

Package ‘phytools’

September 19, 2025

Version 2.5-2

Date 2025-09-18

Title Phylogenetic Tools for Comparative Biology (and Other Things)

Depends R ($\geq 3.5.0$), ape (≥ 5.7), maps

Imports clusterGeneration, coda, combinat, DEoptim, doParallel, expm, foreach, graphics, grDevices, MASS, methods, mnormt, nlme, numDeriv, optimParallel, parallel, phangorn ($\geq 2.3.1$), scatterplot3d, stats, utils

Suggests animation, geiger, plotrix, RColorBrewer, rgl

ZipData no

Description

A wide range of methods for phylogenetic analysis - concentrated in phylogenetic comparative biology, but also including numerous techniques for visualizing, analyzing, 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 trait data. 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 phenotype space or a onto a geographic map, and for visualizing correlated speciation between trees. Lastly, numerous functions are designed for reading, writing, analyzing, inferring, simulating, and manipulating phylogenetic trees and comparative data. For instance, there are functions for computing consensus phylogenies from a set, for simulating phylogenetic trees and data under a range of models, for randomly or non-randomly attaching species or clades to a tree, as well as for a wide range of other manipulations and analyses that phylogenetic biologists might find useful in their research.

License GPL (≥ 2)

URL <https://github.com/liamrevell/phytools>

Repository CRAN

Date/Publication 2025-09-19 05:10:02 UTC

NeedsCompilation no

Author Liam J. Revell [aut, cre]

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

Contents

phytools-package	6
add.arrow	6
add.color.bar	8
add.everywhere	9
add.random	10
add.simmap.legend	11
add.species.to.genus	12
aic.w	13
allFurcTrees	14
anc.Bayes	15
anc.ML	17
anc.trend	19
ancr	21
ancThresh	23
anoletree	25
ansi_phylo	29
applyBranchLengths	30
as.multiPhylo	31
as.Qmatrix	32
ave.rates	33
averageTree	34
bd	35
bind.tip	36
bind.tree.simmap	37
bmPlot	38
bounded_bm	40
branching.diffusion	42
brownie.lite	43
brownieREML	45
cladelabels	47
collapse.to.star	48
collapseTree	49
compare.chronograms	50
consensus.edges	51
contMap	52
cophylo	54
cospeciation	56
countSimmap	58
ctt	59
density.multiSimmap	60
densityMap	61
densityTree	63
describe.simmap	64
di2multi.simmap	66
dotTree	67
drop.clade	69

drop.leaves	69
drop.tip.contMap	70
drop.tip.multiSimmap	71
drop.tip.simmap	72
Dtest	73
edge.widthMap	74
edgeProbs	75
estDiversity	76
evol.rate.mcmc	77
evol.vcv	79
evolvcv.lite	81
exhaustiveMP	83
expand.clade	84
export.as.xml	85
fancyTree	86
fastAnc	88
fastBM	90
fastMRCA	91
findMRCA	92
fit.bd	94
fitBayes	96
fitcontMk	97
fitDiversityModel	98
fitMk	99
fitmultiBM	104
fitPagel	106
fitThresh	108
force.ultrametric	110
gammatest	111
gamma_pruning	112
genSeq	113
geo.legend	114
get.treepos	115
getCladesofSize	116
getDescendants	117
getExtant	118
getSisters	119
getStates	120
labelnodes	120
ladderize.simmap	121
lambda.transform	122
likMlambda	123
linklabels	124
locate.fossil	125
locate.yeti	126
ls.tree	127
ltt	127
ltt95	130

make.era.map	131
make.simmap	132
map.overlap	135
map.to.singleton	137
mapped.states	138
markChanges	139
matchNodes	140
mergeMappedStates	141
midpoint_root	142
minRotate	143
minSplit	144
modified.Grafen	145
mrp.supertree	146
multi.mantel	147
multiC	148
multirateBM	149
multiRF	151
nodeHeights	152
nodelabels.cophylo	153
optim.phylo.ls	154
orderMappedEdge	155
paintSubTree	156
paste.tree	157
pbtrees	158
pgls.Ives	160
phenogram	162
phyl.cca	164
phyl.pairedttest	165
phyl.pca	167
phyl.resid	169
phyl.RMA	170
phyl.vcv	172
phylANOVA	173
phylo.heatmap	174
phylo.impute	175
phylo.to.map	176
phylo.toBackbone	178
phyloDesign	179
phylomorphospace	180
phylomorphospace3d	182
phylosig	184
plot.backbonePhylo	186
plotBranchbyTrait	187
plotSimmap	189
plotThresh	191
plotTree	193
plotTree.datamatrix	194
plotTree.errorbars	195

plotTree.lollipop	196
plotTree.wBars	198
posterior.evolrate	200
posthoc	201
print.backbonePhylo	202
pscore	203
ratebystate	204
ratebytree	205
rateshift	207
read.newick	209
read.simmap	210
reorder.backbonePhylo	212
reorderSimmap	213
rep.phylo	214
reroot	215
rerootingMethod	216
rescale	217
rescale.simmap	218
resolveNode	220
rotateNodes	221
roundBranches	222
roundPhylogram	223
rstate	224
sampleFrom	225
setMap	226
sim.corr	227
sim.history	228
sim.ratebystate	229
sim.rates	230
simBMphylo	232
skewers	233
splitEdgeColor	234
splitplotTree	235
splitTree	236
starTree	237
strahlerNumber	238
threshBayes	239
threshDIC	241
threshState	242
to.matrix	243
tree.grow	243
treeSlice	244
untangle	245
vcvPhylo	246
write.simmap	247
writeAncestors	248
writeNexus	249

phytools-package	<i>phytools: Phylogenetic Tools for comparative biology (and other things)</i>
------------------	--

Description

phytools provides functions for phylogenetic biology: focused on phylogenetic comparative analysis, but including methods to simulate data and trees, to visualize trees and fitted models, and to analyze and manipulate phylogenetic trees and data.

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 <https://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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

add.arrow	<i>Add an arrow pointing to a tip or node on the tree</i>
-----------	---

Description

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 <code>last_plot.phylo</code> .
tip	label of tip or tip or node number; or vector of such values. If <code>tree=NULL</code> 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 (<code>arrl</code>) in the units of the plot; the length of the arrowhead (<code>hedl</code>); the total angle between the wings in the arrowhead (<code>angle</code>); the line width for the plotted lines (<code>lwd</code>); the offset from the tip or end of tip label, in character widths (<code>offset</code>); and the color (<code>col</code>).

Details

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[nodelabels](#)

Examples

```
## load tree
data(anoletree)

## plot tree
plotTree(anoletree, type="fan", fsize=0.7, ftype="i")

## add lwd=5 arrows in black
add.arrow(anoletree, tip=c("cuvieri", "krugi",
  "pulchellus", "poncensis", "stratulus",
  "evermanni", "cooki", "cristatellus",
  "gundlachi", "occultus"), lwd=5, arrl=1)

## overlay thinner arrows to create outline effect
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,arr1=1)
add.arrow(anoletree,tip=c("cooki","cristatellus",
  "gundlachi"),col="brown",lwd=3,arr1=1)
add.arrow(anoletree,tip="occultus",col="darkgrey",
  lwd=3,arr1=1)

## add legend
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")

## reset plotting parameters
par(mar=c(5.1,4.1,4.1,2.1)) ## reset margins to default
```

add.color.bar	<i>Add color bar to a plot</i>
---------------	--------------------------------

Description

This function adds a color gradient legend to an existing plot.

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.

Details

This function was designed to add 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.

The function is also used internally by S3 methods [plot.contMap](#) and [plot.densityMap](#), as well as by [errorbar.contMap](#).

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.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[contMap](#), [densityMap](#), [errorbar.contMap](#), [plotBranchbyTrait](#)

add.everywhere	<i>Add tip to all edges in a tree</i>
----------------	---------------------------------------

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[allFurcTrees](#), [exhaustiveMP](#)

add.random	<i>Add tips at random to the tree</i>
------------	---------------------------------------

Description

This function adds new tips at random to a tree with branch lengths.

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 assigned 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

If no edge lengths are provided, and the tree is ultrametric, then edge lengths are assigned to keep the tree ultrametric.

The probability that a new tip is added along any branch is directly proportional to the length of the branch.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

add.simmap.legend	<i>Add legend to stochastically mapped tree</i>
-------------------	---

Description

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 indicating 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".

