrow(anoletree,tip="cuvieri",col="green",
    lwd=3,arrl=1)
add.arrow(anoletree,tip=c("krugi","pulchellus",
    "poncensis"),col="#E4D96F",lwd=3,arrl=1)
add.arrow(anoletree,tip=c("stratulus","evermanni"),
    col="darkgreen",lwd=3,arrl=1)
add.arrow(anoletree,tip=c("cooki","cristatellus",
    "gundlachi"),col="brown",lwd=3,arrl=1)
add.arrow(anoletree,tip="occultus",col="darkgrey",
    lwd=3,arrl=1)
legend(x="topleft",c("crown-giant","grass-bush","trunk-crown","trunk-ground",
    "twig"),pch=22,pt.bg=c("green","#E4D96F","darkgreen",
    "brown","darkgrey"),cex=0.9,
    pt.cex=2,title="PR ecomorphs",bty="n")
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margins to default
```

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

Description

This function adds a color bar to a plot created by `plotBranchbyTrait`. A color bar can be added by clicking on a location within the plot (when `prompt=TRUE`) or by setting `prompt=FALSE` and supplying x & y coordinates for the object. This function is also used internally by S3 methods `plot.contMap` and `plot.densityMap`, as well as by `errorbar.contMap`.

Usage

```
add.color.bar(leg, cols, title=NULL, lims=c(0,1), digits=1, prompt=TRUE,
   lwd=4, outline=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.
`lwd`	width of the plotted bar.
`outline`	logical value indicated whether or not to outline the plotted color bar with a 1 pt line.
`...`	optional arguments including: `x` x-coordinate of the legend (if `prompt=FALSE`); `y` y-coordinate of the legend; `subtitle` optional legend subtitle; `direction` direction of the color bar (i.e., increase from left to right or from right to left); and `fsize`, font size for the legend text.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Revell, L. J. (2013) Two new graphical methods for mapping trait evolution on phylogenies. *Methods in Ecology and Evolution*, **4**, 754-759.

See Also

`contMap`, `densityMap`, `errorbar.contMap`, `plotBranchbyTrait`

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

Description

This function adds a tip to all branches of 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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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)`. If `tips` is specified, then `n` is also ignored, regardless of its value.
`tips`	a set of tip names for the added tips. If `NULL`, names will be supplied
`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>`

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`allFurcTrees`, `add.everywhere`

`add.simmap.legend` *Add legend to stochastically mapped tree*

Description

This function adds a legend (by default, interactively) to a plotted stochastic character mapped tree.

Usage

```
add.simmap.legend(leg=NULL, colors, prompt=TRUE, vertical=TRUE, ...)
```

Arguments

`leg`	states for the discrete character in the order of `colors`.
`colors`	colors for the legend in the order of `leg`, or, if `leg=NULL`, named vector of colors in which `names(colors` are the states of the mapped discrete character.
`prompt`	logical value indicating whether the location of the legend should be obtained interactively (i.e., by clicking in the plotting area).
`vertical`	logical value indiciating whether to plot the legend vertically (if `TRUE`) or horizontally.
`...`	optional arguments including: `x` x-coordinate of the legend (if `prompt=FALSE`); `y` y-coordinate of the legend; and `shape` which can be `shape="square"`, the default, or `shape="circle"`.

Author(s)

Liam 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.

See Also

`plotSimmap`

`add.species.to.genus` *Add species to genus on a phylogeny or bind simulated species subtrees to a backbone genus tree*

Description

`add.species.to.genus` adds an additional species to a genus on a phylogeny.

`genus.to.species.tree` simulates pure-birth subtrees and then binds them at a random height along the terminal edge leading to each corresponding genus on a genus-level backbone tree.

Usage

```
add.species.to.genus(tree, species, genus=NULL, where=c("root","random"))
genus.to.species.tree(tree, species)
```

Arguments

`tree`	an object of class `"phylo"`. In the case of `genus.to.species.tree` this should be a genus-level backbone tree.
`species`	string contain species name in the format `"Genus_species"` or `"Genus species"`.
`genus`	for `add.species.to.genus`, optional argument containing the genus to which `species` is to be attached. If `NULL` then `genus` will be extracted from `species`.
`where`	for `add.species.to.genus`, the location to attach `species` to the tree. `where="root"` will cause the species to be attached to the MRCA of all members of `genus`. `where="random"` will cause `species` to be attached at random to the subtree descended from the MRCA of all members of `genus`.

Details

For `add.species.to.genus`, if `genus` contains only one species and `where="root"`, then `species` will be attached midway along the branch leading to the one species. If `where="random"` then `species` will be added at a random position along the edge. If `genus` cannot be found in the tree, then the original tree is returned and a warning printed. If the tree is not ultrametric, then the resultant tree may not contain branch lengths and a warning will be printed. Note that for some cases in which a tree is read in from file, R may initially think it is ultrametric, but then later (as tips are added) decide that it is not due to rounding of the edge lengths when it was written to file. This can most likely be resolved by using `force.ultrametric` to coerce the tree to be exactly ultrametric (to the maximum numerical precision permitted by R) before adding tips to the tree. If `genus` is non-monophyletic then `species` will be attached to the most inclusive group containing members of `genus` and a warning will be printed.

Value

An object of class `"phylo"`.

Author(s)

Liam 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.

See Also

`add.random`, `bind.tip`

`aic.w` *Computes AIC weights*

Description

This function computes AIC weights for a set of fitted models. It returns an object of class `"aic.w"` which is just a vector which allows it to be automatically printed with a numerical precision of 8 significant digits.

Usage

```
aic.w(aic)
```

Arguments

`aic`	vector of AIC values for different fitted models. If the vector has names, these names will be inherited by the vector returned by the function.

Value

A vector of AIC weights.

Author(s)

Liam 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.

`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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`add.everywhere`, `exhaustiveMP`

Examples

```
## compute & plot all bi- and multifurcating
## trees for six taxa
trees<-allFurcTrees(n=6)
par(mfrow=c(16,15))
nulo<-sapply(trees,plot,type="unrooted",
    no.margin=TRUE)
par(mfrow=c(1,1))
```

`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(), ...)
## S3 method for class 'anc.Bayes'
plot(x, ...)
## S3 method for class 'anc.Bayes'
density(x, ...)
```

Arguments

`tree`	an object of class "`phylo`".
`x`	a vector of tip values for species; `names(x)` should be the species names. In the case of the `plot` and `density` methods, an object of class "`anc.Bayes`".
`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.
`...`	optional arguments, including to be passed to `plot` and `density` methods.

Details

`plot.anc.Bayes` generates a likelihood profile plot of the MCMC by default, but can also create a profile plot for any of the sampled variables by using the optional argument `what`. For instance, `what=40` (or, equivalently, `what="40"`) will create a profile plot of the MCMC for node `40`. Additional arguments are passed to `plot`.

`density.anc.Bayes` computes a posterior density from the MCMC sample. Like `plot.anc.Bayes` takes the optional argument `what`, but unlike `plot.anc.Bayes` computes the posterior density for the root node by default. The object computed by this function is of class `"density"` and can be visualized using `plot.density`. Burn-in (in generations) can be set using the optional argument `burnin`, otherwise it will be assumed to be 20

The `print` and `summary` methods for this object class also return (invisibly) a vector of estimated ancestral states based on a user-supplied burn-in (or 20% of the number of generations of MCMC, if no burn-in is provided). Burn-in can be specified with the optional argument `burnin`.

Value

`anc.Bayes` returns an object of class `"anc.Bayes"` including at least two components:

`mcmc`	a data frame with 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.
`tree`	our input phylogeny.

Author(s)

Liam 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.

See Also

`ace`, `anc.ML`, `anc.trend`, `evol.rate.mcmc`, `fastAnc`

Examples

```
## set seed
set.seed(77)
## load data from Garland et al. (1992)
data(mammal.tree)
data(mammal.data)
## extract character of interest
ln.bodyMass<-log(setNames(mammal.data$bodyMass,
    rownames(mammal.data)))
## run MCMC (should be run at least 1e6 generations)
mcmc<-anc.Bayes(mammal.tree,ln.bodyMass,
    ngen=50000)
```

```
print(mcmc,printlen=20) ## estimates
par(mfrow=c(2,1))
plot(mcmc,bty="l",main="Likelihood-profile from MCMC",
    font.main=3) ## likelihood-profile
plot(density(mcmc,what=Ntip(mammal.tree)+1,
    burnin=20000),bty="l",
    main="Posterior density for root state of log(body mass)",
    font.main=3)
par(mfrow=c(1,1)) ## reset par to default
```

`anc.ML` *Ancestral character estimation using likelihood*

Description

This function estimates the evolutionary parameters and ancestral states for Brownian evolution using likelihood. It is also possible (for `model="BM"`) to allow for missing data for some tip taxa.

Usage

```
anc.ML(tree, x, maxit=2000, model=c("BM","OU","EB"), ...)
```

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.
`model`	model of continuous character evolution ont he tree. It's possible that only `model="BM"` & `model="EB"` work in the present version as `model="OU"` has not be thoroughly tested & some bugs were reported for an earlier version.
`...`	other arguments.

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`. For `model="EB"` this should also not be a problem as the numerical optimization is performed for only `sig2` and `r`, while the ML values of the ancestral states are obtained during every iteration of the optimization algorithmically using the re-rooting method.

Value

An object of class `"anc.ML"` with at least the following four elements (if not more, depending on `model`):

`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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Schluter, D., Price, T., Mooers, A. O., and Ludwig, D. (1997) Likelihood of ancestor states in adaptive radiation. *Evolution* **51**, 1699-1711.

See Also

`ace`, `anc.Bayes`, `fastAnc`, `optim`

Examples

```
## load data from Garland et al. (1992)
data(mammal.tree)
data(mammal.data)
## extract character of interest
ln.bodyMass<-log(setNames(mammal.data$bodyMass,
    rownames(mammal.data)))
## estimate ancestral state under BM model
fit.BM<-anc.ML(mammal.tree,ln.bodyMass)
print(fit.BM)
```

`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 optimization.

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

An object of class `"anc.trend"` 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, σ^2.
`logL`	the log-likelihood.
`convergence`	the value of `$convergence` returned by `optim()` (0 is good).

Author(s)

Liam 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.

See Also

`ace`, `anc.Bayes`, `anc.ML`, `optim`

Examples

```
## simulate tree & data using fastBM with a trend (m!=0)
tree<-rtree(n=26,tip.label=LETTERS)
x<-fastBM(tree,mu=4,internal=TRUE)
a<-x[as.character(1:tree$Nnode+Ntip(tree))]
x<-x[tree$tip.label]
## fit no trend model
fit.bm<-anc.ML(tree,x,model="BM")
print(fit.bm)
## fit trend model
fit.trend<-anc.trend(tree,x)
print(fit.trend)
## compare trend vs. no-trend models & estimates
```

```
AIC(fit.bm,fit.trend)
layout(matrix(c(3,4,1,2,5,6),3,2,byrow=TRUE),
    heights=c(1.5,3,1.5),widths=c(3,3))
xlim<-ylim<-range(c(a,fit.bm$ace,
    fit.trend$ace))
plot(a,fit.bm$ace,pch=19,
    col=make.transparent("blue",0.5),
    xlab="true ancestral states",
    ylab="ML estimates",
    main=paste("Comparison of true and estimated",
    "\nstates under a no-trend model"),font.main=3,
    cex.main=1.2,bty="l",cex=1.5,
    xlim=xlim,ylim=ylim)
lines(xlim,ylim,lty="dotted")
plot(a,fit.trend$ace,pch=19,
    col=make.transparent("blue",0.5),
    xlab="true ancestral states",
    ylab="ML estimates",
    main=paste("Comparison of true and estimated",
    "\nstates under a trend model"),font.main=3,
    cex.main=1.2,bty="l",cex=1.5,
    xlim=xlim,ylim=ylim)
lines(xlim,ylim,lty="dotted")
par(mfrow=c(1,1))
```

`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=100000, sequence=NULL, method="mcmc",
   model=c("BM","OU","lambda"), 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"`.
`model`	model for the evolution of the liability. Options are `"BM"` (Brownian motion, the default), `"OU"` (Ornstein-Uhlenbeck), or `"lambda"` (the λ model).
`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; `propalpha` variance on the proposal distribution for `alpha` (for `model="OU"`); `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).

Details

`print` and `plot` S3 methods are now available for the object class `"ancThresh"`.

Value

This function returns an object of class `"ancThresh"` containing the posterior sample from our analysis, although with other components.

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.

Revell, L. J. (2014) Ancestral character estimation under the threshold model from quantitative genetics. *Evolution*, **68**, 743-759.

See Also

`anc.Bayes`, `threshBayes`

Examples

```
## Not run:
## load data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)
## extract character of interest
```

```
fmode<-setNames(sunfish.data$feeding.mode,
    rownames(sunfish.data))
## run MCMC
mcmc<-ancThresh(sunfish.tree,fmode,ngen=1000000)
## plot results
plot(mcmc,mar=c(0.1,0.1,4.1,0.1))
title(main="Posterior probabilities for node states",
    font.main=3)
## End(Not run)
```

`anoletree` *Phylogenetic datasets*

Description

`anoletree` is a phylogeny of Greater Antillean anole species with a mapped discrete character - *ecomorph class*. `anole.data` is a data frame of morphological characters. Data and tree are from Mahler et al. (2010).

`flatworm.tree` and `flatworm.data` are a phylogeny and dataset of habitat preferences for flatworms from Benitez-Alvarez et al. (2000). `flatworm.tree` has been made ultrametric using penalized likelihood.

`mammal.tree` and `mammal.data` are the phylogeny and dataset for mammal body size and home range size from Garland et al. (1992).

`salamanders` is a phylogeny of *Plethodon* salamanders from Highton and Larson (1979). According to Wikipedia, the genus *Plethodon* contains 55 species in total.

`sunfish.tree` and `sunfish.data` are the phylogeny and dataset for Centrarchidae and buccal morphology (respectively) from Revell and Collar (2009).

`vertebrate.tree` is a time-calibrated phylogeny of vertebrates and `vertebrate.data` is a dataset of phenotypic traits. The phylogeny is from `http://www.timetree.org/` (Hedges et al. 2006).

`wasp.trees` and `wasp.data` are the phylogeny and host-parasite associations from Lopez-Vaamonde et al. (2001).

Usage

```
data(anole.data)
data(anoletree)
data(flatworm.tree)
data(flatworm.data)
data(mammal.data)
data(mammal.tree)
data(salamanders)
data(sunfish.data)
data(sunfish.tree)
data(vertebrate.tree)
data(vertebrate.data)
data(wasp.data)
data(wasp.trees)
```

Format

`anoletree` is an object of class `"simmap"`. `anole.data` is a data frame.

`flatworm.tree` is an object of class `"phylo"`. `flatworm.data` is a data frame.

`mammal.tree` is an object of class `"phylo"`. `mammal.data` is a data frame.

`salamanders` is an object of class `"phylo"`.

`sunfish.tree` is an object of class `"simmap"`. `sunfish.data` is a data frame.

`vertebrate.tree` is an object of class `"phylo"`. `vertebrate.data` is a data frame.

`wasp.trees` is an object of class `"multiPhylo"`. `wasp.data` is a data frame.

Source

Benitez-Alvarez, L., A. Maria Leal-Zanchet, A. Oceguera-Figueroa, R. Lopes Ferreira, D. de Medeiros Bento, J. Braccini, R. Sluys, and M. Riutort. (2020) Phylogeny and biogeography of the Cavernicola (Platyhelminthes: Tricladida): Relicts of an epigean group sheltering in caves? *Molecular Phylogenetics and Evolution*, **145**, 106709.

Garland, T., Jr., P. H. Harvey, and A. R. Ives. (1992) Procedures for the analysis of comparative data using phylogenetically independent contrasts. *Systematic Biology*, **41**, 18-32.

Hedges, S. B., J. Dudley, and S. Kumar. (2006) TimeTree: A public knowledgebase of divergence times among organisms. *Bioinformatics*, **22**, 2971-2972.

Highton, R., and A. Larson. (1979) The genetic relationships of the salamanders of the genus *Plethodon*. *Systematic Zoology*, **28**, 579-599.

Lopez-Vaamonde, C., J. Y. Rasplus, G. D. Weiblen, and J. M. Cook. (2001) Molecular phylogenies of fig wasps: Partial cocladogenesis of pollinators and parasites. *Molecular Phylogenetics and Evolution*, **21**, 55-71.

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.

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

`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

An object of class `"phylo"` or `"simmap"`.

Author(s)

Liam 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.

`as.multiPhylo` *Conversion to object of class* `"multiPhylo"`

Description

This function converts between object classes.

Usage

```
as.multiPhylo(x, ...)
## S3 method for class 'multiSimmap'
as.multiPhylo(x, ...)
## S3 method for class 'phylo'
as.multiPhylo(x, ...)
```

Arguments

`x`	object to be converted to `"multiPhylo"`. Presently an object of class `"multiSimmap"`, or an object of class `"phylo"`. In the latter case an object of class `"multiPhylo"` with length 1 is generated.
`...`	optional arguments.

Value

An object of class `"multiPhylo"`.

Author(s)

Liam 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.

`as.Qmatrix` *Convert a fitted M*k *model to a Q-matrix*

Description

This function extracts a **Q**-matrix (in the form of an object of class `"Qmatrix"` from a fitted M*k* model.

Usage

```
as.Qmatrix(x, ...)
## S3 method for class 'fitMk'
as.Qmatrix(x, ...)
## S3 method for class 'Qmatrix'
plot(x, ...)
## S3 method for class 'Qmatrix'
print(x, ...)
```

Arguments

`x`	fitted M*k* model. (For instance, an object of class `"fitMk"`.) In the case of `print.Qmatrix`, an object of class `"Qmatrix"`.
`...`	optional arguments.

Value

An object of class `"Qmatrix"`.

`plot.Qmatrix` invisibly returns the coordinates of vertices of the plotted **Q**-matrix.

Author(s)

Liam Revell <`liam.revell@umb.edu`>, Joan Maspons

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`fitMk`

`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`	σ_1^2.
`sig2`	σ_2^2.
`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>`

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`evol.rate.mcmc`, `minSplit`, `posterior.evolrate`

`averageTree` *Compute an average tree from a set of trees and related operations*

Description

These functions compute average trees or consensus trees by various criteria.

Usage

```
averageTree(trees, start=NULL, method="quadratic.path.difference",
   tol=1e-12, quiet=FALSE, ...)
ls.consensus(trees, start=NULL, tol=1e-12, quiet=FALSE, ...)
minTreeDist(tree, trees, method="quadratic.path.difference", ...)
```

Arguments

`trees`	object of class `"multiPhylo"`.
`tree`	object of class `"phylo"`. For `minTreeDist` the tree on which to find the edge lengths that minimize the distance to the phylogenies in `trees`.
`start`	starting tree for optimization.
`method`	distance criterion for minimization. Options are `"symmetric.difference"`, `"branch.score.difference"`, `"path.difference"`, and `"quadratic.path.difference"`.
`tol`	tolerance value for optimization.
`quiet`	logical value indicating whether to run "quietly" or not.
`...`	other arguments to be passed internally.

Value

An object of class `"phylo"` with edge lengths.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

`bd` *Convert object of class "birthdeath" to raw birth & death rates*

Description

This function converts an object of class `"birthdeath"` to a vector with the ML birth & death rates.

Note that this is somewhat unnecessary as *phytools* now contains functions to fit birth-death and pure-birth diversification models from trees (`fit.bd` and `fit.yule`) that also take into account incomplete sampling fraction.

Usage

```
bd(x)
```

Arguments

`x`	object of class `"birthdeath"`.

Value

A vector.

Author(s)

Liam 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.

See Also

`birthdeath`, `fit.bd`, `fit.yule`

`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,
   interactive=FALSE, ...)
```

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 *rootward* of 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 *rootward* of the node to add the new tip.
`interactive`	logical value indicating whether or not the species should be added interactively. (Defaults to `FALSE`.)
`...`	arguments to be passed to `plotTree` (for `interactive=TRUE`.)

Details

Wrapper function for 'ape' `bind.tree`. Note that `interactive=TRUE` works only for right-facing phylograms.

Value

An object of class `"phylo"`.

Author(s)

Liam 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.

Examples

```
set.seed(123)
## generate tree
tree<-pbtree(b=0.1, n=10)

## plot original tree
plot(tree)
axisPhylo()

## add an extant tip ("t_extant") sister to taxon 't5'
## with divergence time of 4.5 Ma
node <- which(tree$tip.label=="t5")
tree <- bind.tip(tree, tip.label="t_extant",
    where=node, position=4.5)
# plot to see the result
plot(tree)
axisPhylo()

## add an extinct tip ("t_extinct") sister to 't2' with
## divergence time of 7.8 Ma and duration (edge length) of
## 3.3 Ma
node <- which(tree$tip.label=="t2")
tree <- bind.tip(tree, tip.label="t_extinct", where=node,
    position=7.8, edge.length=3.3)

## plot to see the result
plot(tree)
axisPhylo()
```

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

Description

This function grafts tree `y` onto tree `x` at node `where`.

Usage

```
bind.tree.simmap(x, y, where="root")
```

Arguments

`x`	an object of class `"simmap"`. (The receptor tree.)
`y`	an object of class `"simmap"`. (The tree being grafted.)
`where`	node number to attach new tip, or the root node if `where="root"`.

Details

This function wraps around `bind.tree` for objects of class `"simmap"`; however it presently only allows `y` to be grafted at a node of `x` and it does not allow `y` to possess a root edge.

Value

An object of class `"simmap"`.

Author(s)

Liam 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.

`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` to be equal to the total height of the tree in generations (and thus avoid rescaling).

For `type="threshold"` the visualization is of the threshold model (Felsenstein 2012), in which the evolving character is liability and the segments 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 `colors`. 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 `colors` is provided, the function will recycle a set of four colors up to the number of times required by `thresholds`. Finally, the optional argument `return.tree=TRUE` will tell the function to return a list with the tip and note states and an object of class `"phylo"` with (for `type="threshold"`), the state for the threshold model through time mapped on the branches of the tree in discrete time.

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), σ^2.
`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.