Details

Redundant with [legend](#), which is much more powerful but somewhat more complicated to use.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[plotSimmap](#)

add.species.to.genus	<i>Add species to genus on a phylogeny or bind simulated species subtrees to a backbone genus tree</i>
----------------------	--

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[add.random](#), [bind.tip](#)

aic.w

Computes Akaike weights

Description

Computes Akaike weights based on a set of AIC values.

Usage

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

Arguments

aic	vector of Akaike Information Criterion (AIC; Akaike, 1974) values for different fitted models. If the vector has names, these names will be inherited by the vector returned by the function.
-----	---

Details

This function computes Akaike weights from a set of AIC values obtained from 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.

Value

A vector of Akaike weights.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Akaike, H. (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, **19**, 716-723.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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 extreme 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

Examples

```
## compute & plot all bi- and multifurcating
## trees for six taxa
trees<-allFurcTrees(n=6)

## plot trees
par(mfrow=c(16,15))
nulo<-sapply(trees,plot,type="unrooted",
             no.margin=TRUE)

## reset par
par(mfrow=c(1,1))
```

anc.Bayes	<i>Bayesian ancestral character estimation</i>
-----------	--

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[ace](#), [anc.ML](#), [anc.trend](#), [ancr](#), [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 object to review estimates
print(mcmc,printlen=20)

## plot likelihood trace and root density
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)

## reset par to default
par(mfrow=c(1,1))
```

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

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

model	model of continuous character evolution on the tree. It's possible that only model="BM" & model="EB" work in the present version as model="OU" has not been thoroughly tested & some bugs were reported for an earlier version.
...	optional 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 issue has been ameliorated in **phytools** $\geq 0.2-48$ 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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](#), [ancr](#) [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 results
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 a 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-contemporaneous tips (i.e., a tree with some fossil species).

The function uses `optim` with method= "L-BFGS-B" internally; 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 <code>optim()</code> (0 is good).

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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")
```

```
## reset par to defaults
par(mfrow=c(1,1))
```

ancr

Compute marginal or joint ancestral state estimates

Description

By default, ancr computes marginal ancestral states, also known as empirical Bayes posterior probabilities, conditioning on the fitted (or set) model of object.

Can also perform *joint* ancestral state estimation, if the optional argument type is set to type="joint".

hide.hidden merges hidden states (if any).

Usage

```
ancr(object, ...)
hide.hidden(object, ...)
## S3 method for class 'ancr'
plot(x, args.plotTree=list(...), args.nodelabels=list(...), ...)
```

Arguments

object	fitted model (presently object of class "bounded_bm", "fitgammaMk", "fitHRM", "fitMk", "fitpolyMk", "fitThresh", or "fitmultiBM"), or a set of models in the form of a table from anova comparison. For hide.hidden, an object of class "ancr".
...	optional arguments. Depending on the input object class, may include type (whether to undertake marginal or joint estimation, defaults to type="marginal") and local (whether to do global or local estimation, following Pagel 1999, defaults to local=FALSE).
x	in the case of plot.ancr, an object of class "ancr".
args.plotTree	arguments to be passed to plotTree , in a list.
args.nodelabels	arguments to be passed to nodelabels , in a list.

Details

If the optional argument tips=TRUE, then the matrix returned contains empirical Bayes posterior probabilities (marginal scaled likelihoods) for both tips *and* internal nodes. Otherwise (the default) only node marginal states are returned.

If the input object is a set of models (in the form of an anova table), then ancr will compute model-averaged marginal ancestral states (for type="marginal", unless the optional argument weighted=FALSE, in which case only the best-supported model is used).

Value

An object of class "ancr".

For most input object types, in the case of type="marginal", this object consists of a matrix of marginal (empirical Bayes) probabilities and a likelihood. In the case of type="joint", the object contains a set of most-likely internal node states stored in a data frame.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Pagel, M. (1999) The Maximum Likelihood approach to reconstructing ancestral character states of discrete characters on phylogenies. *Systematic Biology*, **3**, 612-622.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[fitMk](#)

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 ARD model
ard_fmode<-fitMk(sunfish.tree,fmode,model="ARD",
  pi="fitzjohn")

## compute ancestral states
anc_fmode<-ancr(ard_fmode)

## plot the results
par(mfrow=c(2,1))
cols<-setNames(c("blue","red"),levels(fmode))
plot(anc_fmode,
  args.plotTree=list(lwd=2,direction="upwards",
    mar=c(0.1,1.1,2.1,1.1),fsize=0.8),
  args.nodelabels=list(piecol=cols),
  args.tiplabels=list(cex=0.3),
  legend="bottomright")
mtext("a) marginal states under ARD model",adj=0)

## fit ER model
```

```

er_fmde<-fitMk(sunfish.tree,fmode,model="ER",
  pi="fitzjohn")

## compare models
aov_fmde<-anova(er_fmde,ard_fmde)

## compute model-averaged ancestral states
anc_fmde_model.averaged<-ancr(aov_fmde)
plot(anc_fmde_model.averaged,
  args.plotTree=list(lwd=2,direction="upwards",
    mar=c(0.1,1.1,2.1,1.1),fsize=0.8),
  args.nodelabels=list(piecol=cols),
  args.tiplabels=list(cex=0.3),
  legend="bottomright")
mtext("b) marginal states model-averaging ER & ARD models",
  adj=0)

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

```

ancThresh	<i>Ancestral character estimation under the threshold model using Bayesian MCMC</i>
-----------	---

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

According to the threshold model from evolutionary quantitative genetics, values for our observed discrete character are determined by an unseen continuous trait, normally referred to as liability. Every time the value for liability crosses a threshold, the observed discrete character changes in state.

Felsenstein (2012) first had the insight that this model could be used to study the evolution of discrete character traits on a reconstructed phylogenetic tree.

This function uses Bayesian MCMC to sample ancestral liabilities and thresholds for a discrete character evolution under the threshold model.

`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, along 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.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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)

## reset par to default
par(mar=c(5.1,4.1,4.1,2.1))
## End(Not run)
```

anoletree

Phylogenetic datasets

Description

Various phylogenetic datasets for comparative analysis.

Usage

```
data(anole.data)
data(anoletree)
data(ant.geog)
data(ant.tree)
data(bat.tree)
data(bat_virus.data)
data(betaCoV.tree)
data(bonyfish.data)
data(bonyfish.tree)
data(butterfly.data)
data(butterfly.tree)
data(cordylid.data)
data(cordylid.tree)
data(darter.tree)
data(eel.data)
data(eel.tree)
data(elapidae.tree)
```

```

data(flatworm.data)
data(flatworm.tree)
data(liolaemid.tree)
data(mammal.data)
data(mammal.geog)
data(mammal.tree)
data(primate.data)
data(primate.tree)
data(salamanders)
data(sunfish.data)
data(sunfish.tree)
data(tortoise.geog)
data(tortoise.tree)
data(tropidurid.data)
data(tropidurid.tree)
data(vertebrate.data)
data(vertebrate.tree)
data(wasp.data)
data(wasp.trees)
data(whale.tree)

```

Format

anoletree is an object of class "simmap". anole.data is a data frame.
 ant.tree is an object of class "phylo". ant.geog is a vector.
 bat.tree and betaCoV.tree are objects of class "phylo". bat_virus.data is a data frame.
 bonyfish.tree is an object of class "phylo". bonyfish.data is a data frame.
 butterfly.tree is an object of class "phylo". butterfly.data is a data frame.
 cordylid.tree is an object of class "phylo". cordylid.data is a data frame.
 darter.tree is an object of class "phylo".
 eel.tree is an object of class "phylo". eel.data is a data frame.
 elapidae.tree is an object of class "phylo".
 flatworm.tree is an object of class "phylo". flatworm.data is a data frame.
 liolaemid.tree is an object of class "phylo". liolaemid.data is a data frame.
 mammal.tree is an object of class "phylo". mammal.data is a data frame. mammal.geog is a numerical matrix.
 primate.tree is an object of class "phylo". primate.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.
 tortoise.tree is an object of class "phylo". tortoise.geog is a data frame.
 tropidurid.tree is an object of class "simmap". tropidurid.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.
 whale.tree is an object of class "phylo".

Details

`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).

`ant.tree` is a phylogeny containing 84 species of fungus farming attine ants. `ant.geog` is a vector containing biogeographic information for the species of the tree. Both the phylogeny and biogeographic information were originally published by Branstetter et al. (2017).

`bat.tree` is a ML phylogeny of bats from the families Mormoopidae, Molossidae, and Phyllostomidae, based on GenBank accession numbers reported in Caraballo (2022) and estimated using **phangorn**. `betaCoV.tree` is a ML phylogenetic tree of betacoronaviruses based on GenBank accession numbers reported in Caraballo (2022) and estimated using **phangorn**. `bat_virus.data` is an association table of host and virus from Caraballo (2022).

`butterfly.tree` and `butterfly.data` are a phylogeny and dataset of habitat use in Mycalesina butterflies from Halali et al. (2020). The phylogeny of that study was adapted from one originally published by Brattstrom et al. (2020).

`bonyfish.tree` and `bonyfish.data` are a phylogeny and dataset of spawning mode and parental care in bony fishes from Benun Sutton and Wilson (2019). Their phylogenetic tree was adapted from a tree originally published by Betancur-R et al. (2017)

`cordylid.tree` and `cordylid.data` are a phylogeny and dataset of morphological traits for three different principal components axes from Broeckhoven et al. (2016).

`darter.tree` is a phylogeny of 201 species of darter fish (Etheostominae) from Near et al. (2011).

`eel.tree` and `eel.data` are a phylogeny and dataset of feeding mode and maximum total length from 61 species of elopomorph eels from Collar et al. (2014).

`elapidae.tree` is a reconstructed phylogeny containing 175 species of snakes of the family Elapidae from Lee et al. (2016).

`flatworm.tree` and `flatworm.data` are a phylogeny and dataset of habitat preferences for flatworms from Benitez-Alvarez et al. (2020). `flatworm.tree` has been made ultrametric using penalized likelihood.

`liolaemid.tree` and `liolaemid.data` are a phylogenetic tree and phenotypic trait dataset of lizards from the family Liolaemidae from Esquerre et al. (2019).

`mammal.tree` and `mammal.data` are the phylogeny and dataset for mammal body size and home range size from Garland et al. (1992). `mammal.geog` is a matrix containing geographic coordinates for various of the species in `mammal.tree` pulled from citizen science observations registered on the *iNaturalist* web platform.

`primate.tree` and `primate.data` are a phylogeny and phenotypic trait dataset from Kirk and Kay (2004).

`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).

`tortoise.tree` and `tortoise.geog` are a phylogeny and dataset of geographic coordinates for Galapagos tortoises. The geographic locality information was approximated from Figure 1 of

Poulakakis et al. (2020), and the tree was estimated from nucleotide sequence data published with the same study.

tropidurid.tree and tropidurid.data are the phylogeny and bivariate quantitative trait dataset of tropidurid lizards from Revell et al. (2022).

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).

whale.tree is a phylogeny of cetaceans originally published in Steeman et al. (2009).

Source

Benitez-Alvarez, L., A. Maria Leal-Zanchet, A. Ocegüera-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 epigeal group sheltering in caves? *Molecular Phylogenetics and Evolution*, **145**, 106709.

Benun Sutton, F., and A. B. Wilson (2019) Where are all the moms? External fertilization predicts the rise of male parental care in bony fishes. *Evolution*, **73**, 2451-2460.

Betancur-R., R., E. O. Wiley, G. Arratia, A. Acero, N. Bailly, M. Miya, G. Lecointre, and G. Orti (2017) Phylogenetic classification of bony fishes. *BMC Ecology and Evolution*, **17**, 162.

Branstetter, M. G., A. Jesovnik, J. Sosa-Calvo, M. W. Lloyd, B. C. Faircloth, S. G. Brady, and T. R. Schultz (2017) Dry habitats were crucibles of domestication in the evolution of agriculture in ants. *Proceedings of the Royal Society B*, **284**, 20170095.

Brattstrom, O., K. Aduse-Poku, E. van Bergen, V. French, and P. M. Brakefield (2020) A release from developmental bias accelerates morphological diversification in butterfly eyespots. *Proceedings of the National Academy of Sciences*, **177**, 27474-27480.

Broeckhoven, C., G. Diedericks, C. Hui, B. G. Makhubo, P. le Fras N. Mouton (2016) Enemy at the gates: Rapid defensive trait diversification in an adaptive radiation of lizards. *Evolution*, **70**, 2647-2656.

Caraballo, D. A. (2022) Cross-species transmission of bat coronaviruses in the Americas: Contrasting patterns between alphacoronavirus and betacoronavirus. *Microbiology Spectrum*, **10**, e01411-22.

Collar, D. C., P. C. Wainwright, M. E. Alfaro, L. J. Revell, and R. S. Mehta (2014) Biting disrupts integration to spur skull evolution in eels. *Nature Communications*, **5**, 5505.

Esquerre, D., D. Ramirez-Alvarez, C. J. Pavon-Vazquez, J. Troncoso-Palacios, C. F. Garin, J. S. Keogh, and A. D. Leache (2019) Speciation across mountains: Phylogenomics, species delimitation and taxonomy of the *Liolaemus leopardinus* clade (Squamata, Liolaemidae). *Molecular Phylogenetics and Evolution*, **139**, 106524.

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.

Kirk, E. C., and R. F. Kay (2004) The evolution of high visual acuity in the Anthropeoidea. In: Ross, C. F., Kay R. F. (Eds), *Anthropoid Origins. Developments in Primatology: Progress and Prospects*, 539-602. Springer, Boston, MA.

- Halali, S., E. van Bergen, C. J. Breuker, P. M. Brakefield, and O. Brattstrom (2020) Seasonal environments drive convergent evolution of a faster pace-of-life in tropical butterflies. *Ecology Letters*, **24**, 102-112.
- 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.
- Lee, M. S. Y., K. L. Saunders, B. King, and A. Palci (2016) Diversification rates and phenotypic evolution in venomous snakes (Elapidae). *Royal Society Open Science*, **3**, 150277.
- 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.
- Near, T. J., C. M. Bossu, G. S. Bradburd, R. L. Carlson, R. C. Harrington, P. R. Hollingsworth Jr., B. P. Keck, and D. A. Etnier. (2011) Phylogeny and temporal diversification of darters (Percidae: Etheostomatinae). *Systematic Biology*, **60**, 565-595.
- Poulakakis, N., J. M. Miller, E. L. Jensen, L. B. Beheregaray, M. A. Russello, S. Glaberman, J. Boore, and A. Caccone. (2020) Colonization history of Galapagos giant tortoises: Insights from mitogenomes support the progression rule. *Journal of Zoological Systematics and Evolutionary Research*, **58**, 1262-1275.
- 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 (2022) A simple hierarchical model for heterogeneity in the evolutionary correlation a phylogenetic tree. *PeerJ*, **10**, e13910.
- Steeman, M. E., M. B. Hebsgaard, R. E. Fordyce, S. Y. W. Ho, D. L. Rabosky, R. Nielsen, C. Rahbek, H. Glenner, M. V. Sorensen, and E. Willerslev (2009) Radiation of extant cetaceans driven by restructuring of the oceans. *Systematic Biology*, **58**, 573-585.

ansi_phylo

Graph a phylogeny using only ANSI characters

Description

Plots a phylogeny in a silly, ANSI graphics style.

Usage

```
ansi_phylo(tree, vertical=c("|", "-"), ...)
```

Arguments

tree	object of class "phylo".
vertical	character for vertical lines
...	optional arguments.

Value

A plotted tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[plotTree](#), [splinePhylogram](#)

Examples

```
## load trees
data(salamanders)
data(sunfish.tree)

## plot using defaults
ansi_phylo(salamanders)

## plot using special characters
ansi_phylo(sunfish.tree,vertical="//",horizontal="*",
  xlim=c(0,0.3),fsize=0.9)

## reset par to default
par(mar=c(5.1,4.1,4.1,2.1))
```

applyBranchLengths	<i>Applies the branch lengths of a reference tree to a target</i>
--------------------	---

Description

Applies the branch lengths of a reference tree to a target.

Usage

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

Arguments

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

Details

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.

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

as.multiPhylo	<i>Conversion to object of class "multiPhylo"</i>
---------------	---

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

as.Qmatrix

Convert a fitted Mk model to a Q-matrix

Description

Extracts a **Q**-matrix from a fitted *Mk* 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 *Mk* model. (For instance, an object of class "*fitMk*".) In the case of `print.Qmatrix`, an object of class "*Qmatrix*".

... optional arguments.

Details

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

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>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[fitMk](#)

ave.rates	<i>Average the posterior rates</i>
-----------	------------------------------------

Description

Internal function for [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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

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

Description

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

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

Details

The function `averageTree` tries to find the (hypothetical) tree topology and branch lengths that has a minimum distance to all the trees in an input set, according to some user-specified tree distance measure.

The function `ls.consensus` computes the least-squares consensus tree (Lapointe & Cucumel, 1997) from a set of input trees.