It also returns, by default (although this can be turned off) a tree with the branch lengths in discrete time and with a mapped discrete character (for `type="threshold"`).

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J. (2014) Ancestral character estimation under the threshold model from quantitative genetics. *Evolution*, bold68, 743-759.

See Also

`fastBM`, `pbtree`, `phenogram`, `threshBayes`

Examples

```
set.seed(999)
## plot BM simulation on 12 taxon tree
tree<-pbtree(n=12)
par(mfrow=c(1,2),mar=c(5.1,4.1,4.1,0.1))
x<-bmPlot(tree,bty="l")
plotTree(tree,direction="upwards",
    mar=c(5.1,0.1,4.1,1.1),ftype="off")
## reset par to default values
par(mfrow=c(1,1),mar=c(5.1,4.1,4.1,2.1))
## plot simulation of a threshold character
par(mfrow=c(1,2),mar=c(5.1,4.1,4.1,0.1))
tt<-bmPlot(tree,type="threshold",thresholds=c(0,1,2),
    bty="l")
plot(tt$tree,direction="upwards",
    mar=c(5.1,0.1,4.1,1.1),ftype="off",
    colors=setNames(c("black","red","blue"),
    letters[1:3]),lwd=3)
## reset par to default values
par(mfrow=c(1,1),mar=c(5.1,4.1,4.1,2.1))
```

`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, σ^2.
`b`	birthrate for branching process.
`time.stop`	number of generations to run.
`ylim`	*y* limits (for plotting).
`smooth`	no longer used.
`pause`	pause (in s) between generations.
`record`	filename for video file output (no video if `NULL`).
`path`	full path to file for video rendering. (By default `branching.diffusion` will look for the executable `ffmpeg.exe` in the directory `C:/Program Files/ffmpeg/bin`, even though this will not make sense on non-Windows machines.)
`...`	optional arguments.

Value

An animated plot and (optionally) a recorded video file. For animation to be recorded to file, the function requires the package *animation* as well as a video renderer.

Author(s)

Liam 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.

See Also

`bmPlot`, `fastBM`

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

Description

This function takes an object of class `"phylo"` or class `"simmap"` 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 ("noncensored") model of O'Meara et al. (2006; *Evolution*). This is also the basic model implemented in Brian O'Meara's *Brownie* software.

Usage

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

Arguments

`tree`	a phylogenetic tree either as an object of class `"phylo"` or `"simmap"`. (See `read.simmap`, `make.simmap`, or `paintSubTree` for more details about the latter object class.)
`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`.
`...`	optional arguments.

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

An object of class `"brownie.lite"` containing the following components:

`sig2.single`	is the rate, σ^2, 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 (σ_1^2, σ_2^2, and so on) 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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`brownieREML`, `evol.vcv`, `ratebytree`

Examples

```
## load data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)
## extract character of interest
buccal.length<-setNames(sunfish.data$buccal.length,
    rownames(sunfish.data))
## fit model
multiBM.fit<-brownie.lite(sunfish.tree,
    buccal.length)
print(multiBM.fit)
```

`brownieREML` *REML version of brownie.lite*

Description

This function takes an object of class `"phylo"` or an object of class `"simmap"` 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`	an object of class `"phylo"` or `"simmap"`. (See `read.simmap` and `make.simmap` for more information about the latter object class.)
`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.
`...`	optional arguments.

Value

An object of class `"brownieREML"` containing the following components:

`sig2.single`	is the rate, σ^2, 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 (σ_1^2, σ_2^2, and so on) 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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

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

`cladelabels` *Add labels to subtrees of a plotted phylogeny*

Description

This function adds clade labels to a plotted tree.

Usage

```
cladelabels(tree=NULL, text, node, offset=NULL, wing.length=NULL, cex=1,
   orientation="vertical")
arc.cladelabels(tree=NULL, text, node=NULL, ln.offset=1.02,
   lab.offset=1.06, cex=1, orientation="curved", stretch=1, ...)
```

Arguments

`tree`	an object of class `"phylo"`. If not supplied, the function will obtain the last plotted phylogeny from the environmental variable `last_plot.phylo`.
`text`	desired clade label text.
`node`	node number for the most recent common ancestor of members of the clade. For `arc.cladelabels` this defaults to `NULL` which means that the node of the clade to be labeled should be specified interactively (that is, by clicking on the graphical device).
`offset`	offset (as a multiplier of character width) for the label. Defaults to `offset=1` if `tree` is supplied or `offset=8` otherwise.
`wing.length`	length of the wings to add to the top & bottom of the label bar (in character widths).
`cex`	character expansion factor.
`orientation`	orientation of the text. Can be `orientation = "vertical"` (the default) or `"horizontal"`.
`ln.offset`	line offset (as a function of total tree height) for `arc.cladelabels`.
`lab.offset`	label offset for `arc.cladelabels`.
`stretch`	argument for `arc.cladelabels` to be passed to `arctext`.
`...`	optional arguments for `arc.cladelabels`.

Details

`cladelabels` presently works only for rightward facing plotted phylogenies - but no warning will be returned if your tree does not conform to this requirement!

`arc.cladelabels` is designed to do a similar thing to `cladelabels`, but for plotted fan trees. This function checks to ensure that the most recently plotted tree was plotted with `type="fan"` style.

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`nodelabels`

`collapse.to.star` *Collapse a subtree to a star phylogeny*

Description

This function collapses a subtree to a star. If the tree has edge lengths, the function will keep the tips at the same height above the root as in the original tree.

If `node` is the global root of the tree a star phylogeny will be created.

Usage

```
collapse.to.star(tree, node)
```

Arguments

`tree`	an object of class `"phylo"`.
`node`	node for the clade to be collapsed.

Value

An object of class `"phylo"`.

Author(s)

Liam 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.

See Also

`splitTree`, `starTree`

`collapseTree` *Interactive tree visualizer*

Description

Function creates an interactive visualization of collapsing & expanding clades on the tree.

Usage

```
collapseTree(tree, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`...`	optional arguments. These *mostly* match the arguments of `plotSimmap`, but also include the argument `drop.extinct=TRUE` which will (if the input tree is ultrametric) drop any 'extinct' lineages from the tree that is returned by the function.

Details

Function first plots a fan style tree, and then the user collapses node on the tree by clicking on them. Collapsed nodes are collapsed to the common ancestor of the clade. Nodes that have been collapsed can also be expanded by clicking. Right-click to end.

Value

Returns the final plotted tree.

Author(s)

Liam 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.

See Also

`plotTree`, `plotSimmap`

Examples

```
## Not run:
data(anoletree)
pruned<-collapseTree(anoletree)
## End(Not run)
```

`compare.chronograms` *Compares two chronograms with precisely matching nodes in a visual manner*

Description

This function plots two trees, with semi-transparent colors by default, & uses arrows to highlight differences in depth of corresponding nodes between the trees.

Usage

```
compare.chronograms(t1, t2, ...)
```

Arguments

`t1`	object of class `"phylo"`.
`t2`	object of class `"phylo"` that matches `t1` precisely in topology & node rotations, but differs in edge lengths.
`...`	optional arguments.

Value

Function creates a plot.

Author(s)

Liam 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.

`consensus.edges` *Compute consensus edges for a tree under some criterion*

Description

This function computes consensus edge lengths by different methods.

Usage

```
consensus.edges(trees, method=c("mean.edge","least.squares"), ...)
```

Arguments

`trees`	object of class `"multiPhylo"` in which the trees must have edge lengths. This could be, for instance, a sample from the posterior distribution of trees in a Bayesian analysis.
`method`	method for computing the edge lengths. Could be the mean of all trees in which the edge is present, or it could be the least-squares edge lengths computed on the mean patristic distance matrices from the input phylogenies in `trees`. Note that in the latter case the phangorn function `nnls.tree` is used and the option `rooted` will be set to `is.rooted(tree)` for the consensus tree.

`...`	optional arguments, the most popular of which is `consensus.tree` - a user supplied consensus tree. Another optional argument for `method="mean.edge"` is `if.absent` which tells the function how to include absent edges in the computation of average edge lengths. Possible values are `"zero"` (the default) or `"ignore"`.

Value

An object of class `"phylo"` with edge lengths.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

`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 then interpolating the states along each edge using equation [2] of Felsenstein (1985).

`errorbar.contMap` adds error bars to an existing plot.

Usage

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

Arguments

`tree`	object of class `"phylo"`.
`x`	a numerical 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 (to a point) indicate a finer (smoother) gradient.
`fsize`	relative font size - can be a vector of length 2 in which the first element gives the font size for the tip labels & 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. Can be a single integer number or a vector. In the latter case, the second number will be taken to be the desired legend width.
`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.
`direction`	plotting direction for `type="phylogram"`.
`plot`	logical value indicating whether or not to plot the tree. If `plot=FALSE` then an object of class `"contMap"` will be returned without plotting.
`obj`	object of class `"contMap"`.
`...`	optional arguments for `plot.contMap` which include all the arguments of `contMap` except for `tree`, `x`, `res`, and `lims`. Also `method`, `"fastAnc"`, `"anc.ML"`, or `"user"` (for user-supplied states) specifying which function to use for ancestral state estimation; `hold` specifies whether or not to hold output to graphical device before plotting (defaults to `hold=TRUE`); and `anc.states` a vector containing some or multiple ancestral user-supplied ancestral states at nodes. Some other plotting arguments, such as `xlim` and `ylim`, may also work. Optional arguments for `errorbar.contMap` include `x`, a vector containing the original trait values mapped onto the tree (otherwise these will be obtained from `obj`), `scale.by.ci`, a logical argument (defaulting to `TRUE`) that determines whether or not the length of the error bars will be scaled by the CI width, and `lwd`, which detemines the line width of the plotted error bars.

Value

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

`errorbar.contMap` adds colorful error bars to a plotted tree.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J. 2013. Two new graphical methods for mapping trait evolution on phylogenies. *Methods in Ecology and Evolution*, **4**, 754-759.

See Also

`anc.ML`, `densityMap`, `fastAnc`, `plotSimmap`

Examples

```
## load data from Garland et al. (1992)
data(mammal.tree)
data(mammal.data)
## extract character of interest
ln.bodyMass<-log(setNames(mammal.data$bodyMass,
    rownames(mammal.data)))
## create "contMap" object
mammal.contMap<-contMap(mammal.tree,
    ln.bodyMass,plot=FALSE,res=200)
## change color scheme
mammal.contMap<-setMap(mammal.contMap,
    c("white","#FFFFB2","#FECC5C","#FD8D3C",
    "#E31A1C"))
plot(mammal.contMap,fsize=c(0.7,0.8),
    leg.txt="log(body mass)")
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margins to default
```

`cophylo` *Creates a co-phylogenetic plot*

Description

This function creates an object of class `"cophylo"` or, in the case of `plot.cophylo`, plots that object. The function can (optionally) first attempt to rotate the nodes of both trees to optimize vertical matching of tips.

`cotangleplot` creates a co-phylogenetic plot in which the edges of the matched trees are crossing.

Usage

```
cophylo(tr1, tr2, assoc=NULL, rotate=TRUE, ...)
## S3 method for class 'cophylo'
plot(x, ...)
cotangleplot(tr1, tr2, type=c("cladogram","phylogram"),
use.edge.length=TRUE, tangle=c("both","tree1","tree2"), ...)
```

Arguments

`tr1`	object of class `"phylo"`.
`tr2`	object of class `"phylo"`.
`assoc`	matrix containing the tip labels in `tr1` to match to the tip labels in `tr2`. Note that not all labels in either tree need to be included; and, furthermore, one label in `tr1` can be matched with more than one label in `tr2`, or vice versa.

`rotate`	logical argument indicating whether nodes on both trees should be rotated to attempt to match in vertical position.
`x`	in the case of `plot.cophylo`, an object of class `"cophylo"` to be plotted.
`type`	for `cotangleplot`, the tree plotting style.
`use.edge.length`	for `cotangleplot`, a logical value indicating whether or not to plot trees with edge lengths.
`tangle`	for `cotangleplot`, whether to tangle the left tree, the right tree, or both.
`...`	optional arguments to be passed to `tipRotate`, or, in the case of `plot.cophylo`, to the internally used tree plotting function, `phylogram`. `phylogram` takes similar arguments to `plotSimmap`, such as `fsize`, `ftype`, `lwd`, and `pts`, though not all options from `plotSimmap` and `plotTree` are available. If `fsize` is supplied as a vector, different size fonts for the left & right facing trees may be used. In addition, the optional argument `scale.bar`, which should be a vector containing the lengths of the scale bars desired for the right & left trees, will add scale bars to the plot when supplied to `plot.cophylo`. If either tree contains polytomies, the `cophylo` argument `rotate.multi` should be set to `TRUE`. If curved linking lines are desired, the `plot.cophylo` argument `link.type` should be set to `"curved"`. Other arguments for the `plot` method include `link.col`, `link.lty`, and `link.lwd`, which can be supplied as a scalar or a vector in which the order of the elements corresponds to the order of the associations in `assoc`. Finally, `edge.col`, a list consisting of two vectors (`left` and `right`) can be used to specify the edge colors of the two left & right plotted trees. Note that the edge order is the same as in the *rotated* trees, assuming that a rotation has been performed on `x`.

Details

If no matrix of associations, `assoc`, is provided, then `cophylo` will look for exact matches of tip labels between trees.

Value

An object of class `"cophylo"` which includes the following components or a pair of plotted facing phylogenies with links between tips as specified in `assoc`.

Author(s)

Liam 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.

See Also

`cophyloplot`, `plotSimmap`

Examples

```
## load data from Lopez-Vaamonde et al. (2001)
data(wasp.trees)
data(wasp.data)
## create co-phylogenetic object
wasp.cophylo<-cophylo(wasp.trees[[1]],wasp.trees[[2]],
    assoc=wasp.data)
## plot co-phylogenies
plot(wasp.cophylo,link.type="curved",link.lwd=4,
     link.lty="solid",link.col=make.transparent("red",
     0.25))
par(mar=c(5.1,4.1,4.1,2.1))
```

`cospeciation` *Conducts a statistical test of cospeciation between two trees*

Description

This function conducts a test for cospeciation based on tree distance, applying a distance metric selected by the user.

Note that this method should be prone to be quite liberal as the null hypothesis is no similarity between trees!

Usage

```
cospeciation(t1, t2, distance=c("RF","SPR"),
   method=c("simulation","permutation"), assoc=NULL,
   nsim=100, ...)
## S3 method for class 'cospeciation'
plot(x, ...)
## S3 method for class 'cospeciation'
print(x, ...)
```

Arguments

`t1`	object of class `"phylo"`.
`t2`	object of class `"phylo"`.
`distance`	distance method to compare trees.
`method`	method to use (simulation of pure-birth trees, or permutation of tip labels on a fixed tree) to obtain a null distribution of tree distances via `distance`.
`assoc`	matrix containing the tip labels in `t1` to match to the tip labels in `t2`. Note that not all labels in either tree need to be included; however, unlike `cophylo`, one label in `t1` cannot be matched with more than one label in `t2`, nor vice versa. If `NULL` then an exact match of tip labels will be sought.
`nsim`	number of simulations or permutations.
`x`	for `plot` and `print` methods, an object of class `"cospeciation"`.
`...`	optional arguments.

Value

An object of class `"cospeciation"`, which includes the test-statistic, the null distribution, and a p-value for the test of the null hypothesis of no topological similarity between the two trees.

Author(s)

Liam 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.

See Also

`cophylo`

Examples

```
## load data from Lopez-Vaamonde et al. (2001)
data(wasp.trees)
data(wasp.data)
## test for cospeciation
wasp.cosp<-cospeciation(wasp.trees[[1]],wasp.trees[[2]],
    assoc=wasp.data)
print(wasp.cosp)
plot(wasp.cosp)
title(main=paste("Simulated distribution of RF distances\n",
    "between unassociated trees"),font.main=3)
```

`countSimmap` *Counts the number of character changes on a object of class "simmap" or "multiSimmap"*

Description

This function takes a tree or a set of trees with a mapped discrete character (that is, an object of class `"simmap"` or `"multiSimmap"`), 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`	an object of class `"simmap"` or `"multiSimmap"`.
`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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Examples

```
## load data from Revell & Collar (2009)
data(anoletree)
anoletree
countSimmap(anoletree)
```

`ctt` *Generates (or simulates) a 'changes through time' plot from a set of stochastic map character histories*

Description

This function generates a 'changes through time' plot in the style of a lineage-through-time (LTT) plot. It shows the mean rate or the mean number of changes per unit time from a set of stochastic character map trees.

Usage

```
ctt(trees, segments=20, ...)
sim.ctt(tree, Q, anc=NULL, nmaps=100, ...)
sim.multiCtt(tree, Q, anc=NULL, nmaps=100, nsim=100, ...)
```

Arguments

`trees`	an object of class `"multiSimmap"`.
`segments`	number of segments to break up the history of the tree.
`tree`	for `sim.ctt`, an object of class `"phylo"`.
`Q`	for `sim.ctt`, a transition matrix to use for simulation.
`anc`	ancestral state at the root node for simulation.
`nmaps`	number of stochastic maps per simulation.
`nsim`	for `sim.multiCtt` only, the number of simulations to run.
`...`	optional arguments.

Value

An object of class `"ctt"` or `"multiCtt"`.

Author(s)

Liam 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.

See Also

`ltt`

`density.multiSimmap` *Computes a posterior distribution for the number and types of changes on the tree*

Description

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

Usage

```
## S3 method for class 'multiSimmap'
density(x, ...)
## S3 method for class 'changesMap'
plot(x, ...)
```

Arguments

`x`	object of class `"multiSimmap"` (see `make.simmap`), or, in the case `plot.changesMap`, an object of class `"changesMap"` produced via a call to `density.multiSimmap`.
`...`	optional arguments. For `density.multiSimmap` these consist of `bw` (bandwidth) & `method` (`"changes"`, `"densityMap"`, or `"timings"`).

Details

In `density.multiSimmap` `method="changes"`, the default, results in a posterior distribution of the number & types of changes on the tree. If the package *coda* has been installed, then the function `HPD.interval` is used to compute a 95-percent high probability density interval for the number of changes of each type on the tree. Otherwise, the central 95-percent of the posterior sample is returned as an estimate of the 95-percent HPD interval for each change type. The method also computes the full posterior density for each change type using a bandwidth specified by the user. `method="densityMap"` computes a standard `"densityMap"` object, and thus only permits binary characters. Finally `method="changes"` has not yet been implemented.

`plot.changesMap` plots the posterior density returned by `density.multiSimmap` for `method= "changes"`.

Value

For `method="changes"` `density.multiSimmap` returns an object of class `"changesMap"`.

For `method="densityMap"` `density.multiSimmap` returns an object of class `"densityMap"`.

`plot.changesMap` generates a plot.

Author(s)

Liam 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.

`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", direction="rightwards",
   plot=TRUE, ...)
## S3 method for class 'densityMap'
plot(x, ...)
```

Arguments

`trees`	set of phylogenetic trees in a modified `"multiPhylo"` object. Values for a two-state 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. If a vector of two elements is supplied, the second element will be taken to be the desired width of the legend bar.
`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.
`plot`	logical value indicating whether or not to plot the tree. If `plot=FALSE` then an object of class `"densityMap"` will be returned without plotting.
`direction`	plotting direction for `type="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`. Additional optional arguments include `mar` (margins), `offset` (tip label offset: in units of the edge length or character widths, as in `plotSimmap`), and `hold` (whether or not to use `dev.hold` to hold output to graphical device before plotting; defaults to `hold=TRUE`). Also, the argument `states` can be used to 'order' the states on the probability axis (that is, which state should correspond to a posterior probability of 0 or 1). Some other plotting arguments, such as `xlim` and `ylim`, may also work.

Value

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

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.

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J. 2013. Two new graphical methods for mapping trait evolution on phylogenies. *Methods in Ecology and Evolution*, **4**, 754-759.

See Also

`make.simmap`, `plotSimmap`, `read.simmap`

Examples

```
## Not run:
## load tree and data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)
## extract discrete character (feeding mode)
fmode<-setNames(sunfish.data$feeding.mode,
     rownames(sunfish.data))
## do stochastic mapping
smap.trees<-make.simmap(sunfish.tree,fmode,model="ER",
     nsim=100)
## compute "densityMap" object
sunfish.dmap<-densityMap(smap.trees,plot=FALSE,
res=50) ## res should be higher
## plot density map
plot(sunfish.dmap,lwd=5,outline=TRUE)
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margins to default

## End(Not run)
```

`densityTree` *Plots a posterior sample of trees*

Description

Functions plots a posterior sample of trees, including with mapped discrete characters.

`make.transparent` is used internally and converts a color to transparent with a certain user-specified `alpha` level.

Usage

```
densityTree(trees, colors="blue", alpha=NULL, method="plotTree",
   fix.depth=FALSE, use.edge.length=TRUE, compute.consensus=TRUE,
   use.gradient=FALSE, show.axis=TRUE, ...)
make.transparent(color, alpha)
```

Arguments

`trees`	an object of class `"multiPhylo"` or `"multiSimmap"`.
`colors`	a color or a named vector of colors in which names correspond to mapped states in an object of class `"multiSimmap"`.
`alpha`	transparency level for plotted trees which is passed to internally used function, `make.transparent`. (`0` is fully transparent, which `1` is fully opaque.) By default will be one divided by the number of trees.
`method`	plotting method to be used internally. Can be `"plotTree"` or `"plotSimmap"`.
`fix.depth`	logical value indicating whether or not to plot trees with a fixed depth or to permit plotted trees to have different depths.
`use.edge.length`	logical value indicating whether to use the edge lengths of the input tree. Defaults to `use.edge.length=TRUE` unless any input tree edge lengths are `NULL`.
`compute.consensus`	logical value indicating whether or not to use the tip order from a consensus tree. (Defaults to `compute.consensus=TRUE` Defaulted to `FALSE` in earlier version of this function.)
`use.gradient`	logical value indicating whether to plot all trees slightly offset using a rainbow color gradient. (Defaults to `use.gradient=FALSE`.)
`show.axis`	logical value indicating whether or not to include a horizontal axis in the plot.
`...`	arguments to be passed to `plotTree` or `plotSimmap`. Some may be ignored if they are incompatible with the method.
`color`	in `make.transparent`, the color (or colors in a vector) to render transparent.