Finally, the function `minTreeDist` finds the tree in the input set that is a minimum distance to all the other trees in the set. (This contrasts with `averageTree` which can return a tree not in the input set.)

Value

An object of class "phylo" with edge lengths.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

- Lapointe, F.-J., G. Cucumel (1997) The average consensus procedure: Combination of weighted trees containing identical or overlapping sets of taxa. *Systematic Biology*, **46**, 306-312.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

bd	<i>Convert object of class "birthdeath" to raw birth & death rates</i>
----	--

Description

Converts an object of class "birthdeath" (from [birthdeath](#)) to a vector with the ML birth & death rates.

Usage

```
bd(x)
```

Arguments

x object of class "birthdeath".

Details

This is now obsolete 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.

Value

A vector.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[birthdeath](#), [fit.bd](#), [fit.yule](#)

bind.tip	<i>Attaches a new tip to a tree</i>
----------	-------------------------------------

Description

Adds a new tip to the tree.

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 <i>rootward</i> 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 " <i>species</i> " type: <code>which(tree\$tip.label=="species")</code> .
position	distance <i>rootward</i> 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

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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Examples

```
## set seed
set.seed(123)

## subdivide plotting device
par(mfrow=c(3,1))

## generate tree
tree<-pbtree(b=0.1, n=10)

## plot original tree
plotTree(tree,mar=c(2.1,0.1,2.1,0.1))
axisPhylo()
mtext("a"),adj=0)

## 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)

## re-plot to see the result
plotTree(tree,mar=c(2.1,0.1,2.1,0.1))
axisPhylo()
mtext("b"),adj=0)

## 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)

## re-plot to see the result
plotTree(tree,mar=c(2.1,0.1,2.1,0.1))
axisPhylo()
mtext("c"),adj=0)

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

bind.tree.simmap	<i>Binds two trees of class "simmap"</i>
------------------	--

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

bmPlot

Simulates and visualizes continuous trait evolution on a phylogeny

Description

bmPlot conducts a 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.

sim.absorbing uses time discretization as well, but to conduct a simulation of either unbounded Brownian motion (if bounds=c(-Inf, Inf)), reflective bounded Brownian motion (if reflective=TRUE), or bounded Brownian motion with absorbing bounds.

sim.reflective is a synonym for sim.absorbing, but with the optional argument reflective=TRUE by default.

sim.multiTrend takes a "simmap" or "phylo" object as input and simulates trended Brownian motion on the tree, including for multiple trends depending on the mapped trait value.

Usage

```
bmPlot(tree, type="BM", anc=0, sig2=1/1000, ngen=1000, ...)
sim.absorbing(tree, x0=0, sig2=1, bounds=c(-Inf, Inf), nsteps=1000, ...)
sim.reflective(tree, x0=0, sig2=1, bounds=c(-Inf, Inf), nsteps=1000, ...)
sim.multiTrend(tree, x0=0, sig2=1, mu=0, ...)
```

Arguments

tree	a phylogenetic tree as an object of class "phylo" or "simmap".
type	the type of plot to create in bmPlot. See Description.
anc	the ancestral value for the root node in bmPlot.
sig2	the BM rate (variance of the stochastic diffusion process), σ^2 .
ngen	number of generations for the simulation: will rescale the tree to this total length in bmPlot.
x0	the ancestral value of the root node in sim.absorbing, sim.reflective, and sim.multiTrend.
bounds	absorbing or reflecting bounds in sim.absorbing. If bounds=c(-Inf, Inf) then the simulated process is unbounded.
nsteps	in sim.absorbing, the number of steps for the time discretization required for forward-in-time simulation.
mu	scalar or vector of mean trend for sim.multiTrend.
...	arguments to be passed to different methods.

Details

bmPlot 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 node 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.

Optional arguments in sim.absorbing include reflective, which defaults to reflective=FALSE. This function uses discretization to conduct a forward-in-time simulation, but does not rescale the tree. sim.reflective calls sim.absorbing internally, but with the optional argument reflective set to TRUE by default.

Value

bmPlot 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").

sim.absorbing, sim.reflective, and sim.multiTrend return named vectors of numeric values containing the tip states.

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.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[bounded_bm](#), [fastBM](#), [pbtree](#), [phenogram](#), [threshBayes](#)

Examples

```
## set seed
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))
```

Description

Fits a bounded Brownian motion model using the discretized diffusion approximation of Boucher & Demery (2016), or the "wrapped" or circular model of Martinet et al. (In revision).

Usage

```
bounded_bm(tree, x, lims=NULL, ...)
```

Arguments

tree	object of class "phylo".
x	continuous character data in a vector.
lims	bounds. If NULL will be set to three times the observed range of the trait. (This is designed to be an effectively "unbounded" null model.)
...	optional arguments.

Details

Optional arguments include (but are not limited to): `parallel`, defaults to `parallel=FALSE`; `df`, to adjust the degrees of freedom of the model, depending on whether or not the bounds have been set from the observed data; `levs`, number of levels for the discretization (defaults to `levs=200`); and `wrapped`, logical indicating whether to fit the "circular" or wrapped model, instead of the standard bounded model (defaults to `wrapped=FALSE`).

Value

An object of class "bounded_bm".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Boucher, F. C., and V. Demery (2016) Inferring bounded evolution in phenotypic characters from phylogenetic comparative data. *Systematic Biology*, **65**, 651-661.

Martinet, K. M., M. S. Juhn, F. C. Boucher, L. J. Harmon, L. J. Revell, S. I. A. Foerster, A. J. Shultz, K. Burns, M. E. Alfaro. (In revision) A wrapped Brownian motion model for traits on a circular scale.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[fitMk](#), [fitmultiBM](#), [fitThresh](#)

Examples

```
## Not run:
## load tree and data
data(eel.tree)
data(eel.data)

## extract trait of interval
maxTL<-setNames(eel.data$Max_TL_cm,
  rownames(eel.data))

## fit bounded model
eel_bounded<-bounded_bm(eel.tree,maxTL,levs=200,
  parallel=TRUE,lims=range(maxTL))

## fit unbounded model
eel_unbounded<-bounded_bm(eel.tree,maxTL,levs=200,
  parallel=TRUE)

## compare models using AIC
AIC(eel_bounded,eel_unbounded)
## End(Not run)
```

branching.diffusion *Animation of branching random diffusion*

Description

This function creates an animation of branching random diffusion (i.e., Brownian motion evolution 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.

Details

For animation to be recorded to file, the function requires the package *animation* as well as a video renderer.

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[bmPlot](#), [fastBM](#)

brownie.lite	<i>Likelihood test for rate variation in a continuous trait</i>
--------------	---

Description

Fits a multi-rate Brownian motion evolution model using maximum likelihood.

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

This function takes an object of class "phylo" or class "simmap" with a mapped binary or multi-state 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.

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 elements:

<code>sig2.single</code>	is the rate, σ^2 , for a single-rate model. This is usually the "null" model.
<code>a.single</code>	is the estimated state at the root node for the single rate model.
<code>var.single</code>	variance on the single rate estimator - obtained from the Hessian.
<code>logL1</code>	log-likelihood of the single-rate model.
<code>k1</code>	number of parameters in the single rate model (always 2).
<code>sig2.multiple</code>	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.
<code>a.multiple</code>	is the estimated state at the root node for the multi-rate model.
<code>var.multiple</code>	$p \times p$ variance-covariance matrix for the p rates - the square-roots of the diagonals should give the standard error for each rate.
<code>logL.multiple</code>	log-likelihood of the multi-rate model.
<code>k2</code>	number of parameters in the multi-rate model ($p+1$).
<code>P.chisq</code>	P-value for a likelihood ratio test against the χ^2 distribution; or
<code>P.sim</code>	P-value for a likelihood ratio test against a simulated null distribution.
<code>convergence</code>	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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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 fitted model
print(multiBM.fit)
```

brownieREML	<i>REML version of brownie.lite</i>
-------------	-------------------------------------

Description

Fits a multi-rate Brownian motion evolution model using REML. (See [brownie.lite](#) for more details.)

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. <code>names(x)</code> 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.

Details

This function takes an object of class "phylo" or an object of class "simmap" with a mapped binary or multi-state 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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[brownie.lite](#), [evol.vcv](#), [evol.rate.mcmc](#), [ratebytree](#)

Examples

```
## load data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)

## extract character of interest
gape.width<-setNames(sunfish.data$gape.width,
  rownames(sunfish.data))

## fit model
multiBM.reml<-brownieREML(sunfish.tree,
  gape.width)

## print fitted model
print(multiBM.reml)
```

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, ...)
cladebox(node, col="#0000FF40", ...)
```

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.
col	in the case of cladebox, the fill color for the plotted clade box.
...	optional arguments for arc.cladelabels or cladebox.

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.

cladebox draws a box over a clade in a plotted tree and should work for type="phylogram", type="fan", and type="arc" style trees. It returns the vertices of the plotted box (or polygon, in the case of a circular tree) invisibly.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[nodelabels](#)

collapse.to.star

Collapse a subtree to a star phylogeny

Description

Collapses a subtree to a star.

Usage

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

Arguments

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

Details

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.

Value

An object of class "phylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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 <i>mostly</i> match the arguments of plotSimmap , but also include the argument <code>drop.extinct=TRUE</code> 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 (or select Finish button in **RStudio**) to end.

Note, for the animation to work probably when run in the **RStudio** IDE the "zoom" level should be set to 100%.

Value

Returns the final plotted tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[plotTree](#), [plotSimmap](#)

Examples

```
## Not run:
## load phylogeny
data(liolaemid.tree)

## run collapseTree
pruned<-collapseTree(liolaemid.tree,fsize=0.5)

## reset par to default
par(mar=c(5.1,4.1,4.1,2.1))
## End(Not run)
```

compare.chronograms	<i>Compares two chronograms with precisely matching nodes in a visual manner</i>
---------------------	--

Description

Creates a visual comparison of two chronograms that differ in edge lengths but not topology.

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.

Details

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

Value

Function creates a plot.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

consensus.edges	<i>Compute consensus edges for a tree under some criterion</i>
-----------------	--

Description

Computes consensus edge lengths from a set of input trees using multiple 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 <code>nnls.tree</code> is used and the option <code>rooted</code> will be set to <code>is.rooted(tree)</code> for the consensus tree.
...	optional arguments, the most popular of which is <code>consensus.tree</code> - a user supplied consensus tree. Another optional argument for <code>method="mean.edge"</code> is <code>if.absent</code> 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>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

contMap

*Map continuous trait evolution on the tree***Description**

Projects the observed and reconstructed values of a continuous trait onto the edges of a tree using a color gradient.

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 determines the line width of the plotted error bars.

Details

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). This makes these interpolated values equal to the maximum likelihood estimates under a Brownian evolutionary process.

The default color palette is *not* recommended as it is not colorblind friendly and does not render well into gray scale; however, this can be updated using the helper function `setMap`.

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

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. (2013) Two new graphical methods for mapping trait evolution on phylogenies. *Methods in Ecology and Evolution*, **4**, 754-759.
- Revell, L. J. (2014) Graphical methods for visualizing comparative data on phylogenies. Chapter 4 in *Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice* (L. Z. Garamszegi ed.), pp. 77-103.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.
- Revell, L. J., K. Schliep, E. Valderrama, and J. E. Richardson (2018) Graphs in phylogenetic comparative analysis: Anscombe's quartet revisited. *Methods in Ecology and Evolution*, **9**, 2145-2154.

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","#FFFB2","#FECC5C","#FD8D3C",
    "#E31A1C"))

## plot object
plot(mammal.contMap,fsiz=c(0.7,0.8),
  leg.txt="log(body mass)")

## reset margins to default
par(mar=c(5.1,4.1,4.1,2.1))
```

cophylo

Creates a co-phylogenetic plot

Description

Visualize co-phylogenetic trees by multiple methods.

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".

<code>assoc</code>	matrix containing the tip labels in <code>tr1</code> to match to the tip labels in <code>tr2</code> . Note that not all labels in either tree need to be included; and, furthermore, one label in <code>tr1</code> can be matched with more than one label in <code>tr2</code> , or vice versa.
<code>rotate</code>	logical argument indicating whether nodes on both trees should be rotated to attempt to match in vertical position.
<code>x</code>	in the case of <code>plot.cophylo</code> , an object of class "cophylo" to be plotted.
<code>type</code>	for <code>cotangleplot</code> , the tree plotting style.
<code>use.edge.length</code>	for <code>cotangleplot</code> , a logical value indicating whether or not to plot trees with edge lengths.
<code>tangle</code>	for <code>cotangleplot</code> , whether to tangle the left tree, the right tree, or both.
<code>...</code>	optional arguments to be passed to <code>tipRotate</code> , or, in the case of <code>plot.cophylo</code> , to the internally used tree plotting function, <code>phylogram</code> . <code>phylogram</code> takes similar arguments to <code>plotSimmap</code> , such as <code>fsize</code> , <code>ftype</code> , <code>lwd</code> , and <code>pts</code> , though not all options from <code>plotSimmap</code> and <code>plotTree</code> are available. If <code>fsize</code> is supplied as a vector, different size fonts for the left & right facing trees may be used. In addition, the optional argument <code>scale.bar</code> , 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 <code>plot.cophylo</code> . If either tree contains polytomies, the <code>cophylo</code> argument <code>rotate.multi</code> should be set to <code>TRUE</code> . If curved linking lines are desired, the <code>plot.cophylo</code> argument <code>link.type</code> should be set to "curved". Other arguments for the plot method include <code>link.col</code> , <code>link.lty</code> , and <code>link.lwd</code> , 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 <code>assoc</code> . Finally, <code>edge.col</code> , 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 <i>rotated</i> trees, assuming that a rotation has been performed on <code>x</code> .

Details

`cophylo` 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.

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

`cotangleplot` creates a co-phylogenetic plot in which the edges of the matched trees are crossing and is designed to be used *only* on phylogenies with matching tip labels.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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))

## reset par to default
par(mar=c(5.1,4.1,4.1,2.1))
```

cospeciation	<i>Conducts a statistical test of cospeciation between two trees</i>
--------------	--

Description

Test for cospeciation based on tree distance.

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.

Details

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 whatsoever* between the two trees!

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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 results
print(wasp.cosp)

## plot null distribution & test statistic
```

```
plot(wasp.cosp)
title(main=paste("Simulated distribution of RF distances\n",
  "between unassociated trees"),font.main=3)
```

countSimmap	<i>Counts the number of character changes on a object of class "simmap" or "multiSimmap"</i>
-------------	--

Description

Count the number of changes of different types on a stochastically mapped trees or set of trees (objects of class "simmap" or "multiSimmap").

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.

Details

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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Examples

```
## load anole tree with ecomorphs mapped
data(anoletree)
anoletree

## countSimmap
countSimmap(anoletree)
```

ctt	<i>Generates (or simulates) a 'changes through time' plot from a set of stochastic map character histories</i>
-----	--

Description

Create a 'changes through time' plot from a "multiSimmap" object.

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.

Details

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.

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[ltt](#)

density.multiSimmap	<i>Computes a posterior distribution for the number and types of changes on the tree</i>
---------------------	--

Description

S3 methods to summarize the results 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

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% high probability density interval for the number of changes of each type on the tree. Otherwise, the central 95% of the posterior sample is returned as an estimate of the 95% 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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

densityMap	<i>Plot posterior density of stochastic mapping on a tree</i>
------------	---

Description

Visualize posterior probability density from stochastic mapping using a color gradient on the tree.

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.

Details

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.

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. Nielsen, and J. P. Bollback (2003) Stochastic mapping of morphological characters. *Systematic Biology*, **52**, 131-138.
- Revell, L. J. (2013) Two new graphical methods for mapping trait evolution on phylogenies. *Methods in Ecology and Evolution*, **4**, 754-759.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[make.simmap](#), [plotSimmap](#), [read.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 discrete character models
sunfish_er<-fitMk(sunfish.tree,fmode,model="ER")