Value

Function creates a plot.

Author(s)

Liam 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.

---

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, ...)
## S3 method for class 'simmap'
summary(object, ...)
## S3 method for class 'multiSimmap'
summary(object, ...)
## S3 method for class 'describe.simmap'
plot(x, ...)
```

Arguments

`tree`	a single tree or a set of trees as an object of class `"simmap"` or `"multiSimmap"`, respectively.
`object`	object of class `"simmap"` or `"multiSimmap"`.
`x`	for S3 `plot` method, an object of class `"describe.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

An object of class `"describe.simmap"` with the following elements:

`count`	a matrix containing the number and types of transitions for each tree, if `tree` is an object of class `"multiSimmap"`.
`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 `tree` is an object of class `"multiSimmap"`.
`legend`	a vector containing the plot legend, if `plot=TRUE`.

if `class(tree)="simmap"` 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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`di2multi.simmap` *Collapse or resolve polytomies in a tree with a character painted on the edges*

Description

The method `di2multi` collapses branches of zero length (or, more specifically, branches with length shorter than `tol`) to create a polytomy in a tree or set of trees. The method `multi2di` resolves polytomies by adding branches of zero length (while preserving the mappings) in a tree or set of trees.

Usage

```
## S3 method for class 'simmap'
di2multi(phy, ...)
## S3 method for class 'simmap'
multi2di(phy, ...)
## S3 method for class 'multiSimmap'
di2multi(phy, ...)
## S3 method for class 'multiSimmap'
multi2di(phy, ...)
## S3 method for class 'contMap'
di2multi(phy, ...)
## S3 method for class 'contMap'
multi2di(phy, ...)
## S3 method for class 'densityMap'
di2multi(phy, ...)
## S3 method for class 'densityMap'
multi2di(phy, ...)
```

Arguments

`phy`	object of class `"simmap"`, `"multiSimmap"`, `"contMap"`, or `"densityMap"` containing a character mapped onto the edges of a tree or set of trees.
`...`	optional arguments: `tol`, length below which edges should be treated as having zero length; and `random`, specifying whether to resolve polytomies randomly (if `TRUE`) or in the order in which they are encountered.

Details

This methods should theoretically behave similarly to `di2multi` and `multi2di` from the *ape* package.

Value

An object of class `"simmap"`, `"multiSimmap"`, `"contMap"`, or `"densityMap"`, depending on the class of `phy`.

Author(s)

Liam 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.

See Also

`contMap`, `densityMap`, `di2multi`, `make.simmap`, `multi2di`, `read.simmap`

`dotTree` *Creates a phylogenetic dot plot*

Description

Creates a plot in which different sized dots/circles represent different tip values for a quantitative trait.

Usage

```
dotTree(tree, x, legend=TRUE, method="plotTree", standardize=FALSE, ...)
dot.legend(x, y, min, max, Ntip, length=5, prompt=FALSE, method="plotTree",
   ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`x`	vector of trait values; or a matrix. If `x` is a vector it must have names that correspond to the tip labels of `tree`. If `x` is a matrix (and it probably should be a *matrix*, not a data frame) then the row names of the matrix should correspond to the tip labels of the phylogeny. In the case of `dot.legend`, the *x* coordinate of the legend.
`legend`	logical value indicating whether or not a legend should be plotted.

`method`	tree plotting method to be used internally. Will switch to `method="phylogram"` if the number of traits is greater than one. For `dot.legend`, it should be the method that was used for the plot.
`standardize`	a logical value indicating whether or not to standardize `x`, or each column of `x`, to have a mean of zero & variance of one prior to analysis.
`y`	*y* coordinate of the legend.
`min`	minimum value for `dot.legend`.
`max`	maximum value for `dot.legend`.
`Ntip`	number of tips in the plotted tree for `dot.legend`.
`length`	length of legend.
`prompt`	logical value indicating whether or not to prompt for legend position.
`...`	optional arguments. In the case of `dotTree`, these will be passed to `plotTree` or a different internally used plotting function for `method="phylogram"`. See `phylo.heatmap` for more detail on these arguments. Other option for `dotTree` also include `data.type` (`"continuous"` or `"discrete"`), `colors`, `length`, for data type `"continuous"` the length of the legend in terms of plotted circles, `x.space`, the spacing of the columns in the plotted data matrix, and `leg.space`, the spacing of the legend dots (again, for `data.type="continuous"` only).

Value

Function creates a plot.

Author(s)

Liam 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.

Examples

```
## load data from Garland et al. (1992)
data(mammal.tree)
data(mammal.data)
## log-transform trait data
log.mammal<-log(mammal.data)
## plot dotTree
dotTree(mammal.tree,log.mammal,fsize=0.7,
    standardize=TRUE,length=10)
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margins to default
```

`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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`drop.leaves` *Drop all the leaves (tips) from a tree*

Description

Drops all the leaves from a tree, leaving behind only the structure leading to internal nodes.

Usage

```
drop.leaves(tree, ...)
```

Arguments

`tree`	object of class `"phylo"`.
`...`	optional arguments. Presently includes only the logical value `keep.tip.labels` which tells the function how to labels the tips on the reduced tree.

Value

An object of class `"phylo"`.

Author(s)

Liam 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.

`drop.tip.contMap` *Drop tip or tips from an object of class "contMap" or "densityMap"*

Description

This function drops one or multiple tips from an object of class `"contMap"` or `"densityMap"`. This function is equivalent to `drop.tip` but for an object of this class.

Usage

```
drop.tip.contMap(x, tip, ...)
drop.tip.densityMap(x, tip, ...)
keep.tip.contMap(x, tip, ...)
```

Arguments

`x`	an object of class `"contMap"` or `"densityMap"`.
`tip`	name or names of species to be dropped or kept.
`...`	optional arguments to be passed to `drop.tip.simmap`.

Details

For more information about objects of class `"contMap"` or `"densityMap"`, please refer to the documentation pages for `contMap` or `densityMap`, respectively.

Value

An object of class `"contMap"` or `"densityMap"`.

Author(s)

Liam 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.

See Also

`contMap`, `densityMap`, `drop.tip`, `drop.tip.simmap`, `keep.tip`

`drop.tip.multiPhylo` *Drop tip or tips from an object of class "multiPhylo" or "multiSimmap"*

Description

This function drops one or multiple tips from all the trees of an object of class `"multiPhylo"` or `"multiSimmap"`.

Usage

```
drop.tip.multiPhylo(phy, tip, ...)
drop.tip.multiSimmap(phy, tip, ...)
```

Arguments

`phy`	an object of class `"multiPhylo"` or `"multiSimmap"`.
`tip`	name or names of species to be dropped.
`...`	optional arguments to be passed to `drop.tip` or `drop.tip.simmap`. Most optional arguments work, with the exception of `interactive=TRUE` which will return an error.

Details

This function merely wraps `drop.tip` and `drop.tip.simmap`. Note that `drop.tip.multiSimmap` is merely just an alias of `drop.tip.multiPhylo`.

Value

An object of class `"multiPhylo"` or `"multiSimmap"`, depending on the input object class.

Author(s)

Liam 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.

See Also

`drop.tip`, `drop.tip.simmap`

`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.
`...`	optional arguments. Currently the logical argument `untangle` which if set to `TRUE` will call `untangle` before returning the `"simmap"` object to the user.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

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

Dtest *Conducts correlational D-test from stochastic mapping*

Description

This function conducts the 'D-test' of Huelsenbeck et al. (2003).

Usage

```
Dtest(t1, t2, nsim=100, ...)
```

Arguments

`t1`	set of stochastic map trees (i.e., object of class `"multiSimmap"` for character 1. Note that `t1` and `t2` should be of the same length.
`t2`	set of stochastic map trees (i.e., object of class `"multiSimmap"` for character 2. Note that `t1` and `t2` should be of the same length.
`nsim`	number of simulations to use in the test.
`...`	arguments to be passed internally to `make.simmap`. Note that (for now) these must be the same for both `t1` and `t2` (that is to say, we are not able to assume different trait evolution models for each tree).

Details

Note that this function has been included without much testing, and so the user should be wary.

Value

An object of class `"Dtest"`.

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`make.simmap`, `map.overlap`

`edge.widthMap` *Map continuous trait evolution on the tree*

Description

Function maps a discrete character onto the edges of the tree using variable edge widths.

Usage

```
edge.widthMap(tree, x, ...)
## S3 method for class 'edge.widthMap'
plot(x, max.width=0.9, legend="trait value", ...)
```

Arguments

`tree`	object of class `"phylo"`.
`x`	a numerical vector of phenotypic trait values for species. `names(x)` should contain the species names and match `tree$tip.label`. Or, for `plot.edge.widthMap`, an object of class `"edge.widthMap"`.
`max.width`	maximum edge width in plot units.
`legend`	label for the plot legend.
`...`	optional arguments - especially for the `plot` method. Perhaps the most important of these is `min.width`, which defaults to `0` but could probably be increased for many datasets and graphical devices. Other arguments are passed internally to `plotTree`.

Value

`edge.widthMap` returns an object of class `"edge.widthMap"`.

`plot.edge.widthMap` can be used to plot this object.

Author(s)

Liam 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.

See Also

`contMap`, `fastAnc`

Examples

```
## load data from Garland et al. (1992)
data(mammal.tree)
data(mammal.data)
## extract character of interest
ln.bodyMass<-log(setNames(mammal.data$bodyMass,
    rownames(mammal.data)))
## create "edge.widthMap" object
mammal.ewMap<-edge.widthMap(mammal.tree,ln.bodyMass,
min.width=0.05)
## plot it
plot(mammal.ewMap,legend="log(body mass)")
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margins to default
```

`edgeProbs` *Compute the relative frequencies of state changes along edges*

Description

This function computes the relative frequencies of character state changes along edges from a sample of stochastically mapped character histories. This function assumes that all trees in the sample differ only in their mapped histories & not at all in topology or branch lengths. Note that it only asks whether the starting and ending states of the edge differ in a particular way, and thus ignores multiple-hits along a single edge.

Usage

```
edgeProbs(trees)
```

Arguments

`trees`	an object of class `"multiSimmap"` containing a sample of trees that are identical in topology & branch lengths with different stochastically mapped character histories.

Value

The object that is returned is a matrix with the state changes & the relative frequency of each state change. Rows are in the order of the matrix edge for any of the mapped trees.

Author(s)

Liam 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.

See Also

`plotSimmap`

`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`; 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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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(), ...)
## S3 method for class 'evol.rate.mcmc'
print(x, ...)
## S3 method for class 'evol.rate.mcmc'
summary(object, ...)
## S3 method for class 'summary.evol.rate.mcmc'
print(x, ...)
## S3 method for class 'summary.evol.rate.mcmc'
plot(x, ...)
```

Arguments

`tree`	an object of class `"phylo"` (a phylogenetic tree).
`x`	a vector of tip values for species in which `names(x)` contains the species names of `tree`, an object of class `"evol.rate.mcmc"`, or (in the case of the S3 `summary` method) an object of class `"summary.evol.rate.mcmc"`.
`ngen`	an integer value indicating the number of generations for the MCMC.
`control`	a list of control parameters containing the following elements: `sig1`: starting value for σ_1^2; `sig2`: starting value for σ_2^2; `a`: starting value for a; `sd1`: standard deviation for the normal proposal distribution for σ_1^2; `sd2`: standard deviation for the normal proposal distribution for σ_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 σ_1^2/σ_2^2; `rand.shift`: probability of proposing a random shift in the tree (improves mixing); `print`: print frequency for the MCMC; `sample`: sample frequency.
`object`	for the S3 `summary` method, an object of class `"evol.rate.mcmc"`.
`...`	other optional arguments.

Details

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

Value

An object of class `"evol.rate.mcmc"` consisting of at least the following elements:

`mcmc`	results from the MCMC run.
`tips`	list of stips in rate σ_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. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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 variance-covariance matrix*

Description

This function takes an object of class `"simmap"` 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`	an object of class `"simmap"`. If `tree` is an object of class `"phylo"` then a simple multivariate Brownian motion model will be fit to the data in `X`.
`X`	an `n` x `m` matrix of tip values for `m` continuously valued traits in `n` species - row names should be species names. If `X` is supplied as a data frame it will be coerced into a matrix without warning.
`maxit`	an optional integer value indicating the maximum number of iterations for optimization. This quantity may need to be increased for difficult optimizations.
`vars`	an optional logical value indicating whether or not to estimate the variances of the parameter estimates from the Hessian matrix.
`...`	optional arguments. The most important optional argument at this time is `error_vcv` which should contain a list of matrices of sampling *variances* and covariances for (and between) the means of each species. The sampling variance for the mean is just the square of the sampling error. Sampling covariances will tend to be zero (or close to zero) if error for different traits is uncorrelated, for instance, because different specimens were used to estimate the means for different traits, and non-zero otherwise.

Details

This function performs optimization by maximizing 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. Users may have to increase `maxit` for large trees and/or more than two traits.

Value

An object of class `"evol.vcv"` 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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`evol.rate.mcmc`, `brownie.lite`

Examples

```
## load data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)
## fit multi-correlation model
sunfish.fit<-evol.vcv(sunfish.tree,sunfish.data[,2:3])
print(sunfish.fit)
```

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

Description

This function takes an object of class `"simmap"` with a mapped binary or multistate trait and data for two and only two continuously valued character. It then fits (by default) four different evolutionary models: common rates and correlation; different rates, common correlation; different correlations, common rates; no common structure.

In addition to the four default models specified above, `evolvcv.lite` now fits an additional four additional models.

The set of models to be fit can be specified using the optional argument `models` in multiple ways.

First, if left unspecified, then the four models listed above will be fit.

Second, if `models` is set to `"all models"` than eight models will be fit.

Lastly, one or more (up to all eight) models can be fit by encoding the models to be fit into a single vector containing a subset or all of the following elements: `"1"`, `"2"`, `"2b"`, `"3"`, `"3b"`, `"3c"`, and `"4"`. These codes correspond to the following eight models: 1. common rates, common correlation; 2. different rates, common correlation; 2b. different rates for trait 1 only, common correlation; 2c. different rates for trait 2 only, common correlation; 3. common rates, different correlations; 3b. different rates for trait 1 only, different correlations; 3c. different rates for trait 2 only, different correlation; and 4. no common structure.

Usage

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

Arguments

`tree`	an object of class `"simmap"`. If `tree` is an object of class `"phylo"` then a simple multivariate Brownian motion model will be fit to the data in `X`.
`X`	an `n x m` matrix of tip values for `m` continuously valued traits in `n` species - row names should be species names. If `X` is supplied as a data frame it will be coerced into a matrix without warning.
`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.
`...`	other optional arguments. The most important optional argument is probably `models` which species the models to be fit. See *Description* for more information. A second useful argument is `error_vcv` which should be supplied as a list of matrices of sampling *variances* and covariances for (and between) the means of each species. The sampling variance for the mean is just the square of the sampling error. Sampling covariances will tend to be zero (or close to zero) if error for different traits is uncorrelated, for instance, because different specimens were used to estimate the means for different traits, and non-zero otherwise.

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.

Revell, L. J., K. S. Toyama, and D. L. Mahler (Submitted) A simple hierarchical model for heterogeneity in the evolutionary correlation on a phylogenetic tree.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`brownie.lite`, `evol.vcv`

Examples

```
## load data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)
## fit heirarchical common-structure models
sunfish.fit<-evolvcv.lite(sunfish.tree,sunfish.data[,2:3])
print(sunfish.fit)
```

`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"`, implementing a branch-and-bound search (obviously), 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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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

See Also

`mrp.supertree`, `optim.parsimony`, `pratchet`

`expand.clade` *Expands (or contracts) the tip-spacing of a given clade or clades*

Description

The purpose of this function is to compute a custom tip-spacing for users who want to expand or contract the tip-spacing of the descendant taxa from a given node or nodes.

Usage

```
expand.clade(tree, node, factor=5)
## S3 method for class 'expand.clade'
plot(x, ...)
```

Arguments

`tree`	tree an object of class `"phylo"` or `"simmap"`.
`node`	node index or vector of node indices.
`factor`	expansion factor for the tip-spacing of the taxa descended from node or nodes in `node`.
`x`	for `plot` method, an object of class `"expand.clade"`.
`...`	optional arguments to be passed to `plotTree` or `plotSimmap`, depending on the class of `x$tree`.

Value

The function returns an object of class `"expand.clade"` which consists of the (possibly re-ordered) tree and a numerical vector with the calculated tip spacing based on the expansion factor specified by the user. This object can be plotted using the S3 `plot` method for the object class; or it can be plotted simplying by calling a standard plotting function on the tree & tip spacings.

Author(s)

Liam 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.

`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).

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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="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"`.) This option just calls the function `densityMap` internally.

If `type="contmap"`, reconstructed continuous trait evolution is mapped on the tree. Again, see `contMap` for additional arguments if `type="contmap"`. Much like `type="densitymap"`, this option just calls the function `contMap` internally.

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.

Finally, if `type="scattergram"` a phylogenetic scatter plot matrix containing `contMap` style trees on the diagonal and `phylomorphospace` plots in non-diagonal panels is produced. For this type a trait matrix `X` must also be supplied. The only additional arguments available for this type are `ftype`, `fsize`, `colors`, and `label`. (See `phylomorphospace` for details on how these arguments should be used.) This function calls `phyloScattergram` (which is also now exported to the name space) internally. In addition to creating a plot, `phyloScattergram` also returns an object of class `"phyloScattergram"` which can be replotted using different options if desired.

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`.

Finally, the optional argument `hold` will be passed to multiple methods if supplied. It is a logical value that indicates whether or not the output to the graphical device should be held using `dev.hold` before plotting (defaults to `hold=TRUE`).

Usage

```
fancyTree(tree, type=c("extinction","traitgram3d","droptip","densitymap",
   "contmap","phenogram95","scattergram"), ..., control=list())
phyloScattergram(tree, X=NULL, ...)
phenogram95(tree, x=NULL, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`type`	the type of special plot to create. See Description.
`...`	arguments to be passed to different methods. See Description.
`control`	a list of control parameters, depending on `type`.
`X`	in `phyloScattergram`, a matrix of continuous trait values. Row names in the matrix should correspond to species names in the tree.
`x`	in `phenogram95`, a named vector with values for a continuously distributed trait.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

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

Examples

```
# plot tree with extinction
set.seed(10)
tree<-pbtree(b=1,d=0.4,t=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)

## reset par to defaults
par(mar=c(5.1,4.1,4.1,2.1))
par(mfrow=c(1,1))
```

`fastAnc` *(Reasonably) fast estimation of ML ancestral states*

Description

This function performs (reasonably) 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 re-roots 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.
`...`	optional arguments. Presently this consists of `anc.states`, a named vector containing ancestral states to fix. Names should correspond to node numbers in the input tree.

Value

An object of class `"fastAnc"` consisting of either: 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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`ace`, `anc.Bayes`, `anc.ML`, `pic`

Examples

```
## load data from Garland et al. (1992)
data(mammal.tree)
data(mammal.data)
## extract character of interest
ln.bodyMass<-log(setNames(mammal.data$bodyMass,
    rownames(mammal.data)))
## estimate ancestral body sizes
fit.BM<-fastAnc(mammal.tree,ln.bodyMass,CI=TRUE)
print(fit.BM,println=10)
```

`fastBM` *(Reasonably) fast quantitative trait simulation on phylogenies*

Description

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

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, σ^2.
`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.
`...`	optional arguments `alpha` and `theta` used for OU simulation. If `alpha` is set then `mu` and `bounds` are ignored with a warning.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`sim.corrs`

Examples

```
## simulate 10 characters on the Anolis tree
## under Brownian motion
data(anoletree)
X<-fastBM(anoletree,nsim=10)
head(X)
```

`fastMRCA` *Get the MRCA (or height above the root of the MRCA) of a pair of tip taxa*

Description

This function returns the most recent common ancestor (node number) for a pair of taxa; or, in the case of `fastHeight`, the height above the root of the MRCA of a pair of taxa; or, in the case of `fastDist`, the patristic distance between a pair of taxa.

Usage

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

Arguments

`tree`	an object of class `"phylo"`.
`sp1`	species one name.
`sp2`	species two name.

Details

This function is mostly redundant with `findMRCA` (or `findMRCA(...,type="height")` in the case of `fastHeight`) but for very large trees will be considerably faster. (Also see `getMRCA` in the ape package.)

Value

The node number of the MRCA, the height above the root (for `fastHeight`), or the patristic distance between two taxa (for `fastDist`).

Author(s)

Liam 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.

See Also

`getMRCA`, `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 node number of the most recent common ancestor of a set of taxa. If `tips=NULL` the function is redundant with `mrca` (for `type="node"`) or `vcv.phylo`, but much slower (for `type="height"`).

Usage

```
findMRCA(tree, tips=NULL, type=c("node","height"))
```

Arguments

`tree`	a phylogenetic tree as an object of class `"phylo"`.
`tips`	a vector containing a set of tip labels.
`type`	either `"node"` to return the node of the MRCA; or `"height"` to return the height above the root of the MRCA of `tips`.

Details

If `tips==NULL` and `type="node"` (the default) it will return the result of a normal function call to `mrca`.

If `tips=NULL` and `type="height"` it will return a matrix equal to that produced by `vcv.phylo`.

From `phytools` `0.5-66` forward `findMRCA` uses `getMRCA` in the *ape* package internally, which results in a big speed-up. Even though the two functions are thus totally redundant I have left `findMRCA` in the package to ensure backward compatibility.

Value

The node number of the MRCA, or a matrix of node numbers (if `tips==NULL`) - for `type="node"`; or the height of the MRCA, or a matrix of heights (if `tips==NULL`) - for `type="height"`.

Author(s)

Liam 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.

See Also

`findMRCA`, `mrca`

Examples

```
data(anoletree)
anc<-findMRCA(anoletree,c("cristatellus","cooki",
    "gundlachi"))
plotTree(anoletree,type="fan",fsize=0.7,lwd=1)
nodelabels(node=anc,frame="circle",pch=21,cex=1.5,
    bg="blue")
legend("topleft","common ancestor of\nPuerto Rican TG anoles",
pch=21,pt.cex=1.5,pt.bg="blue",cex=0.7,bty="n")
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margin to default
```

`fit.bd` *Fits birth-death (speciation/extinction) model to reconstructed phylogeny*

Description

The function `fit.bd` fits a birth-death model to a phylogenetic tree with edge lengths and a (potentially) incomplete sampling fraction.