## do stochastic mapping
smap.trees<-simmap(sunfish_er)

## compute "densityMap" object
sunfish.dmap<-densityMap(smap.trees,plot=FALSE,
  res=100)

## re-color
sunfish.dmap<-setMap(sunfish.dmap,hcl.colors(n=10))

## plot density map
plot(sunfish.dmap,lwd=5,outline=TRUE)

## reset margins to default
par(mar=c(5.1,4.1,4.1,2.1))
```

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

<code>trees</code>	an object of class "multiPhylo" or "multiSimmap".
<code>colors</code>	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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

describe.simmap	<i>Summarizes a stochastic mapped tree or set of trees</i>
-----------------	--

Description

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 containing 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

di2multi.simmap	<i>Collapse or resolve polytomies in a tree with a character painted on the edges</i>
-----------------	---

Description

Collapses or resolves polytomies in special types of phylogenetic trees. (See [di2multi](#) and [multi2di](#).)

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

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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[contMap](#), [densityMap](#), [di2multi](#), [make.simmap](#), [multi2di](#), [read.simmap](#)

dotTree	<i>Creates a phylogenetic dot plot</i>
---------	--

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 <i>matrix</i> , 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.

<code>min</code>	minimum value for <code>dot.legend</code> .
<code>max</code>	maximum value for <code>dot.legend</code> .
<code>Ntip</code>	number of tips in the plotted tree for <code>dot.legend</code> .
<code>length</code>	length of legend.
<code>prompt</code>	logical value indicating whether or not to prompt for legend position.
<code>...</code>	optional arguments. In the case of <code>dotTree</code> , these will be passed to plotTree or a different internally used plotting function for <code>method="phylogram"</code> . See phylo.heatmap for more detail on these arguments. Other option for <code>dotTree</code> also include <code>data.type</code> ("continuous" or "discrete"), <code>colors</code> , <code>length</code> , for data type "continuous" the length of the legend in terms of plotted circles, <code>x.space</code> , the spacing of the columns in the plotted data matrix, and <code>leg.space</code> , the spacing of the legend dots (again, for <code>data.type="continuous"</code> only).

Value

Function creates a plot.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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)

## reset margins to default
par(mar=c(5.1,4.1,4.1,2.1))
```

drop.clade	<i>Drop a clade from a tree</i>
------------	---------------------------------

Description

Internal function for `posterior.evolrate`.

Usage

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

Arguments

tree	object of class "phylo".
tip	set of tips in a clade.

Details

Function drops the clade containing the species in `tip`.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

drop.leaves	<i>Drop all the leaves (tips) from a tree</i>
-------------	---

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

drop.tip.contMap	<i>Drop tip or tips from an object of class "contMap" or "densityMap"</i>
------------------	---

Description

Drops one or multiple tips from an object of class "contMap" or "densityMap".

Usage

```
## S3 method for class 'contMap'
drop.tip(phy, tip, ...)
## S3 method for class 'densityMap'
drop.tip(phy, tip, ...)
## S3 method for class 'contMap'
keep.tip(phy, tip, ...)
## S3 method for class 'densityMap'
keep.tip(phy, tip, ...)
```

Arguments

phy 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

These functions are equivalent to [drop.tip](#) and [keep.tip](#) in the **ape** package, but for objects of class "contMap" and "densityMap".

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[contMap](#), [densityMap](#), [drop.tip](#), [drop.tip.simmap](#), [keep.tip](#)

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

Description

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

Usage

```
## S3 method for class 'multiSimmap'
drop.tip(phy, tip, ...)
## S3 method for class 'multiSimmap'
keep.tip(phy, tip, ...)
```

Arguments

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

Details

Equivalent to [drop.tip](#) and [keep.tip](#) in **ape**.
 This function merely wraps [drop.tip.simmap](#).

Value

An object of class "multiSimmap".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[drop.tip](#), [drop.tip.multiPhylo](#), [drop.tip.simmap](#), [keep.tip](#)

drop.tip.simmap

Drop tips or extract clade from tree with mapped discrete character

Description

This function drops one or multiple tips from an object of class "simmap".

Usage

```
## S3 method for class 'simmap'
drop.tip(phy, tip, ...)
## S3 method for class 'simmap'
keep.tip(phy, tip, ...)
extract.clade.simmap(tree, node)
```

Arguments

phy	an object of class "simmap".
tip	name or names of species to be dropped (or kept).
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.
tree	for <code>extract.clade.simmap</code> , an object of class "simmap".

Details

Equivalent to [drop.tip](#) and [keep.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.

Following [drop.tip](#) in **ape**, the returned tree is always in "cladewise" order.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[drop.tip](#), [extract.clade](#), [keep.tip](#), [make.simmap](#), [read.simmap](#), [sim.history](#)

Dtest	<i>Conducts correlational D-test from stochastic mapping</i>
-------	--

Description

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

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[make.simmap](#), [map.overlap](#)

edge.widthMap

Map continuous trait evolution on the tree

Description

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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)",
  color=palette()[4])

## reset margins to default
par(mar=c(5.1,4.1,4.1,2.1))
```

edgeProbs

Compute the relative frequencies of state changes along edges

Description

Computes the relative frequencies of character state changes along edges from a sample of stochastically mapped character histories.

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.
-------	---

Details

The function assumes that all trees in the sample differ only in their mapped histories & not at all in topology or branch lengths.

Note that edgeProbs 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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[plotSimmap](#)

estDiversity

Estimate diversity at each node of the tree

Description

Estimates the lineage density at each node in the tree based on a biogeographic model (similar to 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.

For 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 across 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

Estimates the phylogenetic location of a *single* rate shift on the tree using Bayesian MCMC (as described in Revell et al., 2012b).

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

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.

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

Value

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

mcmc	results from the MCMC run.
tips	list of tips 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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	<i>Likelihood test for variation in the evolutionary variance-covariance matrix</i>
----------	---

Description

Fits a multi-regime multivariate Brownian motion model following Revell & Collar (2009).

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 <code>err_vcv</code> which should contain a list of matrices of sampling <i>variances</i> 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

The function takes an object of class "simmap" with a mapped binary or multi-state 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*).

evol.vcv 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 elements:

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 matrix 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

See Also

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

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

Fits a multi-regime model for differences in the evolutionary variance-covariance structure or correlation between two continuous traits, following Revell & Collar (2009) and Revell et al. (2022).

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 $n \times m$ 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 <code>models</code> which specifies the models to be fit. See <i>Description</i> for more information. A second useful argument is <code>err_vcv</code> which should be supplied as a list of matrices of sampling <i>variances</i> 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 takes an object of class "simmap" with a mapped binary or multi-state 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.

Value

A list with the results summarized for each model.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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 (2022) A simple hierarchical model for heterogeneity in the evolutionary correlation on a phylogenetic tree. *PeerJ*, **10**, e13910.

See Also

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

Examples

```
## load data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)

## fit hierarchical common-structure models
sunfish.fit<-evolvcv.lite(sunfish.tree,
```

```

sunfish.data[,2:3],models="all models")

## print fitted models
print(sunfish.fit)

## compare models
anova(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 indicating which 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 *i*th tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

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

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

Modify the tip-spacing of a plotted tree.

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.

Details

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.

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 simply by calling a standard plotting function on the tree & tip spacings.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

Description

Exports trees & character data in XML format.

Usage

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

Arguments

<code>file</code>	filename for export.
<code>trees</code>	a phylogenetic tree or trees in "phylo" or "multiPhylo" format.
<code>X</code>	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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[make.simmap](#), [read.nexus](#), [read.simmap](#), [write.simmap](#)

fancyTree

*Plots special types of phylogenetic trees***Description**

Plots phylogenies (or phylogenetic trees and comparative data) in a variety of different styles.

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.

Details

This function plots a phylogeny or phylogenetic tree and comparative data in a variety of different styles, depending on the value of type. In some instances, fancyTree is now just a wrapper for other **phytools** functions, such as [contMap](#) and [densityMap](#).

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 extinct 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 "multiSimmap" object containing a binary [0,1] mapped character. (See [densityMap](#) for additional optional arguments if type="densitymap".)

If `type="contmap"`, reconstructed continuous trait evolution is mapped on the tree. Again, see [contMap](#) for additional arguments if `type="contmap"`.

If `type="phenogram95"` a 95% traitgram (aka. "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 `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 namespace) 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`).

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

- 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. (2013) Two new graphical methods for mapping trait evolution on phylogenies. *Methods in Ecology and Evolution*, **4**, 754-759.
- Revell, L. J. (2014) Graphical methods for visualizing comparative data on phylogenies. Chapter 4 in *Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice* (L. Z. Garamszegi ed.), pp. 77-103.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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
## load data from Revell & Collar (2009)
data(sunfish.tree)
data(sunfish.data)
fancyTree(sunfish.tree,type="traitgram3d",
          X=sunfish.data[,2:3],
          control=list(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)

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

## Not run:
## plot 95-percent CI phenogram
data(mammal.tree)
data(mammal.data)
bodyMass<-setNames(mammal.data$bodyMass,
                   rownames(mammal.data))
fancyTree(mammal.tree,type="phenogram95",x=bodyMass,
          fsize=0.7,ftype="i")
## End(Not run)

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

fastAnc

(Reasonably) fast estimation of ML ancestral states

Description

Estimates ancestral states for a continuous character under maximum likelihood.

Usage

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

Arguments

<code>tree</code>	an object of class "phylo".
<code>x</code>	a vector of tip values for species; <code>names(x)</code> should be the species names.
<code>vars</code>	a logical value indicating whether or not to compute variances on the ancestral state estimates. Variances are based on Equation (6) of Rohlf (2001).
<code>CI</code>	a logical value indicating whether or not to compute 95% confidence intervals on state estimates.
<code>...</code>	optional arguments. Presently this consists of <code>anc.states</code> , a named vector containing ancestral states to fix. Names should correspond to node numbers in the input tree.

Details

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% confidence intervals on the estimates.

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% confidence intervals (`CI95`).

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[ace](#), [anc.Bayes](#), [anc.ML](#), [ancr.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,printlen=10)
```

fastBM

(Reasonably) fast quantitative trait simulation on phylogenies

Description

Simulates one or multiple continuous traits on the tree under various evolutionary models.

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 $\mu \neq 0$.
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.

Details

This function conducts (reasonably) fast quantitative trait simulation on a phylogeny under several different models: Brownian motion (default), BM with a trend (for $\mu \neq 0$), bounds (for $\text{bounds} \neq c(-\text{Inf}, \text{Inf})$), and OU.

Value

A vector (for $\text{nsim}=1$) 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[sim.corrs](#)

Examples

```
## load example tree
data(anoletree)

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

## create a plot
phylo.heatmap(anoletree,X,fsiz=c(0.4,0.8,0.8),
  standardize=TRUE,colors=terrain.colors(n=20))
```

fastMRCA

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

Description

Finds the most recent common ancestor (MRCA) for a pair of tip 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

Function (`fastMRCA`) 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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

`getMRCA`, `findMRCA`, `mrca`

Examples

```
## simulate 2000-tip tree
tree<-pbtree(n=2000)

## find the MRCA of two randomly chosen tips
tips<-sample(tree$tip.label,2)
anc<-fastMRCA(tree,tips[1],tips[2])
```

findMRCA

Get the MRCA of a set of taxa

Description

Finds the most recent common ancestor (MRCA) of a set of tips.

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

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

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 namespace 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[findMRCA](#), [mrca](#)

Examples

```
## load tree
data(anoletree)

## find the common ancestor of three taxa
anc<-findMRCA(anoletree,c("cristatellus","cooki",
  "gundlachi"))

## graph the tree
plotTree(anoletree,type="fan",fsize=0.7,lwd=1)

## label the identified ancestor
nodelabels(node=anc,frame="circle",pch=21,cex=1.5,
  bg="blue")
```

```

legend("topleft",
      "most recent common ancestor\nof Puerto Rican TG anoles",
      pch=21,pt.cex=1.5,pt.bg="blue",cex=0.7,bty="n")

## reset margin to default
par(mar=c(5.1,4.1,4.1,2.1))

```

fit.bd	<i>Fits birth-death (speciation/extinction) model to reconstructed phylogeny</i>
--------	--

Description

Fits a birth-death (fit.bd) or pure-birth (fit.yule) model to a reconstructed phylogenetic tree with branch lengths.

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.

Details

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.

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.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

See Also

[birthdeath](#)

Examples

```
## load tree
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	<i>Evolutionary model fitting with intraspecific variability using Bayesian MCMC</i>
----------	--

Description

fitBayes uses Bayesian MCMC to sample terminal states (species means) as well as the parameters of an evolutionary model from their joint posterior distribution, following Revell & Reynolds (2012).

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 value 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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](#)

fitcontMk	<i>Fits binary discrete character evolution model where the transition rate varies as a function of a continuous trait</i>
-----------	--

Description

Fits a model where the rate of evolution of a binary trait depends on the value of a continuous character using the discretized diffusion approximation of Boucher & Demery (2016).

Usage

```
fitcontMk(tree, x, y, model="sigmoidal", ...)
```

Arguments

- | | |
|-------|--|
| tree | object of class "phylo". |
| x | binary discrete character, normally in the form of a factor or character vector. |
| y | continuous character data in a vector. |
| model | functional form describing the relationship between the value of the continuous trait and the binary discrete character transition rate. Presently the only option is model="sigmoidal". |
| ... | optional arguments. |

Details

fitcontMk uses the discretized diffusion approximation of Boucher & Demery (2016) to fit a model where the rate of evolution of a binary trait (in x) depends on the state of a continuous trait (in y), where the discrete and continuous character are modeled jointly. In some ways this is philosophically equivalent to the binary trait correlation method of Pagel (1994), but where one of the traits is continuously-valued.

Optional arguments include parallel (whether to distribute matrix exponentiation across nodes, which defaults to parallel=TRUE), levs (the number of levels for the discrete approximation, which defaults to levs=100), and plot_model (a logical indicating whether or not to graph the model while optimizing (which defaults to plot_model=FALSE)).

Value

An object of class "fitcontMk".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Boucher, F. C., and V. Demery (2016) Inferring bounded evolution in phenotypic characters from phylogenetic comparative data. *Systematic Biology*, **65**, 651-661.

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

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[bounded_bm](#), [brownie.light](#), [fitMk](#), [fitmultiBM](#), [fitPagel](#)

fitDiversityModel	<i>Fit diversity-dependent phenotypic evolution model</i>
-------------------	---

Description

Fits a diversity-dependent phenotypic evolution model (similar to 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 elements:

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

fitMk	<i>Fits extended Mk model for discrete character evolution</i>
-------	--

Description

The functions `fitMk`, `fitmultiMk`, `fitpolyMk`, `fitHRM`, `fitMk.parallel`, `fitgammaMk`, `fitfnMk`, and `mcmcMk` fit various flavors of the extended Mk model (Lewis, 2001) for discrete character evolution on a reconstructed phylogeny.

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, ...)
fitgammaMk(tree, x, model="ER", fixedQ=NULL, nrates=8, ...)
fitfnMk(tree, x, model="polynomial", degree=2, ...)

```

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 (or numeric matrix) of tip values for species; names(x) (or rownames(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. For fitfnMk the only option is presently model="polynomial".
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.
nrates	number of rate categories for discretized Γ distribution.

degree the degree of the polynomial for `fitfnMk`. (Defaults to `degree=2`.)

... 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`, both `order` (an evolutionary sequence for the monomorphic condition) and `max.poly` can be set for the `ordered=TRUE` model. If not set, `order` defaults to alphanumeric order, and `max.poly` defaults to the highest level of polymorphism observed in the data. (`max.poly` can also be specified for an unordered polymorphic trait model, and if not specified, defaults to the number of distinct monomorphic states of the data.)

Details

The function `fitMk` fits a so-called extended *Mk* 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 *Mk* 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 following Revell et al. (2024). 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 `fitgammaMk` fits an *Mk* model in which the edge rates are assumed to have been sampled randomly from a Γ distribution with mean of 1.0 and shape parameter α (Revell, May, and Harmon, In review).

The function `fitfnMk` fit an ordered *Mk* model in which the backward and forward transition rates between adjacent levels of the trait vary according to a functional form. Presently that function form is an *n*th degree polynomial, in which `degree` is set by the user (but defaults to `degree = 2`).

The function `fitpolyMk` fits an *Mk* 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 plus sign + (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 Mk 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 that 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 same 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 Mk 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`. To fit the same default hidden-rates model as is implemented in **corHMM**, one should set `corHMM_model=TRUE` and `ordered_hrm=FALSE`.

Note that (by default) both `fitMk` and `fitmultiMk` recycle code from **ace** in the **ape** package for computing the likelihood. (If the optional argument `lik.func="pruning"` then alternative, slightly faster, **phytools** code for the pruning algorithm is used.) `fitpolyMk`, `mcmcMk`, and `fitHRM` use `fitMk` internally to compute the likelihood.

Value

An object of class "fitMk", "fitmultiMk", "fitpolyMk", "mcmcMk", "fitHRM", "fitgammaMk", or "fitfnMk". 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Revell, L. J. and L. J. Harmon (2022) *Phylogenetic Comparative Methods in R*. Princeton University Press.

Revell, L. J., (M. May), and L. J. Harmon. In review. A discrete character evolution model for phylogenetic comparative biology with Γ -distributed rate heterogeneity among branches of the tree.

Revell, L. J., K. P. Schliep, D. L. Mahler, and T. Ingram (2024) Testing for heterogeneous rates of discrete character evolution on phylogenies. *Journal of Evolutionary Biology*, **37**, 1591-1602.

See Also

[ace](#), [make.simmap](#), [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)

## reset par to default
par(mfrow=c(1,1))

```

fitmultiBM

Fits multi-rate (or multi-trend) discrete character dependent continuous trait evolution

Description

Fits a multi-rate (or multi-trend) discrete-state-dependent Brownian motion model using the discretized diffusion approximation of Boucher & Demery (2016).