The function `fit.yule` fits a pure-birth model with a (potentially) incomplete sampling fraction.

The function `lik.bd` computes the likelihood of a set of birth & death rates given the set of branching times computed for a tree and a sampling fraction.

Usage

```
fit.bd(tree, b=NULL, d=NULL, rho=1, ...)
fit.yule(tree, b=NULL, d=NULL, rho=1, ...)
lik.bd(theta, t, rho=1, N=NULL)
## S3 method for class 'fit.bd'
print(x, ...)
```

Arguments

`tree`	object of class `"phylo"`.
`b`	birth (speciation) rate. Presently doesn't do anything as the rate cannot be fixed.
`d`	death (extinction) rate. Presently doesn't do anything as the rate cannot be fixed.
`rho`	sampling fraction.
`theta`	vector of `b` and `d` for likelihood function.
`t`	branching times for calculation of the likelihood.
`N`	number of tips in the tree.
`x`	object of class `"fit.bd"` for `print` method.
`...`	optional arguments.

Value

`fit.bd` returns an object of class `"fit.bd"` which can be printed. This object is a list containing the fitted model parameters, likelihood, optimization conditions, a summary of the optimization, and a likelihood function.

`fit.yule` returns an object of class `"fit.yule"`. This object is a list containing the fitted model parameter, likelihood, optimization conditions, a summary of the optimization, and a likelihood function.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Nee, S., May, R. M. and Harvey, P. H. (1994) The reconstructed evolutionary process. *Philosophical Transactions of the Royal Society of London B*, **344**, 305-311.

Stadler, T. (2012) How can we improve the accuracy of macroevolutionary rate estimates? *Systematic Biology*, **62**, 321-329.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`birthdeath`

Examples

```
data(salamanders)
## compute sampling fraction based on 55 species of Plethodon
sampling.f<-Ntip(salamanders)/55
## fit birth-death model
bd.fit<-fit.bd(salamanders,rho=sampling.f)
print(bd.fit)
## fit Yule model
yule.fit<-fit.yule(salamanders,rho=sampling.f)
print(yule.fit)
## compare b-d and yule models
anova(yule.fit,bd.fit)
## create a likelihood surface for b-d model
ngrid<-100
b<-seq(0.01,0.06,length.out=ngrid)
d<-seq(0.005,0.03,length.out=ngrid)
logL<-sapply(d,function(d,b) sapply(b,function(b,d)
    bd.fit$lik(c(b,d)),d=d),b=b)
contour(x=b,y=d,logL,nlevels=100,
    xlab=expression(lambda),
    ylab=expression(mu),bty="l")
title(main="Likelihood surface for plethodontid diversification",
    font.main=3)
points(bd.fit$b,bd.fit$d,cex=1.5,pch=4,
```

```
    col="blue",lwd=2)
legend("bottomright","ML solution",pch=4,col="blue",
    bg="white",pt.cex=1.5,pt.lwd=2)
```

`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

An object of class `"fitBayes"` that includes a matrix (`mcmc`) with a 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. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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)
## S3 method for class 'fitDiversityModel'
logLik(object, ...)
## S3 method for class 'fitDiversityModel'
print(x, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`x`	a vector with tip values for a continuously distributed trait. For `print` method, an object of class `"fitDiversityModel"`.
`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.
`object`	for `logLik` method, an object of class `"fitDiversityModel"`.
`...`	optional arguments for `logLik` and `print` methods. Note that for the `logLik` method the number of fitted parameters (`"df"`) is assumed to be 3 for the diversity dependent model (that is, if `psi` is estimated) and 2 for the diversity independent model, unless otherwise specified (using the argument `df`).

Value

An object of class `"fitDiversityModel"` consisting of the following components:

`logL`	log-likelihood of the fitted model.
`sig0`	estimated starting value for the rate at the root of the tree, σ_0^2.
`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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`brownie.lite`, `estDiversity`, `evol.rate.mcmc`

`fitMk` *Fits Mk model*

Description

The function `fitMk` fits a so-called extended M*k* model for discrete character evolution (Lewis, 2001).

`plot.fitMk` plots an object of class `"fitMk"` returned by `fitMk`. `plot.gfit` plots an object of class `"gfit"` from geiger's `fitDiscrete` function. Both plots portray the fitted model using a graph of arrows connecting states.

The function `fitmultiMk` fits an M*k* model in which the transition rates between character states are allowed to vary depending on the mapped state of a discrete character on the tree. It can be combined with, for example, `paintSubTree` to test hypotheses about how the process of discrete character evolution for `x` varies between different parts of the tree.

The function `fitpolyMk` fits an M*k* model to data for a discrete character with intraspecific polymorphism. Polymorphic species should be coded with the name of the two or more states recorded for the species separated by a space (e.g., `A+B` would indicate that both states `A` and `B` are found in the corresponding taxon). Invariably it's assumed that transitions between states must occur through a polymorphic condition, whereas transitions *cannot* occur directly between two incompatible polymorphic conditions. For instance, a transition between `A+B` and `B+C` would have to occur through the monomorphic state `B`. At time of writing, this function permits the models `"ER"` (equal rates for all permitted transitions), `"SYM"` (symmetric backward & forward rates for all permitted transitions), `"ARD"` (all-rates-different for permitted transitions), and a new model called `"transient"` in which the acquisition of polymorphism (e.g., `A` -> `A+B`) is assumed to occur at a different rate than its loss (e.g., `A+B` -> `B`). The method `plot.fitpolyMk` plots the fitted M*k* model with intraspecific polymorphism.

The function `mcmcMk` runs a Bayesian MCMC version of `fitMk`. The shape of the prior distribution of the transition rates is Γ, with α and β via the argument `prior`, which takes the form of a list. The default value of α is 0.1, and β defaults to a value such tha α/β is equal to the parsimony score for `x` divided by the sum of the edge lengths of the tree. The shape of the proposal distribution is normal, with mean zero and a variance that can be controlled by the user via the optional argument `prior.var`. The argument `auto.tune`, if `TRUE` or `FALSE`, indicates whether or not to 'tune' the proposal variance up or down to target a particular acceptance rate (defaults to 0.5). `auto.tune` can also be a numeric value between 0 and 1, in which case this value will be the target acceptance ratio. The argument `plot` indicates whether the progress of the MCMC should be plotted (defaults to `TRUE`, but runs much faster when set to `FALSE`).

The method `plot.mcmcMk` plots a log-likelihood trace and a trace of the rate parameters from the MCMC. (This the samem graph that is created by setting `plot=TRUE` in `mcmcMk`.) The method `density.mcmcMk` computes a posterior density on the transition rates in the model from the posterior sample obtained in the MCMC, will import the package *coda* if it is available, and returns an object of class `"density.mcmcMk"`. Finally, the method `plot.density.mcmcMk` creates a plot of the posterior density (or a set of plots) for the transition rates between states.

Finally, the function `fitHRM` fits a hidden-rate M*k* model following Beaulieu et al. (2013). For the hidden-rate model we need to specify a number of rate categories for each level of the trait - and this can be a vector of different values for each trait. We can also choose a model (`"ER"`, `"SYM"`, or `"ARD"`), as well as whether or not to treat the character as a 'threshold' trait (`umbral=TRUE`, defaults to `FALSE`). This latter model is basically one that allows absorbing conditions for some hidden states. Since this can be a difficult optimization problem, the optional argument `niter` sets the number of optimization iterations to be run. `niter` defaults to `niter=10`.

Usage

```
fitMk(tree, x, model="SYM", fixedQ=NULL, ...)
## S3 method for class 'fitMk'
plot(x, ...)
## S3 method for class 'gfit'
plot(x, ...)
fitmultiMk(tree, x, model="ER", ...)
fitpolyMk(tree, x, model="SYM", ordered=FALSE, ...)
graph.polyMk(k=2, model="SYM", ordered=FALSE, ...)
## S3 method for class 'fitpolyMk'
plot(x, ...)
mcmcMk(tree, x, model="ER", ngen=10000, ...)
## S3 method for class 'mcmcMk'
plot(x, ...)
## S3 method for class 'mcmcMk'
density(x, ...)
## S3 method for class 'density.mcmcMk'
```

```
plot(x, ...)
fitHRM(tree, x, model="ARD", ncat=2, ...)
## S3 method for class 'fitHRM'
plot(x, ...)
fitMk.parallel(tree, x, model="SYM", ncores=1, ...)
```

Arguments

`tree`	an object of class `"phylo"`. In the case of `fitmultiMk` an object of class `"simmap"` with a mapped discrete character.
`x`	a vector of tip values for species; `names(x)` should be the species names. In the case of `plot` and `density` methods, an object of the appropriate class.
`model`	model. See `make.simmap` or `ace` for details.
`fixedQ`	fixed value of transition matrix `Q`, if one is desired.
`ordered`	for `fitpolyMk`, a logical value indicating whether or not the character should be treated as ordered. For now the function assumes alphanumerical order (i.e., numbers sorted by their initial and then successive digits followed by characters or character strings in alphabetical order).
`k`	For `graph.polyMk`, the number of monomorphic states for the discrete trait.
`ngen`	number of generations of MCMC for `mcmcMk`.
`ncat`	number of rate categories (per level of the discrete trait) in the hidden-rate model.
`ncores`	number of cores for `fitMk.parallel`.
`...`	optional arguments, including `pi`, the prior distribution at the root node (defaults to `pi="equal"`). Other options for `pi` include `pi="fitzjohn"` (which implements the prior distribution of Fitzjohn et al. 2009), `pi="estimated"` (which finds the stationary distribution of state frequencies and sets that as the prior), or an arbitrary prior distribution specified by the user. For `plot` method optional arguments include (but may not be limited to): `signif`, the number of digits for the rates to be plotted; `main`, a character vector of length two with the headings for each subplot; `cex.main`, `cex.traits`, and `cex.rates`, font sizes for the various text elements of the plot; and `show.zeros`, a logical argument specifying whether or not to plot arrows with the ML estimated transition rate is not different from zero (with tolerance specified by the optional argument `tol`). Finally, for `fitpolyMk`, `max.poly` can be set for the `ordered=TRUE` model. `max.poly` defaults to the highest level of polymorphism observed in the data.

Details

Note that both `fitMk` and `fitmultiMk` recycle code from `ace` in the *ape* package for computing the likelihood. `fitpolyMk`, `mcmcMk`, and `fitHRM` use `fitMk` internally to compute the likelihood.

Value

An object of class `"fitMk"`, `"fitmultiMk"`, `"fitpolyMk"`, `"mcmcMk"`, or `"fitHRM"`. In the case of `density.mcmcMk` an object of class `"density.mcmcMk"`.

`plot.fitMk`, `plot.gfit`, and `plot.HRM` invisibly return the coordinates of vertices of the plotted **Q**-matrix.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Beaulieu, J. M., B. C. O'Meara, and M. J. Donoghue (2013) Identifying hidden rate changes in the evolution of a binary morphological character: The evolution of plant habit in campanulid angiosperms. *Systematic Biology*, **62**, 725-737.

Fitzjohn, R. G., W. P. Maddison, and S. P. Otto (2009) Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. *Systematic Biology*, **58**, 595-611.

Lewis, P. O. (2001) A likelihood approach to estimating phylogeny from discrete morphological character data. *Systematic Biology*, **50**, 913-925.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`ace`, `make.simmap`

Examples

```
## load tree and data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)
## extract discrete character (feeding mode)
fmode<-setNames(sunfish.data$feeding.mode,
     rownames(sunfish.data))
## fit "ER" model
fit.ER<-fitMk(sunfish.tree,fmode,model="ER")
print(fit.ER)
## fit "ARD" model
fit.ARD<-fitMk(sunfish.tree,fmode,model="ARD")
print(fit.ARD)
## compare the models
AIC(fit.ER,fit.ARD)

## load tree and data from Benitez-Alvarez et al. (2000)
data(flatworm.data)
data(flatworm.tree)
## extract discrete character (habitat)
habitat<-setNames(flatworm.data$Habitat,
rownames(flatworm.data))
## fit polymorphic models "ER" and "transient"
fitpoly.ER<-fitpolyMk(flatworm.tree,habitat,
model="ER")
fitpoly.transient<-fitpolyMk(flatworm.tree,habitat,
model="transient")
```

```
## print fitted models
print(fitpoly.ER)
print(fitpoly.transient)
## compare model
AIC(fitpoly.ER,fitpoly.transient)
## plot models
par(mfrow=c(2,1))
plot(fitpoly.ER)
mtext("a) ER polymorphic model",adj=0,line=1)
plot(fitpoly.transient)
mtext("b) Transient polymorphic model",adj=0,
line=1)
par(mfrow=c(1,1))
```

`fitPagel` *Function to test for correlated evolution of binary traits*

Description

This function fit's Pagel's (1994) model for the correlated evolution of two binary characters.

`plot.fitPagel` plots the fitted models using arrows.

Usage

```
fitPagel(tree, x, y, method="fitMk", model="ARD", dep.var="xy", ...)
## S3 method for class 'fitPagel'
plot(x, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`x`	a vector of phenotypic values for a binary trait for the species in `tree`; or a matrix in which the rows of `x` give the probability of being in each column state. (The latter option is only supported for `method="fitMk"`.) For S3 `plot` method, an object of class `"fitPagel"`.
`y`	a second binary character for the species in `tree`; or a matrix in which the rows give the probability of being in each column state.
`method`	function to use for optimization (defaults to `method="fitMk"`). Other options are `"ace"` to use the `ace` function in ape for optimization, or to `"fitDiscrete"` (if the geiger package is installed) to use geiger's `fitDiscrete` for optimization.
`model`	model of evolution for the individual characters. Can be `model="ER"`, `"SYM"` (equivalent to `"ER"` in this case), and `"ARD"`.
`dep.var`	dependent variable. If `dep.var="xy"` than the rate of subsitution in `x` depends on `y` & vice versa. If `dep.var="x"` than the substitution rate in `x` depends on `y`, but not the converse. Finally, if `dep.var="y"` than the rate of substitution in `y` depends on `x`, but not the converse.

`...`	optional arguments to be passed to `fitMk`, `ace`, or `fitDiscrete`. For `plot` method optional arguments include (but may not be limited to): `signif`, the number of digits for the rates to be plotted; `main`, a character vector of length two with the headings for each subplot; `cex.main`, `cex.sub`, `cex.traits`, and `cex.rates`, font sizes for the various text elements of the plot; and `lwd.by.rate`, a logical argument specifying whether or not to scale arrow line widths in proportion to the estimated rates.

Value

An object of class `"fitPagel"` which contains the optimized matrices under an independence & a dependence model, log-likelihoods, a likelihood ratio, and a P-value for the independence model based on a chi-squared test.

`plot.fitPagel` creates a plot showing the different fitted models with arrows.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Pagel, M. (1994) Detecting correlated evolution on phylogenies: A general method for the comparative analysis fo discrete characters. *Proceedings of the Royal Society B*, **255**, 37-45.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`ace`, `fitMk`, `make.simmap`

`force.ultrametric` *Forces a phylogenetic tree to be ultrametric*

Description

This function coerces an object of class `"phylo"` to be ultrametric.

This is achieved either by using `nnls.tree` from the phangorn package to compute the set of edge lengths that result in a minimized sum-of-squares distance between the patristic distance of the output and input trees (`method="nnls"`); or by simply extending all the external edges of the tree to match the external edge with the greatest total height (`method="extend"`).

Note that neither of these should be treated as formal statistical methods for inferring an ultrametric tree. Rather, this method can be deployed when a genuinely ultrametric tree read from file fails `is.ultrametric` for reasons of numerical precision.

Usage

```
force.ultrametric(tree, method=c("nnls","extend"), ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`method`	the method to use to force the tree to be ultrametric. Options are `"nnls"` (which uses the phangorn function `nnls.tree` internally), or `"extend"`.
`...`	optional arguments: principally, `message`. This argument (if set to `FALSE`) can be used to suppress the default warning message that `force.ultrametric` *should not* be used as a formal statistical method to ultrametricize a tree.

Value

An ultrametric tree in an object of class `"phylo"`.

Author(s)

Liam 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.

See Also

`is.ultrametric`, `nnls.tree`

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

Description

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

Usage

```
gammatest(x)
```

Arguments

`x`	an object of class `"ltt"` resulting from a call of the function `ltt`.

Value

A an object of class `"gammatest"` consisting of a list that contains:

`gamma`	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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`ltt`, `mccr`

Examples

```
tree<-pbtree(n=200)
gammatest(ltt(tree,plot=FALSE))
```

`genSeq` *Simulate a DNA alignment on the tree under a model*

Description

Simulates DNA sequence on `tree` under the specified model. Uses `sim.Mk` internally.

Usage

```
genSeq(tree, l=1000, Q=NULL, rate=1, format="DNAbin", ...)
```

Arguments

`tree`	object of class `"phylo"`.
`l`	length of desired sequences.
`Q`	transition matrix for the simulation. Row and column names (`c("a","c","g","t")`, although not necessarily in that order) should be provided. If `NULL`, a single rate is assumed.
`rate`	multiplier for `Q`, or a vector for Γ rate heterogeneity.
`format`	format of the output object. Can be `"DNAbin"`, `"phyDat"`, or `"matrix"`.
`...`	optional arguments.

Value

An object of class `"DNAbin"` or `"phyDat"`, or a matrix of nucleotides.

Author(s)

Liam 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.

Examples

```
data(mammal.tree)
mammal.tree$edge.length<-mammal.tree$edge.length/
    max(nodeHeights(mammal.tree))*0.2 ## rescale tree
## simulate gamma rate heterogeneity
gg<-rgamma(n=100,shape=0.25,rate=0.25)
dna<-genSeq(mammal.tree,l=100,rate=gg)
```

`geo.legend` *Adds a geological (or other temporal) legend to a plotted tree*

Description

The function `geo.legend` adds a geological (or other temporal) legend to a plotted tree.

The function `geo.palette` returns a geological time color palette to the user.

Usage

```
geo.legend(leg=NULL, colors=NULL, alpha=0.2, ...)
geo.palette()
```

Arguments

`leg`	a matrix with the starting & ending point of each plotted era in rows, & names of the time periods as rownames.
`colors`	a vector of colors for the time periods of the rows in `leg`.
`alpha`	transparency level to apply to `colors`.
`...`	optional arguments.

Value

`geo.legend` adds a visual element to a plotted tree and invisible returns an object of class `geo.legend` containing the time periods and colors of the painted legend.

`geo.palette` simply returns a geological timescale color palette as an object of class `"geo.palette"`.

Author(s)

Liam 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.

`get.treepos` *Get position or node of a plotted tree interactively*

Description

Both functions return the phylogenetic position of a mouse click on a plotted tree.

`get.treepos` returns the index of the node at the end of the selected edge, along with the branch distance to that node.

`getnode` simply returns the closest node to the user mouse click.

Usage

```
get.treepos(message=TRUE, ...)
getnode(...)
```

Arguments

`message`	for `get.treepos`, a logical value indicating whether or not to print an instructional message.
`...`	optional arguments.

Details

Both functions are primarily meant to be used internally by other *phytools* functions.

Value

A list for `get.treepos` and a node number for `getnode`.

Author(s)

Liam 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.

`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>`

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`extract.clade`, `getDescendants`

`getDescendants` *Get descendant node numbers*

Description

`getDescendants` returns the set of node & tip numbers descended from `node`.

`getParent` returns the *single* parent node of a specified node number (or `NULL` if `node` is already the root).

Usage

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

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, or for `getParent` the single node preceding `node` in the tree.

Author(s)

Liam 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.

See Also

`Descendants`, `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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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 applicable.

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"` then this function returns a list containing up to two vectors: one for the node numbers of labels of sister nodes (if applicable); and the other containing the tip labels of the sister tips.

Author(s)

Liam 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.

See Also

`getDescendants`, `Siblings`

`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","both"))
```

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

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

`labelnodes` *Function to interactively label nodes of a plotted tree*

Description

Function adds node labels to a plotted object of class `"phylo"`. The nodes to be labels can be selected interactively by the user (i.e., by clicking on the corresponding nodes of the plotted tree).

Usage

```
labelnodes(text, node=NULL, interactive=TRUE, shape=c("circle","ellipse",
   "rect"), ...)
```

Arguments

`text`	text string or vector to be used as labels.
`node`	node numbers (indices) for the labels.
`interactive`	logical value indicating whether or not nodes should be supplied interactively. (I.e., by clicking on the nodes.)
`shape`	shape to plot around the plotted node label(s).
`...`	optional arguments.

Value

Invisibly returns a vector of the node indices for the labeled nodes.

Author(s)

Liam 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.

See Also

`cladelabels`, `nodelabels`

`ladderize.simmap` *Ladderize a tree with a mapped discrete character*

Description

This function 'ladderizes' an object of class `"phylo"` with a mapped discrete character. For more information see `ladderize`.

Usage

```
ladderize.simmap(tree, right=TRUE)
```

Arguments

`tree`	an object of class `"simmap"`.
`right`	a logical specifying how the tree should be ladderized.

Value

A ladderized object of class `"simmap"`.

Author(s)

Liam 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.

See Also

`make.simmap`, `ladderize`

`lambda.transform` *λ transformation of matrix*

Description

Function multiplies the off-diagonals of a square matrix by λ. 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`.

Value

An among-species phylogenetic variance covariance matrix (e.g., `vcv.phylo`) in which the off-diagonal elements have been multiplied by λ.

Author(s)

Liam 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.

`likMlambda` *Likelihood for joint λ*

Description

Computes the joint likelihood of Pagel's λ parameter.

Usage

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

Arguments

`lambda`	scalar, usually on the interval 0,1 (although not required to be).
`X`	data for various continuous character, in the form of a matrix.
`C`	n x n matrix (for n taxa) containing the height above the root for each pair of taxa in the tree (e.g., `vcv.phylo`).

Details

Generally intended to be used internally by other methods that do joint optimization of λ (e.g., `phyl.pca`).

Value

The log-likelihood.

Author(s)

Liam 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.

`linklabels` *Function to add tip labels to a plotted tree with linking lines*

Description

Function adds tip labels to a plotted tree by drawing curved, bent, or straight linking lines.

Usage

```
linklabels(text,tips,link.type=c("bent","curved","straight"),
   ...)
```

Arguments

`text`	text string or vector to be used as labels.
`tips`	node numbers (indices) for the tips to be labeled.
`link.type`	manner in which to draw the linking lines.
`...`	optional arguments, including `cex`, `lty`, `lwd`, and `col`.

Details

The idea underlying this function is that the user should first plot the tree without tip labels, but set the area of the plotting device to be sufficient to accommodate the tip labels once they have been added. The function then can be called to add tip labels connected by linking lines to the tips of the plotted tree.

Value

This function annotates a plot.

Author(s)

Liam 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.

See Also

`cladelabels`, `nodelabels`, `tiplabels`

`locate.fossil` *Locate a fossil lineage in a tree using continuous characters*

Description

This function uses ML to place a fossil lineage into a tree using continuous traits.

Usage

```
locate.fossil(tree, X, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`X`	a matrix with continuous character data.
`...`	optional arguments including `time.constraint` which can be a scalar (positive height above the root of the fossil or negative time before present) or a vector (age range of fossil, either positive or negative); `edge.constraint`, which is equivalent to `constraint` in `locate.yeti`; `plot`, `rotate`, and `quiet`, which have the same interpretation (and defaults) as the equivalent arguments in `locate.yeti`.

Value

Optimized tree as an object of class `"phylo"`.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Felsenstein, J. (1981) Maximum likelihood estimation of evolutionary trees from continuous characters. *American Journal of Human Genetics*, 25, 471-492.

Felsenstein, J. (2002) Quantitative characters, phylogenies, and morphometrics. In: MacLeod, N. and P. Forey (Eds.) *Morphology, Shape and Phylogeny* (pp. 27-44). Taylor and Francis, London.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J., D. L. Mahler, R. G. Reynolds, and G. J. Slater. (2015) Placing cryptic, recently extinct, or hypothesized taxa into an ultrametric phylogeny using continuous, character data: A case study with the lizard *Anolis roosevelti*. *Evolution*, **69**, 1027-1035.

`locate.yeti` *Locate a cryptic, recently extinct, or missing taxon on a tree*

Description

This function uses ML (or REML) to place a recently extinct, cryptic, or missing taxon on an ultrametric (i.e., time-calibrated) phylogeny.

Usage

```
locate.yeti(tree, X, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`X`	a matrix with continuous character data.
`...`	optional arguments including: `method` (`"ML"` or `"REML"`, defaults to `"ML"`); `search` (`"heuristic"` or `"exhaustive"`, defaults to `"heuristic"`); `constraint`, a vector containing the daughter node numbers from `tree$edge` for each edge to try; `plot` a logical argument specifying whether or not to plot the likelihood profile on edges (defaults to `FALSE`); `rotate` a logical indicating whether or not to rotate the data based on the input tree; and `quiet`, which is logical and has an obvious interpretation.

Value

Optimized tree as an object of class `"phylo"`.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Felsenstein, J. (1981) Maximum likelihood estimation of evolutionary trees from continuous characters. *American Journal of Human Genetics*, 25, 471-492.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J., D. L. Mahler, R. G. Reynolds, and G. J. Slater. (2015) Placing cryptic, recently extinct, or hypothesized taxa into an ultrametric phylogeny using continuous, character data: A case study with the lizard *Anolis roosevelti*. *Evolution*, **69**, 1027-1035.

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

Description

Computes the least squares branch lengths conditioned on a topology and distance matrix. Generally intended as a function to be used internally by `optim.phylo.ls`.

Usage

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

Arguments

`tree`	phylogeny.
`D`	distance matrix.

Value

A tree with branch lengths.

Author(s)

Liam 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.

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

Description

The function `ltt` computes LTT plot with extant and extinct lineages, and optionally conducts γ-test of Pybus & Harvey (2000). The object returned by `ltt` can be plotted or re-plotted using `plot`.

The function `gtt` computes the value of Pybus & Harvey's γ statistic through time by slicing the tree at various points - by default in even intervals from the time above the root at which $N = 3$ to the present day.

The function `mccr` performs the MCCR test of Pybus & Harvey (2000) which takes into account incomplete taxon sampling in computing a P-value of the γ statistic.

Usage

```
ltt(tree, ...)
## S3 method for class 'phylo'
ltt(tree, plot=TRUE, drop.extinct=FALSE, log.lineages=TRUE, gamma=TRUE, ...)
## S3 method for class 'multiPhylo'
ltt(tree, drop.extinct=FALSE, gamma=TRUE, ...)
## S3 method for class 'simmap'
ltt(tree, plot=TRUE, log.lineages=FALSE, gamma=TRUE, ...)
## S3 method for class 'multiSimmap'
ltt(tree, gamma=TRUE, ...)
gtt(tree, n=100, ...)
mccr(obj, rho=1, nsim=100, ...)
```

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 γ from Pybus & Harvey (2000; *Proc. Roy. Soc. B*).
`n`	for `gtt` the number of time intervals to use to track γ through time.
`obj`	for `mccr` an object of class `"ltt"`.
`rho`	for `mccr` sampling fraction.
`nsim`	for `mccr` number of simulations to use for the MCCR test.
`...`	other arguments to be passed to plotting methods. See `plot.default`.

Details

Although it is calculated here, it's unclear how to interpret the γ-statistic if not all the tips in the tree are contemporaneous.

Value

`ltt` returns an object of class `"ltt"` which includes the following components:

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

If `tree` is an object of class `"multiPhylo"`, then an object of class `"multiLtt"` is returned consisting of a list of object of class `"ltt"`.

`gtt` returns an object of class `"gtt"`.

`mccr` returns of object of class `"mccr"`.

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`gammatest`, `ltt95`

Examples

```
## LTT plots
set.seed(99)
trees<-pbtree(n=100,scale=100,nsim=10)
obj<-ltt(trees,plot=FALSE)
plot(obj,log="y",log.lineages=FALSE,
    bty="l")
title(main="LTT plots for 10 pure-birth trees",
    font.main=3)
tree<-pbtree(b=1,d=0.25,t=4)
obj<-ltt(tree,gamma=FALSE,show.tree=TRUE,
    bty="l")
title(main="LTT plot with superimposed tree",
    font.main=3)
obj
## GTT plot
data(anoletree)
anole.gtt<-gtt(anoletree,n=40)
plot(anole.gtt)
```

`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"), ...)
## S3 method for class 'ltt95'
plot(x, ...)
```

Arguments

`trees`	is an object of class `"multiPhylo"` containing a list of phylogenetic trees.
`alpha`	confidence level. Defaults to `alpha=0.05`. `alpha=0` will mean that the interval around *all* trees in the set will be plotted.
`log`	logical value indicating whether or not to plot on the semi-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.
`x`	object of class `"ltt95"` for plotting method.
`...`	optional arguments to be used by `ltt95` or the plotting method. So far: `res` gives the number of time-steps (defaults to `res=100`); `xaxis` (`"standard"`, `"negative"`, or `"flipped"`) determines the scale (time from the root, time back from the present, or time from the present) of the x-axis of the plot; `lend` determines the line end type (as in `par`); `shaded` determines whether to plot the (1-α)-percent CI using dotted lines (if `FALSE`) or shading (if `TRUE`); and `bg` is the background color for shading if `shaded=TRUE`.

Details

This function creates a plot and invisibly returns an object of class `"ltt95"`.

Author(s)

Liam 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.

See Also

`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`	an object of class `"phylo"`.
`limits`	a vector containing the temporal limits, in time since the root node of the tree, for the mappings. The first number should be `0`, and each subsequent number should be the start of each subsequent regime or era to be mapped on the tree.
`...`	optional arguments.

Value

An object of class `"simmap"` with the specified eras mapped as different regimes.

Author(s)

Liam 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.

See Also

`make.simmap`, `read.simmap`, `plotSimmap`

Examples

```
tree<-pbtree(n=1000,scale=100)
tree<-make.era.map(tree,c(0,25,50,75))
plot(tree,ftype="off",lwd=1)
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margins to default
```

`make.simmap` *Simulate stochastic character maps on a phylogenetic tree or trees*

Description

This function performs stochastic mapping using several different alternative methods.

For `Q="empirical"`, `make.simmap` 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"`, `make.simmap` 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, `make.simmap` 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 and character states as column names. The names (if `x` is a vector) or row names (if `x` is a matrix) should match the tip labels of the tree. The vector can be of class `"factor"`, `"character"`, or `"numeric"` (although in the lattermost case its content should obviously be only integer values).
`model`	a character string containing the model or a transition model specified in the form of a matrix. See `ace` for more details.
`nsim`	number of simulations. If `tree` is an object of class `"multiPhylo"`, then `nsim` simulations will be conducted *per* input tree.
`...`	optional arguments. So far, `pi` gives the prior distribution on the root node of the tree. Acceptable values for `pi` 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. If `pi="fitzjohn"`, then the Fitzjohn et al. (2009) root prior is used. Finally, if `pi` is a numeric vector then the root state will be sampled from this vector. The function 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. `message` defaults to `TRUE`. For optional argument `Q="mcmc"` (see below) 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`). Optional argument `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 sampled value of Q. Optional argument `vQ` can consist of a single numeric value or a vector containing the variances of the (normal) proposal distributions 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` is a list containing `alpha` and `beta` parameters for the Γ 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 the transition matrix Q. As for `vQ`, the order of `prior` is assumed to correspond with the order of `index.matrix` as 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

`make.simmap` uses code that has been adapted from *ape*'s function `ace` (by Paradis et al.) to perform Felsenstein's pruning algorithm to 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. Giorgio Bianchini pointed out that in *phytools* 1.0-1 (and probably prior recent versions) there was an error sampling the state at the root node of the tree based on the input prior (`pi`) supplied by a user – except for `pi="equal"` (a flat prior, the default) or for a prior distribution in which one or another state was known to be the global root state (e.g., `pi=c(1,0)`, `pi=c(0,1)`, etc.). All of these issues should be fixed in the current and all later versions.

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

Value

A object of class `"simmap"` or `"multiSimmap"` which consists of an object of class `"phylo"` (or a list of such objects with 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.
`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

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

Fitzjohn, R. G., W. P. Maddison, and S. P. Otto (2009) Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. *Systematic Biology*, **58**, 595-611.

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

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`brownie.lite`, `brownieREML`, `countSimmap`, `describe.simmap`, `evol.vcv`, `plotSimmap`, `read.simmap`, `write.simmap`

Examples

```
## Not run:
## load tree and data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)
## extract discrete character (feeding mode)
fmode<-setNames(sunfish.data$feeding.mode,
    rownames(sunfish.data))
## do stochastic mapping
smap.trees<-make.simmap(sunfish.tree,fmode,model="ER",
     nsim=100)
## print a summary of the stochastic mapping
summary(smap.trees)
## plot a posterior probabilities of ancestral states
cols<-setNames(c("blue","red"),levels(fmode))
plot(summary(smap.trees),colors=cols,ftype="i")
legend("topleft",c("non-piscivorous","piscivorous"),
   pch=21,pt.bg=cols,pt.cex=2)
par(mar=c(5.1,4.1,4.1,2.1))
## plot posterior density on the number of changes
plot(density(smap.trees),bty="l")
title(main="Posterior distribution of changes of each type",
    font.main=3)

## End(Not run)
```

`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. In `map.overlap` it will compute a single quantity giving the overall similarity of the maps, consequently this measure only makes sense of some or all of the states are shared between the two mapped tress. In `Map.Overlap` what is computed instead is a matrix in which the rows correspond to the states observed in `tree1` and the columns give the states for `tree2`, with the numerical values of the matrix showing the total overlap between each pair of mapped states on the two trees.

Usage

```
map.overlap(tree1, tree2, tol=1e-6, ...)
Map.Overlap(tree1, tree2, tol=1e-06, standardize=TRUE, ...)
```

Arguments

`tree1`	an object of class `"simmap"`.
`tree2`	an object of class `"simmap"`.
`tol`	an optional tolerance value.
`standardize`	for `Map.Overlap`, a logical value indicating whether or not to standardize overlap by dividing by the summed branch length of the tree.
`...`	optional arguments, such as `check.equal`, a logical value indicating whether or not to check if `tree1` and `tree2` match in underlying topology and branch lengths (they should). This value is `TRUE` by default, but can be set to `FALSE` if `tree1` and `tree2` are known to be equal to speed up calculation.

Value

A numerical value on the interval (0, 1), for `map.overlap`; or a matrix whose elements should sum to 1.0 (`Map.Overlap`).

Author(s)

Liam 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.

See Also

`make.simmap`, `read.simmap`

`map.to.singleton` *Converts a tree without singletons to a tree with singleton nodes*

Description

The function `map.to.singleton` takes an object of class `"simmap"` with a mapped discrete character and converts it to a tree with singleton nodes, in which edge has only one state. The states for each edge are stored in `names(tree$edge.length)`. In a sense this is just an alternative way to use the general structure of the `"phylo"` object to encode a tree with a mapped character.

`plotTree.singletons` plots a tree with singleton nodes. Note that `plotTree` and `plot.phylo` now have no trouble graphing trees with singleton nodes - but they do this by just ignoring the singletons. `plotTree.singletons` marks the singletons as nodes on the plotted phylogeny.

`drop.tip.singleton` drops tips from the tree leaving ancestral nodes for all remaining tips as singletons.

Finally, `rootedge.to.singleton` converts a tree with a root edge to a tree with a singleton node instead.

Usage

```
map.to.singleton(tree)
plotTree.singletons(tree)
drop.tip.singleton(tree, tip)
rootedge.to.singleton(tree)
```

Arguments

`tree`	an object of class `"simmap"` (for `map.to.singleton`, or a tree with one or more singleton nodes (for `plotTree.singletons`, `drop.tip.singleton`, and `rootedge.to.singleton`).
`tip`	for `drop.tip.singleton`, a tip label or vector of tip labels.

Value

An object of class `"phylo"` with singleton nodes. `plotTree.singletons` graphs a tree in which the singleton nodes are shown.

If `names(tree$edge.length)!=NULL` `plotTree.singletons` will use a different color from `palette` for each mapped state.

Author(s)

Liam 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.

See Also

`collapse.singles`, `drop.tip`, `make.simmap`

`mapped.states` *Returns a vector, matrix, or list of the mapped states on a tree or set of trees*

Description

Computes and orders a vector, matrix, or list of the unique mapped states on a tree or state of trees of class `"simmap"` or `"multiSimmap"`.

Usage

```
mapped.states(tree, ...)
```

Arguments

`tree`	a single tree or a set of trees as an object of class `"simmap"` or `"multiSimmap"`, respectively.
`...`	optional arguments.

Value

A vector, matrix, or list.

Author(s)

Liam 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.

`markChanges` *Add marked changes to a plotted tree with mapped discrete character*

Description

This function adds the reconstructed changes to a plotted tree with a stochastically mapped discrete character.

Usage

```
markChanges(tree, colors=NULL, cex=1, lwd=2, plot=TRUE)
```

Arguments

`tree`	an object of class `"simmap"`.
`colors`	a named vector of colors used to plot the stochastically mapped character on the tree.
`cex`	expansion factor for line height.
`lwd`	line width.
`plot`	logical value indicating whether the changes should be plotted or not.

Value

This function returns (invisibly) a matrix containing the x & y coordinates of the marked changes on the plotted tree.

Author(s)

Liam 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.

See Also

`plotSimmap`

`matchNodes` *Matches nodes between two trees*

Description

This function returns a matrix in which the first column contains *all* of the internal nodes of `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.

`matchLabels` is functionally equivalent but matches node (tip) indices based on identifying matching in the labels only.

Usage

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

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 also permitted.
`...`	optional arguments which may or may not be used depending on the value of `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.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`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"` and define the new the state `"not-A"`.

Usage

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

Arguments

`tree`	an object of class `"simmap"` or `"multiSimmap"` containing one or more phylogenetic trees with a mapped discrete character.
`old.states`	states to merge.
`new.state`	name for new state.

Value

An object of class `"simmap"` or `"multiSimmap"`.

Author(s)

Liam 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.

See Also

`make.simmap`, `read.simmap`

`midpoint.root` *Midpoint root a phylogeny*

Description

This function midpoint roots a rooted or unrooted tree (Farris 1972).

Usage

```
midpoint.root(tree)
```

Arguments

`tree`	an object of class `"phylo"`.

Details

Midpoint rooting involves locating the midpoint of the longest path between any two tips and putting the root in that location.

This function performs the same operation as `midpoint` in the phangorn package, but uses no phangorn code internally.

Value

An object of class `"phylo"` containing a rooted phylogenetic tree.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Farris, J. (1972) Estimating phylogenetic trees from distance matrices. *American Naturalist*, **106**, 645-667.

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`midpoint`, `reroot`, `root`

`minRotate` *Rotates all nodes of the tree to minimize the difference in order with a vector*

Description

This function rotates all the nodes of the tree to try and minimize the different between the order of the tips and the rank-order of a numeric vector `x` or (in the case of `tipRotate` the actual integer vector, `x`.

Usage

```
minRotate(tree, x, ...)
tipRotate(tree, x, ...)
```

Arguments

`tree`	tree.
`x`	numeric vector.
`...`	optional arguments to be used by `tipRotate`. Presently optional arguments can be `fn`, function to be used to compute the distance between the order of the tip labels in `tree` and the numeric vector `x` (presently `fn=function(x) x^2` by default); `methods`, the method or methods of tree traversal (can be `"pre"`, `"post"`, or `c("pre","post")`, for pre-, post-, or both pre- and post-order tree traversal); `rotate.multi`, whether to rotate multifurcations in all possible ways using `rotate.multi` (defaults to `FALSE`); and `print`, a logical argument specifying whether to print the search progress or to behave quietly. Only the option `print` is available for `minRotate`.

Details

Both functions are designed primarily to be used internally by other *phytools* functions and particularly by `phylo.to.map` (in the case of `minRotate`) and by `cophylo` (in the case of `tipRotate`).

Value

A node-rotated object of class `"phylo"`.

Author(s)

Liam 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.

`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.

Usage

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

Arguments

`tree`	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. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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`

`modified.Grafen` *Computes modified Grafen edge lengths*

Description

This function computes modified Grafen edge lengths in which the length of the edge is determined not by the number of descendant leaves, but instead by the maximum number of node lengths in the path from the node to any leaf.

`node.paths` is used internally by `modified.Grafen` and computes the set of paths from a node to all tips descended from that node.

Usage

```
modified.Grafen(tree, power=2)
node.paths(tree, node)
```

Arguments

`tree`	object of class `"phylo"`.
`power`	power to raise the depths of each node (in nodes).
`node`	node number for `node.paths`.

Value

An object of class `"phylo"` with edge lengths.

Author(s)

Liam 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.

See Also

`compute.brlen`

`mrp.supertree` *Matrix representation parsimony supertree estimation*

Description

This function estimates the MRP (matrix representation parsimony) supertree from a set of input trees (Baum 1992; Ragan 1992).

Usage

```
mrp.supertree(trees, method=c("pratchet","optim.parsimony"), ...)
compute.mr(trees, type=c("phyDat","matrix"))
```

Arguments

`trees`	an object of class `"multiPhylo"` that consists of a list of phylogenetic trees.
`method`	an argument specifying whether to optimize the tree using the *phangorn* parsimony optimizer `pratchet` or `optim.parsimony`.
`type`	for `compute.mr`, the type of object to return (e.g., `"phyDat"` or `"matrix"`).
`...`	optional arguments - mostly to be passed to `pratchet` or `optim.parsimony`.

Details

`mrp.supertree` uses `pratchet` or `optim.parsimony` from the *phangorn* package (Schliep, 2011) for optimization, and `prop.part` from *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.

The function `compute.mr` computes the matrix-representation matrix of the input trees. It is used internally by `mrp.supertree`, but can also be used to export an object that can be written to file if desired.

Value

An object of class `"phylo"` or `"multiPhylo"` that is the MP or set of MP MRP trees.

In the case of `compute.mr`, an object of class `"phyDat"` or a matrix.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Baum, B. R., (1992) Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. *Taxon*, **41**, 3-10.

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.

Ragan, M. A. (1992) Phylogenetic inference based on matrix representation of trees. *Molecular Phylogenetics and Evolution*, *1*, 53-58.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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"`, or a list of such objects.
`nperm`	number of Mantel permutations to be used to compute a P-value of the test.

Details

Printing the object to screen will result in a summary of the analysis similar to `summary.lm`, but with p-values derived from Mantel permutations.

Methods `residuals` and `fitted` can be used to return residual and fitted matrices, respectively.

Value

An object of class `"multi.mantel"` consisting of the following elements:

`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.
`nperm`	tne number of permutations used.

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`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 (e.g., see `vcv.phylo`) as a list: one for each mapped state.

Usage

```
multiC(tree, internal=FALSE)
```

Arguments

`tree`	an object of class `"simmap"` consisting of a phylogeny with a mapped discrete character.
`internal`	logical value indicating whether or not internal nodes should be returned.

Value

A list of matrices.

Author(s)

Liam 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.

See Also

`evolvcv.lite`, `read.simmap`, `vcvPhylo`, `vcv.phylo`

`multirateBM` *Function to fit a multi-rate Brownian evolution model*

Description

This function fits a flexible Brownian multi-rate model using penalized likelihood.

The model that is being fit is one in which the rate of Brownian motion evolution itself evolves from edge to edge in the tree under a process of geometric Brownian evolution (i.e., Brownian motion evolution on a log scale).

The penalty term, `lambda`, determines the cost of variation in the rate of evolution from branch to branch. If lambda is *high*, then the rate of evolution will vary relatively little between edges (and in the limiting case converge to the single-rate MLE estimate of the rate). By contrast, if the value of `lambda` is set to be low, then the rate of evolution can vary from edge to edge with relatively little penalty.

Decreasing the penalty term, however, is not without cost. As `lambda` is decreased towards zero, estimated rates will tend to become less and less accurate.

Usage

```
multirateBM(tree, x, method=c("ML","REML"),
optim=c("L-BFGS-B","Nelder-Mead","BFGS","CG"),
maxit=NULL, n.iter=1, lambda=1, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`x`	a named numerical vector. Names should correspond to the species names of `tree`.
`method`	method of optimization. Currently only `method="ML"`.
`optim`	optimization routine to be used by `optim`. If more than one is specified and `n.iter>1` then they will be alternated. (This is recommended to improve optimization.)

`maxit`	to be passed to `optim`. If set to `maxit=NULL`, the default value of `maxit` will be used, depending on the optimization method.
`n.iter`	number of times to reiterate failed optimization.
`lambda`	lambda penalty term. High values of `lambda` correspond to high penalty for rate heterogeneity among edges. Low values of `lambda` correspond to low penalty.
`...`	optional arguments.

Value

An object of class `"multirateBM"`.

Author(s)

Liam 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.

Revell, L. J. (2021) A variable-rate quantitative trait evolution model using penalized-likelihood. *PeerJ*, **9**, e11997.

See Also

`brownie.lite`, `evol.rate.mcmc`

Examples

```
## Not run:
## load data
data(sunfish.tree)
data(sunfish.data)
## extract character of interest
gw<-setNames(sunfish.data$gape.width,
    rownames(sunfish.data))
## run penalized-likelihood optimization
## lambda=0.1 is arbitrary
fitBM<-multirateBM(sunfish.tree,gw,
lambda=0.01)
## print and plot the results
print(fitBM)
plot(fitBM,ftype="i",fsize=0.8,lwd=6,
outline=TRUE)
## End(Not run)
```

`multiRF` *Computes Robinson-Foulds distance between a set of trees*

Description

Computes the Robinson-Foulds (Robinson & Foulds 1981) distance between a set of trees in an object of class `"multiPhylo"`.

Usage

```
multiRF(trees,quiet=FALSE,multi2di=FALSE)
```

Arguments

`trees`	object of class `"multiPhylo"` consisting of two or more fully bifurcating, unrooted trees. If trees are rooted, they will be unrooted.
`quiet`	logical argument indicating whether or not to run quietly. (Defaults to `FALSE`.)
`multi2di`	logical argumenet indicating whether or not to resolve multifurcating trees. (Defaults to `FALSE`.)

Details

Computes the Robinson-Foulds distance between all phylogenies in an object of class `"multiPhylo"`. Uses `prop.part` internally for most of the heavy lifting.

Value

A matrix containing distances.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Robinson, D. R., Foulds, L. R. (1981) Comparison of phylogenetic trees. *Mathematical Biosciences*, **53**, 131-147.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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

Description

`nodeHeights` computes the height above the root for all nodes in the tree. `nodeheight` computes the height above the root for a single node.

Usage

```
nodeHeights(tree, ...)
nodeheight(tree, node, ...)
```

Arguments

`tree`	a phylogeny as an object of class `"phylo"`.
`node`	for `nodeheight`, the node for which we want to compute a height above the root (or including the root edge, for `root.edge=TRUE`).
`...`	optional arguments - presently only `root.edge`, a logical value indicating whether or not to include the root edge length in the calculation of node heights.

Details

The function `nodeHeights` 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))`. Generally speaking, `nodeHeights` will be faster if the heights of all or a large proportion of nodes is needed, wherease `nodeheight` will be faster if the height of one or a small number of nodes are needed.

Value

Either a matrix of the same dimensions as `tree$edge` containing the height above the root of each node in `edge` (for `nodeHeights`); or a single positive number (for `nodeheight`).

Author(s)

Liam 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.

See Also

`vcvPhylo`

Examples

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

---

nodelabels.cophylo *Add labels to a plotted "cophylo" object*

Description

This function adds node, edge, or tip labels to the plotted trees of a `"cophylo"` object.

Usage

```
nodelabels.cophylo(..., which=c("left","right"))
edgelabels.cophylo(..., which=c("left","right"))
tiplabels.cophylo(..., which=c("left","right"))
```

Arguments

`...`	arguments to be passed to `nodelabels`, `edgelabels`, or `tiplabels`.
`which`	argument indicated which of the two plotted trees (the `"left"` or `"right"` tree) to be used.

Details

Note that the order of tips, edges, and nodes may be different in the object of class `"cophylo"` than they are in the original input trees, particularly if `cophylo(...,rotate=TRUE)` was used.

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`cophylo`, `edgelabels`, `nodelabels`, `tiplabels`

`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, convergence 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 object 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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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. This is handy if we want to, for instance, run `brownie.lite` on a set of mapped trees, and then average the fitted parameter values across trees. The function 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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`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. `paintSubTree` paints the clade downstream of `node` with a particular state; whereas `paintBranches` paints only a specified branch.

Usage

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

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.
`edge`	an integer or vector of integers specifying the node or tip numbers of the edges that should be painted in `paintBranches`.
`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

An object of class `"simmap"` that contains the specified paintings as a mapped discrete character.

Author(s)

Liam 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.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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.

Lastly, under the taxon number stopping criterion (`n`) for a non-zero extinction rate (`d>0`) sometimes a tree containing fewer than `n` extant tips is returned because it has gone completely extinct before the end of the simulation.

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 for simultaneously conditioning on `n` and `t` - options are `"rejection"` and `"direct"`; `tip.label`, a vector of tip labels (only works for `n!=NULL`); 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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Examples

```
# simulate a pure-birth tree with 400 tips, scaled to a length of 1.0
tree<-pbtree(n=400,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).

`pgls.SEy` fits a simpler model in which only sampling error in `y` is taken into account. This function uses `gls` from the nlme package internally for optimization and returns an object of class `"gls"` that is compatible with all methods for that object class.

Usage

```
pgls.Ives(tree, X, y, Vx=NULL, Vy=NULL, Cxy=NULL, lower=c(1e-8,1e-8),
fixed.b1=NULL)
pgls.SEy(model, data, corClass=corBrownian, tree,
   se=NULL, method=c("REML","ML"), interval=c(0,1000), ...)
```

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 σ_x^2 and σ_y^2, respectively. (Must be > 0.)
`fixed.b1`	fixed regression slope, β. Usually set to zero for null hypothesis testing.
`model`	model to fit. (For `pgls.SEy`.)
`data`	data frame. (For `pgls.SEy`.)
`corClass`	correlation structure. (For `pgls.SEy`.)
`se`	vector of standard errors in y. (For `pgls.SEy`.)
`method`	optimization method. (For `pgls.SEy`.)
`interval`	interval over which to perform optimization. (For `pgls.SEy`.)
`...`	optional arguments. (For `pgls.SEy`.)

Details

In the case of `pgls.Ives`, only the bivariate regression model is implemented. Note that some problems have been reported with the optimization algorithm for this model, which is simple and thus may fail to find the ML solution.

In the case of `pgls.SEy` the user can (theoretically) specify any class of linear model permitted by `gls`.

Value

In the case of `pgls.Ives`, an object of class `"pgls.Ives"` 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.

In the case of `pgls.SEy`, an object of class `"gls"`.

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`brownie.lite`, `phylosig`, `phyl.resid`

`phenogram` *Plot phenogram (traitgram)*

Description

Function plots a traitgram (Evans et al. 2009), 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`, and `digits` are as in `plot.default` or `par`. Note that `axes` overrides `xlim` and `ylim`. `spread.labels` is a logical value indicating whether or not to minimize tip label overlap (default is `TRUE`); `spread.cost` is a numeric vector indicating the relative penalty to be used for label overlap and deviance, respectively (if `spread.labels=TRUE`); `spread.range` is the range over which to (potentially) spread the labels - note that if labels do not overlap, not all of that range will be used; 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 10-percent of the total tree length 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. `hold` indicates whether (or not) the output to the graphical device should be held using `dev.hold` before plotting (defaults to `hold=TRUE`). `quiet` suppresses some system messages if set to `quiet=TRUE`.

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. Note that 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, and (invisibly) returns a matrix containing the coordinates of the plotted tip labels.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Evans, M. E. K., Smith, S. A., Flynn, R. S., Donoghue, M. J. (2009) Climate, niche evolution, and diversification of the "bird-cage" evening primroses (Oenothera, sections Anogra and Kleinia). *American Naturalist*, **173**, 225-240.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Examples

```
## load data from Garland et al. (1992)
data(mammal.tree)
data(mammal.data)
## extract character of interest
ln.bodyMass<-log(setNames(mammal.data$bodyMass,
    rownames(mammal.data)))
## plot traitgram
phenogram(mammal.tree,ln.bodyMass,ftype="i",
    spread.cost=c(1,0),fsize=0.7,xlab="time (ma)",
    ylab="log(body mass)")
```

`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 λ.
`fixed`	optionally, a logical value indicating whether or not to estimate λ using likelihood.

Details

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

Value

An object of class `"phyl.cca"` containing 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., Harrison, A. S. (2008) PCCA: A program for phylogenetic canonical correlation analysis. *Bioinformatics*, **24**, 1018-1020.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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.*). This is not a phylogenetic ANOVA, in which we want to compare the means of different sets of species on the tree. Instead, we are interested in the difference between two characters, or two measures of a character within a species, and we want to know if this difference is significantly different from zero controlling for the phylogenetic non-independence of species.

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 λ (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 λ.
`...`	optional arguments.

Details

Likelihood optimization is performed using `optim` with `method="L-BFGS-B"` with box constraints on λ (0,1).

Value

An object of class `"phyl.pairedttest"` with the following components:

`dbar`	phylogenetic mean difference.
`se`	standard error of `dbar`.
`sig2`	estimated evolutionary variance (of the difference).
`lambda`	fitted (or fixed) value of λ.
`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.
`df`	the degrees of freedom.
`h0`	the null hypothesis that was tested.

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`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", ...)
## S3 method for class 'phyl.pca'
biplot(x, ...)
scores(object, ...)
## S3 method for class 'phyl.pca'
scores(object, ...)
as.princomp(x, ...)
## S3 method for class 'phyl.pca'
as.princomp(x, ...)
```

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"`.
`x`	object of class `"phyl.pca"` for `biplot` and `as.princomp` methods.
`object`	object of class `"phyl.pca"` for `scores` method.
`...`	for S3 plotting method `biplot.phyl.pca`, other arguments to be passed to `biplot`.

Details

If `method="lambda"` then λ will be optimized on the interval (0,1) using `optimize`. Optimization method can be set using the option `opt` which can take values `"ML"`, `"REML"`, or `"fixed"`. If the last of these is selected than the user should also specify a value of λ to use via the argument `lambda`.

S3 methods (`print`, `summary`, and `biplot`) are modified from code provided by Joan Maspons and are based on the same methods for objects of class `"prcomp"`. Function `biplot` now permits the argument `choices` to be supplied, which should be a vector of length two indicated the two PC axes to be plotted.

S3 method `scores` extracts or computes (for a matrix of `newdata`) PC scores given an object of class `"phyl.pca"`.

Value

An object of class `"phyl.pca"` consisting of a list with some or all of 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 λ (`method="lambda"` only).
`logL`	log-likelihood for λ model (`method="logL"` only).

Author(s)

Liam Revell <`liam.revell@umb.edu`>, Joan Maspons

References

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`phyl.cca`, `phyl.resid`

Examples

```
## load data from Mahler et al. (2010)
data(anoletree)
data(anole.data)
## run phylogenetic PCA
anole.pca<-phyl.pca(anoletree,anole.data)
print(anole.pca)
## plot results
plot(anole.pca)
biplot(anole.pca)
```

`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 theoretically redundant with `residuals` applied to a `"gls"` object class in which the correlation structure is based on `corBrownian` or `corPagel`; however 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 λ 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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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)
## S3 method for class 'phyl.RMA'
coef(object, ...)
## S3 method for class 'phyl.RMA'
plot(x, ...)
```

Arguments

`x`	vector with names. In the case of the S3 `plot` method `x` is an object of class `"phyl.RMA"`.
`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 β. Defaults to 1.0. Note that a null hypothesis of 0.0 is not allowed.
`object`	for `coef` method, an object of class `"phyl.RMA"`.
`...`	optional arguments for S3 methods.

Details

Optionally jointly estimates λ if `method="lambda"`. Likelihood optimization of λ is performed using `optimize` on the interval (0,1). Note that some statistician think there is *never* a condition in which a reduced-major-axis regression should be used.

The statistical hypothesis testing is based on Clarke (1980; reviewed in McArdle 1988), which differs from some other implementations of non-phylogenetic major axis regression in R.

Value

An object of class `"phyl.RMA"` consisting of 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 λ (`method="lambda"` only).

`logL`	log-likelihood (`method="lambda"` only).
`test`	a vector containing results for hypothesis tests on β.
`resid`	a vector of residuals for `y` given `x`.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Clarke, M. R. B. (1980) The reduced major axis of a bivariate sample. *Biometrika*, **67**, 441-446.

McArdle, B. H. (1988) The structural relationship: Regression in biology. *Can. J. Zool.*, **66**, 2329-2339.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`phyl.cca`, `phyl.pca`, `phyl.resid`

Examples

```
## load data from Garland et al. (1992)
data(mammal.data)
data(mammal.tree)
## pull out & log transform variables
lnBodyMass<-setNames(log(mammal.data$bodyMass),
rownames(mammal.data))
lnHomeRange<-setNames(log(mammal.data$homeRange),
rownames(mammal.data))
## fit RMA regression & print results
fitted.rma<-phyl.RMA(lnBodyMass,lnHomeRange,
mammal.tree)
print(fitted.rma)
```

`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, λ, and a data matrix.

This function should not be confused with `vcv.phylo` in the *ape* package (although one of the objects returned is the output of `vcv.phylo`).

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`.

Value

A list containing three elements, as follows: `C`, the matrix `vcv.phylo` transformed by `lambda`; `R`, the among trait variance-covariance matrix for the data in `X`; and `alpha`, a vector of ancestral states at the root node of the tree.

Author(s)

Liam 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.

---

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

This function uses a little bit of code from both `phy.anova` in the *geiger* package and `pairwise.t.test`.

Value

An object of class `"phylANOVA"` 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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`anova`, `pairwise.t.test`

`phylo.heatmap` *Creates a phylogenetic heat map*

Description

Functions creates a multivariate phylogenetic `heatmap`.

Usage

```
phylo.heatmap(tree, X, fsize=1, colors=NULL, standardize=FALSE, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`X`	a matrix containing data for multiple continuous characters in which `rownames` correspond to the tip labels of the tree.
`fsize`	an integer or vector of length 3 containing the font size for the tip labels, the trait labels, and the legend text. (If a single integer is supplied, then the value will be recycled.)
`colors`	a vector of colors to be passed to `image`. Can be a function call (e.g., `heat.colors(n=200)[200:1]`).
`standardize`	a logical value indicating whether or not to standardize each column of `X` to have the same variance & mean prior to analysis.
`...`	optional arguments. So far these include: `legend`, a logical value indicating whether or not to plot a figure legend (defaults to `legend=TRUE`); `labels`, a logical value indicating whether or not to plot trait labels (defaults to `labels=TRUE`); `split`, a numeric vector indicating the fraction of the horizontal dimension to use for the tree & heatmap, respectively (defaults to `split=c(0.5,0.5)`); `xlim`, `ylim`, & `mar`, defined as in `par`; and `ftype`, `lwd`, and `pts` as defined in `plotSimmap`.

Value

Function creates a plot.

Author(s)

Liam 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.

Examples

```
## load data
data(anoletree)
data(anole.data)
## compute phylogenetic residuals
anole.data<-as.matrix(anole.data)
anole.resids<-cbind(anole.data[,1],
    phyl.resid(anoletree,anole.data[,1,drop=FALSE],
    anole.data[,2:ncol(anole.data)])$resid)
colnames(anole.resids)[1]<-"SVL"
## plot phylogenetic heatmap
phylo.heatmap(anoletree,anole.resids,
    split=c(0.7,0.3),fsize=c(0.4,0.8,0.8),
    standardize=TRUE)
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margins to default
```

---

phylo.impute *Phylogenetic imputation for multivariate continuous character data*

Description

This function performs phylogenetic imputation using Maximum Likelihood.

Usage

```
phylo.impute(tree, X, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`X`	data matrix with species names as row labels. Missing data to be imputed should be coded `NA`.
`...`	optional arguments.

Details

This function performs phylogenetic imputation in which the evolution of the characters in `X` is assumed to have occured by correlation multivariate Brownian motion. Missing values are imputed by maximizing their likelihood jointly with the parameters of the Brownian model. The function `evol.vcv` is used internally to compute the likelihood. Note that the *Rphylopars* package (`https://CRAN.R-project.org/package=Rphylopars`) also does phylogenetic imputation for multivariate trait data and it seems to be much faster.

Value

An object of class `"phylo.impute"` consisting of a complete data frame with missing values imputed.

Since optimization is performed numerically using likelihood, a summary of the optimization can be seen by evaluating `attr(object,"optim")`, in which `object` is of class `"phylo.impute"`.

Author(s)

Liam 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.

See Also

`evol.vcv`

`phylo.to.map` *Plot tree with tips linked to geographic coordinates*

Description

`phylo.to.map` creates an object of class `"phylo.to.map"` and (optionally) plots that object.

`plot.phylo.to.map` plots an object of class `"phylo.to.map"` in which the tips of the tree point to coordinates on a geographic map.

Usage

```
phylo.to.map(tree, coords, rotate=TRUE, ...)
## S3 method for class 'phylo.to.map'
plot(x, type=c("phylogram","direct"), ...)
```

Arguments

`tree`	an object of class "phylo".
`coords`	a matrix containing the latitude (in column 1) and the longitude of all tip species in the tree. The row names should be the same as `tree$tip.label`; however, more than one set of coordinates per species can be supplied by duplicating some row names.
`rotate`	a logical value indicating whether or not to rotate nodes of the tree to better match longitudinal positions.
`x`	for `plot.phylo.to.map`, an object of class `"phylo.to.map"`.
`type`	a string indicating whether to map the tips of the tree onto a geographic map from a square phylogram (`type="phylogram"`) or to project the tree directly onto the map (`type="direct"`).
`...`	various optional arguments. For the function `phylo.to.map`, which first creates an object of the special class `"phylo.to.map"` and then (optionally) plots this object, arguments include: `database` and `regions` (see `map`), as well as any arguments that should be passed to `plot.phylo.to.map` internally. For `phylo.to.map`, optional arguments `xlim` and `ylim`, which control the plot area for the map; `fsize` for the font size of plot labels and `ftype` for the font type (following `plotSimmap`; `split` which controls the proportion of vertical (or horizontal) space for the tree (first) and map, in a vector; `psize` the size of the plotted points on the map - or `cex.points`, a vector contain the size of the tip points and geographic coordinate points, respectively; `from.tip` a logical value indicating whether to plot the linking lines from the tips (if `TRUE`) or from the end of the tip label, the default; `colors`, a single value or a vector of colors for the points and the linking lines; `pch` a single value or a vector of point types; `lwd` and `lty` for the linking lines; and `pts` a logical value indicating whether or not to plot points at the tips of the tree. `mar` and `asp` are as in `par`.

Value

`phylo.to.map` creates an object of class `"phylo.to.map"` and (if `plot=TRUE`) plots a phylogeny projected onto a geographic map.

`plot.phylo.to.map` plots on object of class `"phylo.to.map"`.

Author(s)

Liam 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.

`phylo.toBackbone` *Converts tree to backbone or vice versa*

Description

Converts between `"phylo"` and `"backbonePhylo"`.

Usage

```
phylo.toBackbone(x, trans, ...)
backbone.toPhylo(x)
```

Arguments

`x`	an object of class `"phylo"` (for the function `phylo.toBackbone`), or an object of the special class `"backbonePhylo"` (for `backbone.toPhylo`).
`trans`	data frame containing the attributes necessary to translate a backbone tree to an object of class `"backbonePhylo"`. The data frame should contain the following variables: `tip.label`: the tip labels in the input tree (not all need be included); `clade.label`: labels for the unobserved subtrees; `N`: number of species in each subtree; and `depth`: desired depth of each subtree. `depth` for each terminal taxon in `x` cannot be greater than the terminal edge length for that taxon.
`...`	optional arguments.

Value

Either an object of class `"phylo"` or an object of class `"backbonePhylo"`, depending on the method.

Author(s)

Liam 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.

See Also

`plot.backbonePhylo`

`phyloDesign` *Compute design matrix for least squares analyses*

Description

Primarily an internal function for `optim.phylo.ls`, this 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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`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.

`project.phylomorphospace` animates the phylomorphospace projection.

Usage

```
phylomorphospace(tree, X, A=NULL, label=c("radial","horizontal","off"),
   control=list(), ...)
project.phylomorphospace(tree, X, nsteps=200, sleep=0,
direction=c("to","from","both"), ...)
```

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`	string indicating whether to plot the tip labels in the same direction as the terminal edge (`label="radial"`), horizontally `label="horizontal"`, or not at all `"off"`. `label=TRUE` and `label=FALSE` are also acceptable, for compatibility with phytools <= 0.3-03.
`control`	a list containing the following optional control parameters: `col.edge`: a vector of edge colors; and `col.node`: a vector of node colors.
`nsteps`	for `project.phylomorphospace` the number of frames in the animation between the phylogeny & the phylomorphospace or *vice versa*.
`sleep`	for `project.phylomorphospace` the time between frames.
`direction`	for `project.phylomorphospace` whether to morph `"to"` a phylomorphospace, `"from"` a phylomorphospace, or there & back again (`"both"`).
`...`	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. `node.size` is a vector containing the point size relative to the default (see `par` for plotted internal nodes and tips, respectively. Defaults to `node.size=c(1,1.3)`. If only one number is

provided it will be recycled. `axes` is a logical value indicating whether or not axes should be plotted (see `plot.default`. Finally, `add` indicates whether to add the phylmorphospace to the current plot.

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**, 289-90.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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

Examples

```
## load tree & data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)
## set colors for mapped discrete character
cols<-setNames(c("blue","red"),
    levels(sunfish.data$feeding.mode))
phylomorphospace(sunfish.tree,sunfish.data[,3:2],
    colors=cols,bty="l",ftype="off",node.by.map=TRUE,
    node.size=c(0,1.2),xlab="relative buccal length",
    ylab="relative gape width")
title(main="Phylomorphospace of buccal morphology in Centrarchidae",
    font.main=3)
```

---

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(),
   method=c("dynamic","static"), ...)
```

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`).
`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); `col.edge` a vector of colors of length `nrow(tree$edge)`.
`method`	a string either `"dynamic"` for a dynamic (animated) plot created using `rgl`; or `"static"` for a flat 3D plot created using `scatterplot3d` and base graphics. The latter has the advantage of being very easy to export in standard format.
`...`	optional arguments to be passed to `scatterplot3d`. Most options not available. `angle` is an important option that does work here.

Value

This function creates a three dimensional phylomorphospace plot. The function returns a function from `spin3d` (for `method="dynamic"`); or a series of functions from `scatterplot3d` (for `method="static"`).

Author(s)

Liam 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.

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

```
data(anoletree)
data(anole.data)
anole.pca<-phyl.pca(anoletree,anole.data)
## Not run:
phylomorphospace3d(anoletree,scores(anole.pca)[,1:3],
```

```
    control=list(spin=FALSE))
## End(Not run)
par(cex=0.5)
phylomorphospace3d(anoletree,scores(anole.pca)[,1:3],
    method="static",angle=-30)
par(cex=1)
```

---

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())
## S3 method for class 'phylosig'
plot(x, ...)
```

Arguments

`tree`	a phylogenetic tree in `"phylo"` format.
`x`	vector containing values for a single continuously distributed trait. In the case of the `plot` method, `x` is an object of class `"phylosig"`.
`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) σ^2 and λ. 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`.
`...`	optional arguments for `plot` method.

Details

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

`plot.phylosig` creates either a plot of the null distribution of K or a likelihood surface, depending on the value of `method`.

Value

The function returns an object of class `"phylosig"`. With default arguments (`method="K"`, `test=FALSE`, and `se=NULL`), this will be a single numeric value. Otherwise, if (`method="K"`), it will consist of a list with up to 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"`), it will be a list with up to the following components:

`lambda`	fitted value of λ.
`sig2`	rate of evolution, σ^2, for estimation with sampling error.
`logL`	log-likelihood.
`logL0`	log-likelihood for $\lambda = 0$.
`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

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.

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Examples

```
## load data from Garland et al. (1992)
data(mammal.tree)
data(mammal.data)
## extract characters of interest
ln.bodyMass<-log(setNames(mammal.data$bodyMass,
    rownames(mammal.data)))
ln.homeRange<-log(setNames(mammal.data$homeRange,
    rownames(mammal.data)))
## compute phylogenetic signal K
K.bodyMass<-phylosig(mammal.tree,ln.bodyMass,
    test=TRUE)
```

```
print(K.bodyMass)
plot(K.bodyMass)
K.homeRange<-phylosig(mammal.tree,ln.homeRange,
    test=TRUE)
print(K.homeRange)
plot(K.homeRange)
## compute phylogenetic signal lambda
lambda.bodyMass<-phylosig(mammal.tree,ln.bodyMass,
    method="lambda",test=TRUE)
print(lambda.bodyMass)
plot(lambda.bodyMass)
lambda.homeRange<-phylosig(mammal.tree,ln.homeRange,
    method="lambda",test=TRUE)
print(lambda.homeRange)
plot(lambda.homeRange)
```

`plot.backbonePhylo` *Plots backbone tree with triangles as clades*

Description

Function plots a backbone tree (stored as an object of class `"backbonePhylo"`) with triangles as subtrees.

Usage

```
## S3 method for class 'backbonePhylo'
plot(x, ...)
```

Arguments

`x`	an object of class `"backbonePhylo"`.
`...`	optional arguments. Includes `vscale` (to rescale the vertical dimension in plotting), `fixed.height` (logical value to fix the plotted height of subtree triangles), `print.clade.size` (logical), `fixed.n1` (logical value indicating whether or not to use the fixed triangle height for subtrees containing only one taxon, or to plot as a leaf - defaults to `FALSE`), and `col` (a single value, or a vector with names, giving the clade colors), as well as `xlim`, `ylim`, and `lwd` (as well as perhaps other standard plotting arguments).

Value

Plots a tree.

Author(s)

Liam 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.

See Also

`phylo.toBackbone`

Examples

```
## first create our backbone tree with
## random subtree diversities
tree<-phytools:::lambdaTree(pbtree(n=10),lambda=0.5)
## create a translation table
## leaving a couple of single-taxon clades for fun
tip.label<-sample(tree$tip.label,8)
clade.label<-LETTERS[1:8]
N<-ceiling(runif(n=8,min=1,max=20))
## set crown node depth to 1/2 the maximum depth
depth<-sapply(tip.label,function(x,y)
0.5*y$edge.length[which(tree$edge[,2]==
which(y$tip.label==x))],y=tree)
trans<-data.frame(tip.label,clade.label,N,depth)
rownames(trans)<-NULL
rm(tip.label,clade.label,N,depth)
## here's what trans looks like
trans
## convert
obj<-phylo.toBackbone(tree,trans)
## plot
plot(obj)
```

`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"`. `palette` can also be a function produced by `colorRampPalette`.
`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` - pretty much all arguments are available. In addition, there `plotBranchbyTrait` has the following additional optional arguments: `tol` a small tolerance value to be added to the range of `x`; `prompt` for `legend=TRUE`, a logical value indicating whether to prompt for the position of the legend (or not) - the default is to put the legend in the lower left hand size of the plot; `title` for `legend=TRUE`, the title of the legend; and `digits` for `legend=TRUE`, the number of digits in the quantitative scale of the legend. Finally, `cex` can be supplied as either a single numeric value, or as a vector of two different values. If the latter is true than the second element of `cex` will be passed internally to the function `add.color.bar`.

Details

Note that if `prompt=TRUE`, the function will prompt for the position of the legend.

Value

Plots a phylogeny.

Author(s)

Liam 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.

---

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=FALSE,
   node.numbers=FALSE, mar=NULL, add=FALSE, offset=NULL,
   direction="rightwards", type="phylogram", setEnv=TRUE, part=1.0,
   xlim=NULL, ylim=NULL, nodes="intermediate", tips=NULL, maxY=NULL,
   hold=TRUE, split.vertical=FALSE,
   lend=2, asp=NA, outline=FALSE, plot=TRUE)
## S3 method for class 'simmap'
plot(x, ...)
## S3 method for class 'multiSimmap'
plot(x, ...)
```

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 `FALSE`.
`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 in character widths.
`direction`	plotting direction. Options are `"rightwards"` (the default), `"leftwards"`, `"upwards"` or `"downwards"`. For `method="fan"` `direction` is ignored.
`type`	plot type. Can be `"phylogram"`, `"fan"`, or `"cladogram"`. Only a subset of options are presently available for `type="fan"`.
`setEnv`	logical value indicating whether or not to set the environment `.PlotPhyloEnv`. Setting this to `TRUE` (the default) will allow compatibility with ape labeling functions such as `nodelabels`.

`part`	value between 0 and 1 for `type="fan"` indicating what fraction of the full circular tree to use as plotting area. For instance, `part=0.5` will plot a half fan phylogeny. It also affects the axis scaling used.
`xlim`	x-limits for the plot.
`ylim`	y-limits for the plot.
`nodes`	node placement following Felsenstein (2004; pp. 574-576). Can be `"intermediate"`, `"centered"`, `"weighted"`, or `"inner"`. So far only works for `type="phylogram"`.
`tips`	labeled vector containing the vertical position of tips. Normally this will be `1:N` for `N` tips in the tree.
`maxY`	maximum value of `y` to use before rotating a tree into fan configuration. This will only make a difference if different from `Ntip(tree)`.
`hold`	logical argument indicating whether or not to hold the output to the graphical device before plotting. Defaults to `hold=TRUE`.
`split.vertical`	split the color of the vertically plotted edges by the state of the daughter edges. Only applies if the edge state changes exactly at a node.
`lend`	line end style. See `par`.
`asp`	aspect ratio. See `plot.window`.
`outline`	logical value indicating whether or not to draw a black outline around the plotted edges of the tree.
`plot`	logical value indicating whether or not to actually plot the tree. (See equivalent argument in `plot.phylo`.)
`x`	for S3 plotting method, object of class `"simmap"` or `"multiSimmap"`.
`...`	for S3 plotting method, other arguments to be passed to `plotSimmap`.

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

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

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

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

densityMap, make.simmap, read.simmap

Examples

```
data(anoletree)
cols<-setNames(c("green","#E4D96F","darkgreen",
    "brown","black","darkgrey"),
    c("CG","GB","TC","TG","Tr","Tw"))
plot(anoletree,cols,fsize=0.5,ftype="i",outline=TRUE,
lwd=3,ylim=c(0,Ntip(anoletree)),
    mar=c(0.1,0.1,1.1,0.1))
add.simmap.legend(colors=cols,prompt=FALSE,x=0,y=-0.5,
    vertical=FALSE)
title(main="Caribbean ecomorphs of anoles",font.main=3,
    line=0)
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margins to default
```

---

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", legend=TRUE,
   ...)
```

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.
`legend`	logical value or text to be plotted in the legend.
`...`	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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J. (2014) Ancestral character estimation under the threshold model from quantitative genetics. *Evolution*, bold68, 743-759.

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.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`plot.phylo`, `plotSimmap`

Examples

```
data(vertebrate.tree)
plotTree(vertebrate.tree,fsize=1.2,ftype="i")
## reset margins
par(mar=c(5.1,4.1,4.1,2.1))
```

`plotTree.datamatrix` *Plot a tree with a discrete character data matrix at the tips*

Description

Function plots a phylogeny next to a matrix of discrete character data.

Usage

```
plotTree.datamatrix(tree, X, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`X`	a data frame with columns as factors.
`...`	optional arguments.

Value

Invisibly returns a list containing the font size, a list of the colors used for each column of the plotted data matrix, and the x-coordinate of the rightmost edge of the matrix.

Author(s)

Liam 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.

See Also

`dotTree`, `phylo.heatmap`

`plotTree.errorbars` *Plot a tree with error bars around divergence dates*

Description

Plots a tree with error bars around divergence times (nodes).

Usage

```
plotTree.errorbars(tree, CI, ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`CI`	confidence intervals around internal nodes of the tree, measured in time since the present.
`...`	optional arguments to be passed to `plotTree`.

Details

The matrix CI show contain (in rows) the lower & upper confidence bounds in time since the present.

Optional arguments specific to the error bar plot include `gridlines`, `bar.lwd`,`cex` (for the points plotted at nodes), and `bar.col`.

Value

Plots a tree with error bars around internal nodes..

Author(s)

Liam 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.

See Also

`ltt`, `plotTree`

`plotTree.wBars` *Plot a tree with bars at the tips*

Description

`plotTree.wbars` plots a phylogeny in phylogram or fan style with bars at the tips representing the values for a phenotypic trait.

`plotTree.barplot` creates a split plot in which a right-facing phylogram is on the left, and a bar plot is shown on the right.

`plotTree.boxplot` creates a split plot in which a right-facing phylogram is on the left, and a box plot is shown on the right.

Usage

```
plotTree.wBars(tree, x, scale=NULL, width=NULL, type="phylogram",
   method="plotTree", tip.labels=FALSE, col="grey", border=NULL,
   ...)
plotTree.barplot(tree, x, args.plotTree=list(), args.barplot=list(),
...)
plotTree.boxplot(tree, x, args.plotTree=list(), args.boxplot=list(),
...)
```

Arguments

`tree`	an object of class `"phylo"`.
`x`	a named vector or matrix of trait values. For `plotTree.boxplot`, the names should repeat for multiple observations per species. For `plotTree.boxplot` `x` can also be supplied as a formula, though in that case the factor levels need to be provided in a valid cladewise order of the tips in `tree`. This order doesn't need to correspond with the current order of the tip labels. For `plotTree.barplot` `x` can be a matrix (or a data frame) in which columns are the values of multiple traits to be simultaneously plotted on the tree.
`scale`	scaling factor for the tip bars (relative to the total tree height). If left as `NULL` a reasonable scaling factor is computed automatically.
`width`	width of the tip bars.
`type`	plot type. Can be `"phylogram"` or `"fan"`.
`method`	plotting method to use. Can be `"plotTree"` (for `plotTree`) or `"plotSimmap"` (for `plotSimmap`).
`tip.labels`	argument indicating whether or not tip labels should be plotted. Defaults to `tip.labels=FALSE`.
`col`	colors of the plotted bars. Can be a single value or a vector with length equal to the number of tips in the tree.
`border`	single value specifying the color of the border for the plotted bars. Defaults to `border=NULL`, which means that black borders will be plotted.

`args.plotTree`	in `plotTree.barplot`, arguments to be passed to `plotTree`.
`args.barplot`	in `plotTree.barplot`, arguments to be passed to `barplot`.
`args.boxplot`	in `plotTree.boxplot`, arguments to be passed to `boxplot`.
`...`	optional arguments to be passed to `plotTree` or `plotSimmap` in the case of `plotTree.wBars`. For `plotTree.barplot`, the only optional arguments are add and `ylim`. Generally add should not be used; however it can be employed to tell the function to draw the tree & barplot, respectively, in the next two open plotting devices - rather than creating a table of figures in the current plotting device. `ylim` (which is also an optional argument for `plotTree.boxplot` should be supplied here rather than using `args.plotTree` or `args.boxplot`\codeargs.barplot because *y* axis limits must match exactly between the two plots.

Value

Plots a tree with an associated bar plot for a continuously valued character at the tips.

Author(s)

Liam 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.

See Also

`barplot`, `dotTree`, `plotSimmap`, `plotTree`

Examples

```
## load data from Mahler et al. (2010)
data(anoletree)
data(anole.data)
## extract overall body size (SVL)
svl<-setNames(anole.data$SVL,rownames(anole.data))
## plotTree.wBars
plotTree.wBars(anoletree,svl,type="fan",scal=0.5)
par(mar=c(5.1,4.1,4.1,2.1))
## plotTree.barplot
plotTree.barplot(anoletree,exp(svl),
    args.plotTree=list(fsize=0.5),
    args.barplot=list(xlab="SVL (mm)"))

## load vertebrate tree and data
data(vertebrate.tree)
data(vertebrate.data)
## plotTree.barplot
options(scipen=3) ## change sci-notation
plotTree.barplot(vertebrate.tree,
```

```
setNames(vertebrate.data$Mass,
rownames(vertebrate.data)),
args.barplot=list(log="x",
xlab="mass (kg)",
xlim=c(0.01,500000),
col=palette()[4]))
options(scipen=0)

## reset par to defaults
par(mfrow=c(1,1),mar=c(5.1,4.1,4.1,2.1))
```

`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 (σ_1^2) and tipward (σ_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 σ_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`

`posthoc` *Generic post-hoc test*

Description

This function conducts posthoc test.

Usage

```
posthoc(x, ...)
```

Arguments

`x`	an object on which to conduct a post-hoc test.
`...`	optional arguments to be passed to method.

Details

So far is only implemented for object class `"ratebytree"`.

Value

An object.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

See Also

`ratebytree`

`print.backbonePhylo` *Print method for backbone phylogeny*

Description

Print method for an object of class `"backbonePhylo"`.

Usage

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

Arguments

`x`	an object of class `"backbonePhylo"`.
`...`	optional arguments.

Value

Prints to screen.

Author(s)

Liam 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.

See Also

`phylo.toBackbone`

`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 an object of class `"ratebystate"` 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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`fastAnc`, `pic`

`ratebytree` *Likelihood test for rate variation among trees, clades, or traits*

Description

This function essentially implements three different methods for comparing the rate or process of evolution between trees: one for continuously-valued traits, a second for discrete characters, and a third for the rate of diversification (speciation & extinction).

In all cases, the function takes an object of class `"multiPhylo"` containing two or more phylogenies (`trees`), and (for the first two analyses) a list of trait vectors (`x`).

For continuous traits, the function then proceeds to fit two models: one in which the rate (or regime, for models `"OU"` and `"EB"`) of trait evolution is equal among all trees; and a second in which the rates or regimes can differ between trees.

The latter model corresponds to an extension of the *censored* approach of O'Meara et al. (2006; Revell et al. 2018) and should also be related to the method of Adams (2012) for comparing rates among traits. See `brownie.lite` for a different implementation of the *noncensored* approach of O'Meara et al. (2006).

For discrete traits, the function instead proceeds to fit two variants of the M*k* model (Lewis 2001): one in which the parameters values (transition rates) of the process are free to vary between trees, and a second in which they are fixed to be the same.

For diversification alone, the function fits two different diversification (speciation & extinction) models (Nee et al. 1994; Stadler 2012): one in which the birth (speciation) and death (extinction) rates are identical between the trees, and a second in which they are permitted to differ in various ways depending on the value of `"model"` (Revell 2018).

The method `posthoc` conducts a post-hoc comparison of parameter estimates between trees in the multi-rate or multi-process model. The parameter that is compared depends on the fitted model. For instance, in `model="BM"` posthoc comparison is made of `sig2`; if `model="OU"` fitted values of `alpha` are compared; and so on. The argument `p.adjust.method` can be used to specify a method for adjusting P-values for multiple tests following `p.adjust` (defaults to `p.adjust.method="none"`.

Usage

```
ratebytree(trees, x, ...)
## S3 method for class 'ratebytree'
posthoc(x, ...)
```

Arguments

`trees`	an object of class `"multiPhylo"`. If `x` consists of a list of different traits to be compared, then `trees` could also be a simple set of duplicates of the same tree, e.g., `rep(tree,length(x))`.
`x`	a list of trait vectors for a continuous trait in which the names of each vectors correspond to the tip labels of `trees`. This is not used if `type="diversification"`. In the case of `posthoc.ratebytree`, an object of class `"ratebytree"`.

`...`	optional arguments, including the argument `type` (`"continuous"`, `"discrete"`, or `"diversification"`), which, if not specified, the function will attempt to ascertain. See `Details` for more information.

Details

At present it is not possible to specify different models to fit for the different trees - although if (for instance) character evolution on tree 1 proceeded by a strong *OU* process while character evolution on tree 2 was by *BM*, we would probably reject a constant-process model and tree 2 should show a very low value of `alpha`.

To compute the standard errors for each fitted paramater value, the function computes the negative inverse of the Hessian matrix at the MLEs; however, if this matrix is computationally singular the generalized inverse (`ginv`) will be used instead without warning.

The function also conducts a likelihood-ratio test to compare the two models.

For continuous character, optional arguments presently include the following: `model`, the model of continuous trait evolution (options are `"BM"`, the default, `"OU"`, and `"EB"`). `tol`, used as a minimum value for the fitting rates, to prevent problems in optimization. `trace`, a logical value indicating whether or not to report progress in the optimization. `test`, the method for hypothesis testing (options are `"chisq"` and `"simulation"`). `quiet`, a logical value indicating whether or not to run perfectly quietly. Finally, `se`, a list of vectors containing the standard errors for each value of `x`.

For `type="discrete"` the optional arguments are slightly different. The argument `model` can be used, but it must assume the values `"ER"`, `"SYM"`, `"ARD"`, or a numeric matrix following `ace`.

Finally, for `type= "diversification"` models are so far `"birth-death"`, `"equal-extinction"`, and `"equal-specation"`, and `"Yule"`. It is also important to consider supplying the sampling fractions, `rho`, which is a vector of values between 0 and 1 of the same length as `trees`. If not provided the method will assume a sampling fraction of 1.0 for all trees - which is seldom true of empirical studies.

Value

An object of class `"ratebytree"` or an object of class `"posthoc.ratebytree"` in the case of the method `posthoc`.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Adams, D. C. (2012) Comparing evolutionary rates for different phenotypic traits on a phylogeny using likelihood. *Syst. Biol.*, **62**, 181-192.

Lewis, P. O. (2001) A likelihood approach to estimating phylogeny from discrete morphological character data. *Systematic Biology*, **50**, 913-925.

Nee, S., May, R. M. and Harvey, P. H. (1994) The reconstructed evolutionary process. *Philosophical Transactions of the Royal Society of London B*, **344**, 305-311.

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.

Stadler, T. (2012) How can we improve the accuracy of macroevolutionary rate estimates? *Systematic Biology*, **62**, 321-329.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J. (2018) Comparing the rates of speciation and extinction between phylogenetic trees. emphEcology and Evolution, **8**, 5303-5312.

Revell, L. J., Gonzalez-Valenzuela, L. E., Alfonso, A., Castellanos-Garcia, L. A., Guarnizo, C. E., and Crawford, A. J. (2018) Comparing evolutionary rates between trees, clades, & traits. *Methods Ecol. Evol.*, **9**, 994-1005.

See Also

`brownie.lite`, `fitMk`

`rateshift` *Find the temporal position of one or more rate shifts*

Description

Function finds the location of one or more rate shifts. `likSurface.rateshift` plots the likelihood surface (which is sometimes quite rugged).

Usage

```
rateshift(tree, x, nrates=1, niter=10, method="ML", ...)
## S3 method for class 'rateshift'
plot(x, ...)
likSurface.rateshift(tree, x, nrates=2, shift.range=NULL,
   density=20, plot=TRUE, ...)
```

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`. For `plot` method, `x` is an object of class `"rateshift"`.
`nrates`	number of rates.
`niter`	number of iterations of optimization routine to ensure convergence.
`method`	optimization method. Can be `"ML"` (maximum likelihood) or `"REML"` (restricted maximum likelihood).
`...`	optional arguments. In the case of the `plot` method, these will be passed to `plotSimmap`. For `rateshift`, optional arguments include: `tol`, tolerance; `plot` & `print`, logical values indicating whether to plot or print the progress of the optimization (default to `FALSE`); `quiet`, logical argument indicating whether to suppress all notifications (defaults to `FALSE`); `minL`, numeric value; and `fixed.shift`, either a vector of fixed shift points, or a logical value.

`shift.range`	for `likSurface.rateshift`.
`density`	for `likSurface.rateshift`.
`plot`	logical argument for `likSurface.rateshift`. If `plot=FALSE` then the surface is returned.

Value

A fitted object of class `"rateshift"`, or, in the case of `likSurface.rateshift`, a likelihood surface for the shift points.

Author(s)

Liam 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.

See Also

`brownie.lite`

`read.newick` *Newick or Nexus style tree reader*

Description

The function `read.newick` reads a simple Newick style tree from file.

The function `readNexus` reads a Nexus formatted tree, optionally with bootstrap values as node labels.

Usage

```
read.newick(file="", text, ...)
readNexus(file="", format=c("standard","raxml"))
```

Arguments

`file`	name of text file with single Newick style tree or multiple trees, one per line. For `readNexus` this should be a Nexus format tree.
`text`	character string containing tree.
`format`	file format (source) for `readNexus`. In the case of `format="standard"`, `read.nexus` from *ape* will be used internally. For `format="raxml"`, the parser assumes that bootstrap values have been stored as node labels in the format `[&label=bootstrap]`.
`...`	optional arguments to be passed to `scan`. Note that if the arguments `sep` or `what` are supplied this could generate an error. Useful optional arguments might include `skip` (number of lines to skip) and `nlines` (number of lines to read).

Details

The function `read.newick` is almost completely redundant with `read.tree`. At the time of development, it was more 'robust' than `read.tree` in that it didn't fail if the tree contained so-called 'singles' (nodes with only one descendant); however, `read.tree` can now handle singleton nodes without difficulty.

The function `readNexus` can read a Nexus formatted tree from file (like `read.nexus`); however, it can also parse the node labels as bootstrap values. This is the output format from the software *RAxML*. For Nexus tree files with complex node labels (e.g., from the software *MrBayes*) it will probably fail to parse node labels correctly, if at all.

Value

An object of class `"phylo"`, possibly containing singles (see `collapse.singles`); or an object of class `"multiPhylo"`.

Author(s)

Liam 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.

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

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

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`brownie.lite`, `evol.vcv`, `read.tree`, `read.nexus`

`reorder.backbonePhylo` *Reorders a backbone phylogeny*

Description

Function reorders an object of class `"backbonePhylo"`.

Usage

```
## S3 method for class 'backbonePhylo'
reorder(x, order="cladewise", ...)
```

Arguments

`x`	an object of class `"backbonePhylo"`.
`order`	order. See `reorder.phylo` for possible orderings.
`...`	optional arguments.

Value

An object of class `"backbonePhylo"`.

Author(s)

Liam 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.

See Also

`phylo.toBackbone`

`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", index.only=FALSE, ...)
```

Arguments

`tree`	a modified object of class `"phylo"`.
`order`	either `"cladewise"` or `"pruningwise"` (see `reorder.phylo`).
`index.only`	logical value indicating whether only an index should be returned.
`...`	other arguments.

Value

A modified object of class `"phylo"`.

Author(s)

Liam 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.

See Also

`reorder.phylo`, `plotSimmap`

`rep.phylo` *Replicate a tree or set of trees*

Description

S3 method `rep` for object of class `"phylo"` or `"multiPhylo"`. `repPhylo` is just an alias for `rep.phylo` and `rep.multiPhylo`.

Usage

```
## S3 method for class 'phylo'
rep(x, ...)
## S3 method for class 'multiPhylo'
rep(x, ...)
repPhylo(tree, times)
```

Arguments

`tree`	object of class `"phylo"`.
`times`	number of times to replicate tree.
`x`	for S3 method an object of class `"phylo"` or `"multiPhylo"`.
`...`	other arguments for `rep` (specifically, `times`).

Value

An object of class `"multiPhylo"`.

Author(s)

Liam 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.

See Also

`c.phylo`, `rep`

Examples

```
tree<-pbtree(n=100)
trees<-rep(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=NULL, interactive=FALSE, ...)
```

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. If not supplied, then the tree will be re-rooted *at* the node or tip.
`interactive`	logical value indicating whether to use interactive mode (defaults to `interactive= FALSE`).
`...`	arguments to be passed to `plotTree` for `interactive=TRUE` only.

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. Earlier versions also had an error related to node labels. This should be fixed in phytools>=0.4-47.

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.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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). Note that `...` is passed internally to `fitMk`, but should be used in this way with caution because any arguments that conflict with the default arguments of the method will cause the function execution to fail. The most practical use of this would be to force a particular value of the transition matrix, `Q`, via the argument `fixedQ`.

Details

This function calls `fitMk` internally. `fitMk` uses some code adapted from `ace` in the ape package.

`print` and `plot` methods are now available for the object class.

Value

An object of class `"rerootingMethod"` containing at least 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`>

References

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Yang, Z., Kumar, S., Nei, M. (1995) A new method of inference of ancestral nucleotide and amino acid sequences. *Genetics*, **141**, 1641-1650.

See Also

`ace`, `fitMk`, `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, ...)
```

Arguments

`tree`	a phylogenetic tree in modified `"phylo"` format with a discrete character mapping (e.g., see `read.simmap` or `make.simmap`).
`...`	other arguments, such as `depth`.

Details

Replaces `rescaleTree` (now `rescale.phylo`) in the 'geiger' package for SIMMAP style trees.

Value

A phylogenetic tree in modified `"phylo"` format.

Author(s)

Liam 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.

See Also

`make.simmap`, `read.simmap`

`resolveNode` *Compute all possible resolutions of a node or all nodes in a multifurcating tree*

Description

This functions resolves a single multifurcation or all multifurcations in a tree in all possible ways. If the input tree has edge lengths, then the resolutions will use internal edges of zero length.

Usage

```
resolveNode(tree,node)
resolveAllNodes(tree)
```

Arguments

`tree`	an object of class `"phylo"`.
`node`	for `resolveNode`, the node with a polytomy to resolve.

Details

For `resolveNode` applied to a multifurcation with n descendants, the number of resolved trees will be equal to the number of possible rooted trees of n taxa. (For instance, three for a trifurcation, 15 for a quadrifurcation, and so on.)

For `resolveAllNodes` the number of fully resolved trees will be equal to the product of numbers for `resolveNode` applied to each multifurcation separately. (For instance, 45 for a tree containing one trifurcation and one quadrifurcation.)

Value

An object of class `"multiPhylo"` - or, if the input tree is already fully resolved, an object of class `"phylo"` indentical to `tree`.

Author(s)

Liam 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.

See Also

`multi2di`

`rotateNodes` *Rotates a node or set of nodes in a phylogenetic tree*

Description

The function `rotateNodes` is a simple wrapper for `rotate` which rotates a set of nodes or all nodes.

The function `rotate.multi` finds all possible rotations around a multifurcating node, given by `node`. This will be an object of class `"multiPhylo"`, assuming that the node specified is indeed a multifurcation.

The function `allRotations` computes all possible rotated trees for a given input phylogeny. For a binary tree, this is generally two raised to the power of the number of internal nodes (so a very large number, if N is even modest in size).

Usage

```
rotateNodes(tree, nodes, polytom=c(1,2), ...)
rotate.multi(tree, node)
allRotations(tree)
```

Arguments

`tree`	object of class `"phylo"`.
`nodes`	either a single node number to rotate, a vector of node numbers, or the string `"all"`.
`polytom`	a vector of mode numeric and length two specifying the two clades that should be exchanged in a polytomy (see `rotate`).
`node`	a single node to rotate (in the case of `rotate.multi`).
`...`	optional arguments.

Details

All three functions also address the problem that the product of multiple rotations from `rotate` can be non-compliant with the implicit `"phylo"` standard because the tip numbers in `tree$edge` are not in numerical order `1:n` for `n` tips.

Value

An object of class `"phylo"` (i.e., a phylogenetic tree), in the case of `rotateNodes`, or an object of class `"multiPhylo"` for `rotate.multi` or `allRotations`.

Author(s)

Liam 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.

`roundBranches` *Rounds the branch lengths of a tree*

Description

This function rounds the branch lengths of a tree or trees and reconciles any mappings for objects of class `"simmap"` or `"multiSimmap"`.

Usage

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

Arguments

`tree`	an object of class `"phylo"`, `"multiPhylo"`, `"simmap"`, or `"multiSimmap"`.
`digits`	number of digits for rounding. Passed to `round`.

Value

An object of class `"phylo"`, `"multiPhylo"`, `"simmap"`, or `"multiSimmap"`, with rounded edge lengths.

Author(s)

Liam 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.

`roundPhylogram` *Plot a round or sigmoidal phylogram or cladogram*

Description

Function plots one or multiple round phylograms or a sigmoidal phylogram or cladogram.

Usage

```
roundPhylogram(tree, fsize=1.0, ftype="reg", lwd=2, mar=NULL, offset=NULL,
   direction="rightwards", type="phylogram", xlim=NULL, ylim=NULL, ...)
sigmoidPhylogram(tree, ...)
```

Arguments

`tree`	an object of class "phylo" or "multiPhylo" containing one or multiple phylogenies.
`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.
`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].
`offset`	offset for the tip labels.
`direction`	plotting direction. Only the option `direction="rightwards"` is presently supported.
`type`	plot type. Can be `"phylogram"` or `"cladogram"`. If `type="cladogram"` then the branch lengths are not necessary (and, indeed, are not used).
`xlim`	x-limits for the plot.
`ylim`	y-limits for the plot.
`...`	optional arguments. In the case of `roundPhylogram`, these are mostly as in `plotTree` and `plotSimmap`.

Details

The underscore character `"_"` is automatically swapped for a space in tip labels, as in `plotSimmap`.

Value

Plots a tree.

Author(s)

Liam 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.

See Also

`plotSimmap`, `plotTree`

Examples

```
data(mammal.tree)
roundPhylogram(compute.brlen(mammal.tree),
    fsize=0.8,ftype="i")
data(anoletree)
sigmoidPhylogram(anoletree,fsize=0.6,
ftype="i",direction="upwards")
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margin to default
```

`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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`setMap` *Set color map for various phylogenetic objects of classes*

Description

Function to change the color map (ramp) in an object of class `"contMap"`, `"densityMap"`, `"phyloScattergram"`, or `"multirateBM_plot"`.

Usage

```
setMap(x, ...)
## S3 method for class 'contMap'
setMap(x, ...)
## S3 method for class 'densityMap'
setMap(x, ...)
## S3 method for class 'phyloScattergram'
setMap(x, ...)
## S3 method for class 'multirateBM_plot'
setMap(x, ...)
```

Arguments

`x`	an object of class `"contMap"`, `"densityMap"`, `"phyloScattergram"`, or `"multirateBM_plot"`.
`...`	arguments to be passed to `colorRampPalette`. Also, the argument `invert` which (if `invert=TRUE`) will just flip the current color ramp.

Value

An object of class `"contMap"`, `"densityMap"`, `"phyloScattergram"`, or `"multirateBM_plot"`.

Author(s)

Liam 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.

Revell, L. J. 2013. Two new graphical methods for mapping trait evolution on phylogenies. *Methods in Ecology and Evolution*, **4**, 754-759.

See Also

`contMap`, `densityMap`, `multirateBM`

`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`>

References

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

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

`sim.history` *Simulate character history or a discrete character at the tips of the tree under some model*

Description

The function `sim.history` 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.

The function `sim.Mk` simulates the states for a discrete character at the tips of the tree only.

Finally, the function `sim.multiMk` is the same as `sim.Mk` except that it permits the user to simulate under different values of `Q` in different parts of the tree.

Usage

```
sim.history(tree, Q, anc=NULL, nsim=1, direction=c("column_to_row",
"row_to_column"), ...)
sim.Mk(tree, Q, anc=NULL, nsim=1, ...)
sim.multiMk(tree, Q, anc=NULL, nsim=1, ...)
```

Arguments

`tree`	a phylogenetic tree as an object of class `"phylo"`. For the case of `sim.multiMk` `tree` should be an object of class `"simmap"` in which the regimes for simulation have been mapped onto the tree.
`Q`	a matrix containing the instantaneous transition rates between states. Note that for `sim.history` *by default* (i.e., when `direction="column_to_row"`, see below) normally this is the *transpose* of the matrix produced by `fitDiscrete` in the geiger package or `make.simmap` in phytools; that is to say the transition rate from `i` -> `j` should be given by `Q[j,i]`. However, if your matrix is properly conformed (i.e., rows *or* columns sum to 0), then `sim.history` will attempt to transpose your matrix correctly & will return an informative message (if `message=TRUE`, see below). For `sim.Mk` and `sim.multiMk` this matrix has the same conformation as in `fitContinuous` and `make.simmap`. For `sim.multiMk` `Q` should be a list of transition matrices with names that correspond to the states mapped onto the tree.
`anc`	an optional value for the state at the root node; if `NULL`, a random state will be assigned. `anc` can be a vector of states, in which one of the states will be chosen randomly for each simulation. For `sim.history` `anc` can be a vector of probabilities with names, in which case a state will be chosen in proportion to the given probabilities.
`nsim`	number of simulations.
`direction`	row/column direction of the input transition matrix, `Q`. `direction="column_to_row"` indicates that the transition rate from `i` -> `j` should be given by `Q[j,i]`, while `direction="row_to_column"` indicates the converse.
`...`	other optional arguments. Currently only `message`, a logical value indicating whether or not to turn on informational messages (defaults to `message=TRUE`).

Value

`sim.history` returns an object of class `"simmap"` (a tree with a mapped discrete character) or `"multiSimmap"` for <code>nsim</code> greater than one.

`sim.Mk` and `sim.multiMk` return a factor with the states of our discrete character at the tips of the tree only.

Author(s)

Liam 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.

See Also

`fitMk`, `fitmultiMk`, `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, σ_x^2.
`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"` indicating 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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`fastBM`, `ratebystate`

`sim.rates` *Brownian or OU simulation with multiple evolutionary regimes*

Description

The function `sim.rates` conducts BM simulation on a tree with multiple rates.

The function `multiOU` conducts multi-regime OU simulations on the tree under a range of conditions.

Usage

```
sim.rates(tree, sig2, anc=0, nsim=1, internal=FALSE, plot=FALSE)
multiOU(tree, alpha, sig2, theta=NULL, a0=NULL, nsim=1, internal=FALSE, ...)
```

Arguments

`tree`	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`.
`alpha`	single value or vector of values of the OU α parameter.
`theta`	single value or vector of values of the OU θ parameter.
`a0`	optional value of the root state. Defaults to zero.
`...`	optional arguments.

Details

`multiOU` uses a difference equation approximation of the OU process.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

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

`simBMphylo` *Creates a graphical illustration of Brownian motion evolution on a phylogeny*

Description

The function simulates a discrete-time pure-birth phylogeny (for fixed *N* and *t* using rejection sampling) and then discrete-time Brownian motion on that tree. It then proceeds to generating a plot of the results.

Usage

```
simBMphylo(n, t, sig2, plot=TRUE, ...)
## S3 method for class 'simBMphylo'
plot(x, ...)
```

Arguments

`n`	number of taxa to simulate in the output tree.
`t`	total time for the simulation.
`sig2`	the rate of evolution under Brownian motion, σ^2, or a vector of rates. If the latter the length of the vector must exactly match `t`, otherwise the first element of `sig2` will just be duplicated `t` times.
`plot`	optional logical value indicating whether or not the simulated object should be plotted.
`...`	optional arguments to be passed to the `plot` method.
`x`	in `plot` method, object of class `"simBMphylo"`.

Value

An object of class `"simBMphylo"` or a plot.

Author(s)

Liam 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.

See Also

`bmPlot`, `fastBM`, `pbtree`

Examples

```
set.seed(777)
simBMphylo(n=6,t=100,sig2=0.01)
## reset par to defaults
par(mfrow=c(1,1))
par(mar=c(5.1,4.1,4.1,2.1))
```

`skewers` *Matrix comparison using the method of random skewers*

Description

This function performs the random skewers matrix comparison method of Cheverud (1996; also see Cheverud & Marroig 2007 for more details). In addition, it includes a more robust hypothesis test in which random covariance matrices are simulated under a variety of models, and then the mean correlation between response vectors to random skewers are computed.

Usage

```
skewers(X, Y, nsim=100, method=NULL)
```

Arguments

`X`	covariance matrix.
`Y`	covariance matrix.
`nsim`	number of random vectors.
`method`	method to generate a null distribution of the random skewers correlation between matrices. If `method=NULL` then the correlation will be compared to the correlation between random vectors; however this test has type I error substantially above the nominal level for ostensibly random matrices. Other values of `method` will be passed as `covMethod` to `genPositiveDefMat` for a more robust hypothesis test (see below). Recommended values include `"unifcorrmat"`.

Value

A list with the following components:

`r`	mean random skewers correlation.
`p`	p-value from simulation.

Author(s)

Liam Revell <`liam.revell@umb.edu`>

References

Cheverud, J. M. (1996) Quantitative genetic analysis of cranial morphology in the cotton-top (Saguinus oedipus) and saddle-back (S. fuscicollis) tamarins. *J. Evol. Biol.*, **9**, 5–42.

Cheverud, J. M. & Marroig, G. (2007) Comparing covariance matrices: Random skewers method compared to the common principal components model. *Genetics & Molecular Biology*, **30**, 461–469.

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`splitEdgeColor` *Split edge colors when descendant edges have different mapped states*

Description

This function splits the vertical line colors to match the daughter edges when the daughters have different states. Mostly to be used with trees generated using `paintBranches` or `paintSubTree`. Also used internally by `plotSimmap`.

Usage

```
splitEdgeColor(tree,colors,lwd=2)
```

Arguments

`tree`	object of class `"simmap"`.
`colors`	named vector of colors to be used for plotting.
`lwd`	width of the plotted lines.

Author(s)

Liam 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.

See Also

`plotSimmap`

`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)
plotTree.splits(tree, splits=NULL, file=NULL, fn=NULL, ...)
```

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.
`splits`	for `plotTree.splits` relative positions (from 0 to 1) to split the tree across pages or devices.
`file`	filename if saving to a PDF file is desired. Otherwise will plot to the default plotting device.
`fn`	function to be executed on each plotted page. For instance, might be: `function() cladelabels()` if clade labels are desired.
`...`	other arguments to be passed to `plotTree`.

Value

Plots a tree.

Author(s)

Liam 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.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`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>`

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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 `https://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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`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=10000, control=list(), ...)
```

Arguments

`tree`	an object of class `"phylo"`.
`X`	a matrix or data frame containing values for a 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.
`...`	other optional arguments. The argument `auto.tune` adjusts the proposal distribution to target an proposal acceptance rate (of 0.234, by default).

Details

The `plot` method for the object class can be used to generate a three panel plot showing the likelihood profile, the mean acceptance rates (using a sliding window), and a profile plot for the correlation coefficient, *r*.

The `density` method for the object can be used to plot a posterior density of the correlation coefficient, *r*. This posterior density is of class `"density.threshBayes"` which can also be plotted using an `plot` method.

Discrete characters must be binary, but can be coded as any factor.

Value

This function returns an object of class `"threshBayes"` consisting of a list with at least the following 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`>

References

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J. (2014) Ancestral character estimation under the threshold model from quantitative genetics. *Evolution*, **68**, 743-759.

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>`

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J. (2014) Ancestral character estimation under the threshold model from quantitative genetics. *Evolution*, **68**, 743-759.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

Revell, L. J. (2014) Ancestral character estimation under the threshold model from quantitative genetics. *Evolution*, **68**, 743-759.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

`tree.grow` *Creates an animation of a tree growing from left-to-right or upwards*

Description

This function simulates a birth-death tree under user-defined conditions and then creates an animation of that tree growing from left-to-right in the plotting device, or upwards.

Usage

```
tree.grow(..., res=200, direction="rightwards", ladderize=TRUE)
```

Arguments

`...`	arguments to pass to `pbtree`.
`res`	number of steps (the resolution of the animation). This also corresponds to the number of frames that will be created if the animation is to be converted to a .gif file.
`direction`	the direction to plot the tree. Only `direction="rightwards"` (the default) and `direction="upwards"` are supported.
`ladderize`	logical value indicating whether or not to 'ladderize' the plotted tree. (Defaults to `TRUE`.)

Value

An object of class `"phylo"`.

Author(s)

Liam 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.

See Also

`pbtree`

Examples

```
## Not run:
## to create a .gif with ImageMagick installed
png(file="pbtree-
tree.grow(b=0.06,d=0.02,t=100)
dev.off()
system("ImageMagick convert -delay 5 -loop 0 *.png pbtree-anim.gif")
## End(Not run)
```

`treeSlice` *Slices the tree at a particular point and returns all subtrees, or the tree rootward of the point*

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.

It can also be used to crop the terminal fraction of a tree for `orientation="rootwards"`.

Usage

```
treeSlice(tree, slice, trivial=FALSE, prompt=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`).
`prompt`	logical value indicating whether or not the height of the slice should be given interactively.
`...`	for `prompt=TRUE`, other arguments to be passed to `plotTree`. In addition, the argument `orientation` can be used to specify whether the `"tipwards"` subtrees or the `"rootwards"` phylogeny are/is to be returned by the function call (using those two argument values, respectively).

Value

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

Author(s)

Liam 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.

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`.

Usage

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

Arguments

`tree`	tree as an object of class `"phylo"` or `"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

An object of class `"phylo"` or `"simmap"`.

Author(s)

Liam 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.

`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 and under different evolutionary models.

Usage

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

Arguments

`tree`	object of class `"phylo"`.
`anc.nodes`	logical value indicating whether or not to include ancestral nodes.
`...`	optional arguments including `internal` (synonym of `anc.nodes`) and `model` (can be `"BM"`, `"OU"`, or `"lambda"`.

Value

A matrix.

Author(s)

Liam 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.

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

Description

This function writes stochastic character mapped trees to file in multiple SIMMAP formats (Bollback 2006).

Usage

```
write.simmap(tree, file=NULL, append=FALSE, map.order=NULL, quiet=FALSE,
format="phylip", version=1.0)
```

Arguments

`tree`	an object of class `"simmap"` or `"multiSimmap"`.
`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.
`quiet`	logical value indicating whether or not to print a warning message when `map.order` is neither specified by a function argument or as an attribute of `tree`.
`format`	file format for output.
`version`	version of SIMMAP. Note that the options are `1.0` and `1.5`. `version=1.5` is generally recommended because in this format the tree can also be parsed by typical tree readers, but absent its mapped trait.

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.

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

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`make.simmap`, `read.simmap`, `plotSimmap`

`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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

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`>

References

Revell, L. J. (2012) phytools: An R package for phylogenetic comparative biology (and other things). *Methods Ecol. Evol.*, **3**, 217-223.

See Also

`write.simmap`, `write.nexus`

Index