Usage

```

fitmultiBM(tree, x, y=NULL, model="ER", ncat=1, ...)
fitmultiTrend(tree, x, y=NULL, model="ER", ncat=1, ...)

```

Arguments

tree	object of class "phylo".
x	continuous character data in a vector.
y	optional discrete character, normally in the form of a factor or character vector.
model	discrete character evolution model. (E.g., "ER", "SYM", "ARD".)
ncat	number of rate categories (if hidden rates).
...	optional arguments.

Details

fitmultiBM uses the discretized diffusion approximation of Boucher & Demery (2016) to fit a model in which the rate of Brownian evolution of a continuous trait (in x) depends on the state of a discrete character (in y), where the discrete and continuous character are modeled jointly. If, in fact, the rates of evolution of x depend on the state of y , this will lead to more accurate model parameter estimates than when the discrete and continuous characters are separately modeled.

fitmultiTrend fits the same model, but in which the mean of the Brownian process, and not only its rate, vary as a function of the character y .

Optional arguments include `parallel` (whether to distribute matrix exponentiation across nodes, which defaults to `parallel=FALSE`), `levs` (the number of levels for the discrete approximation, which defaults to `levs=100`), `nullmodel` (a logical indicating whether or not to fit a null model without state-dependent continuous trait evolution, which defaults to `nullmodel=FALSE`), and `plot_model` (a logical indicating whether or not to graph the discretized model prior to fitting, which defaults to `plot_model=FALSE`).

Value

An object of class "fitmultiBM" or "fitmultiTrend".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Boucher, F. C., and V. Demery (2016) Inferring bounded evolution in phenotypic characters from phylogenetic comparative data. *Systematic Biology*, **65**, 651-661.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[bounded_bm](#), [brownie.lite](#), [fitMk](#)

Examples

```
## Not run:
## load tree and data
data(eel.tree)
data(eel.data)
feed_mode<-setNames(eel.data$feed_mode,
  rownames(eel.data))
lnTL<-setNames(log(eel.data$Max_TL_cm),
  rownames(eel.data))

## fit state-dependent model
eel_sdmodel<-fitmultiBM(eel.tree,lnTL,feed_mode,
  model="ARD",parallel=TRUE,levs=100,
  plot_model=TRUE)
```

```
## fit null model
eel_nullmodel<-fitmultiBM(eel.tree,lnTL,feed_mode,
  model="ARD",parallel=TRUE,levs=100,
  null_model=TRUE,plot_model=TRUE)

## fit null model with hidden-rates
eel_nullmodel.hrm<-fitmultiBM(eel.tree,lnTL,feed_mode,
  model="ARD",ncat=2,parallel=TRUE,levs=100,
  null_model=TRUE,plot_model=TRUE)

## fit state-dependent model with hidden-rates
eel_sdmodel.hrm<-fitmultiBM(eel.tree,lnTL,feed_mode,
  model="ARD",ncat=2,parallel=TRUE,levs=100,
  plot_model=TRUE)

## compare models
anova(eel_nullmodel,eel_sdmodel,eel_nullmodel.hrm,
  eel_sdmodel.hrm)
## End(Not run)
```

fitPagel

Function to test for correlated evolution of binary traits

Description

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

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" then the rate of substitution in x depends on y & vice versa. If dep.var="x" then the substitution rate in x depends on y, but not the converse. Finally, if dep.var="y" then 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.

Details

fitPagel fits both an independent evolution model, as well as Pagel's (1994) binary dependent model, and compares them with a likelihood-ratio test.

plot.fitPagel plots the fitted models using arrows.

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 of discrete characters. *Proceedings of the Royal Society B*, **255**, 37-45.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[ace](#), [fitMk](#), [make.simmap](#)

Examples

```
## Not run:
## load data
data(bonyfish.tree)
data(bonyfish.data)
```

```
## extract discrete characters
spawning_mode<-setNames(bonyfish.data$spawning_mode,
  rownames(bonyfish.data))
paternal_care<-setNames(bonyfish.data$paternal_care,
  rownames(bonyfish.data))

## fit correlational model
bonyfish.pagel<-fitPagel(bonyfish.tree,paternal_care,
  spawning_mode)

## test for a difference between models
anova(bonyfish.pagel)

## plot fitted models
plot(bonyfish.pagel,lwd.by.rate=TRUE)

## reset par
par(mar=c(5.1,4.1,4.1,2.1),mfrow=c(1,1))
## End(Not run)
```

fitThresh	<i>Fits binary, multi-state, and semi-threshold models using the discretized diffusion approximation</i>
-----------	--

Description

fitThresh uses the discretized diffusion approximation of Boucher and Demery (2016) to fit a binary or multi-state discrete character threshold model following Felsenstein (2012; Revell 2014).

fitsemiThresh uses the discretized diffusion approximation to fit a semi-threshold model in which liability is observed on some interval, but unobserved outside of it.

Usage

```
fitThresh(tree, x, sequence=NULL, ...)
fitsemiThresh(tree, x, threshold=c(0,1), ...)
```

Arguments

tree	object of class "phylo".
x	a named vector.
sequence	assumed ordering of the discrete character state. If not supplied and then alphanumeric order is assumed.
threshold	for fitsemiThresh, thresholds for the semi-threshold trait.
...	optional arguments. Optional arguments include levs (the number of levels for the discrete approximation, defaults to levs=200), root (a root prior assumption, defaults to root="fitzjohn" which involves treating the root like a nuisance parameter), and rand_start (defaults to rand_start=TRUE).

Value

An object of class "fitThresh" or "fitsemiThresh".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

- Boucher, F. C., and V. Demery (2016) Inferring bounded evolution in phenotypic characters from phylogenetic comparative data. *Systematic Biology*, **65**, 651-661.
- 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.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[ancThresh](#), [bounded_bm](#), [fitmultiBM](#), [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))

## fit model
sunfish_thresh<-fitThresh(sunfish.tree,fmode,
  levs=100)
sunfish_thresh

## fit ARD model for comparison
sunfish_ard<-fitMk(sunfish.tree,fmode,model="ARD")
sunfish_ard

## compare results
anova(sunfish_thresh,sunfish_ard)

## estimate ancestral states under threshold model
anc_sunfish<-ancr(sunfish_thresh)

## plot results
plot(anc_sunfish,mar=c(0.1,0.1,4.1,0.1),
```

```

args.plotTree=list(lwd=2,type="cladogram",
  nodes="inner",offset=1),
args.nodelabels=list(piecol=hcl.colors(n=2),
  cex=0.6),
args.tiplabels=list(cex=0.5),legend=FALSE)
legend("topleft",c("non-piscivorous","piscivorous"),
  pch=16,col=hcl.colors(n=2),pt.cex=2,bty="n")

## reset par to defaults
par(mar=c(5.1,4.1,4.1,2.1),cex.axis=1)
## End(Not run)

```

force.ultrametric	<i>Coerces a phylogenetic tree to be ultrametric</i>
-------------------	--

Description

Coerces an object of class "phylo" to be ultrametric.

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 <code>force.ultrametric</code> <i>should not</i> be used as a formal statistical method to ultrametricize a tree.

Details

`force.ultrametric` coerces a non-ultrametric tree 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.

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[is.ultrametric](#), [nnls.tree](#)

gammatest	γ 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[ltt](#), [mccr](#)

Examples

```
## load tree
data(darter.tree)

## run gammatest on "ltt" object
gammatest(ltt(darter.tree,plot=FALSE))
```

gamma_pruning	<i>Internally used function</i>
---------------	---------------------------------

Description

Internally used function to compute the likelihood under a Γ model.

Usage

```
gamma_pruning(par, nrates=4, tree, x, model=NULL, median=TRUE,
  expm.method="Higham08.b", ...)
```

Arguments

par	function parameters.
nrates	number of discrete rate categories.
tree	object of class "phylo".
x	data in the form of a binary matrix.
model	design matrix of fitted model.
median	use the median of each rate category.
expm.method	method used for matrix exponentiation.
...	optional arguments.

Value

A value of the log-likelihood.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

genSeq

Simulate a DNA alignment on the tree under a model

Description

Simulates DNA sequence on tree under the specified model.

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 ("a", "c", "g", and "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.

Details

Uses [sim.Mk](#) internally.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Examples

```
## load tree
data(mammal.tree)

## rescale edge lengths
mammal.tree$edge.length<-mammal.tree$edge.length/
  max(nodeHeights(mammal.tree))*0.2
```

```
## simulate gamma rate heterogeneity
gg<-rgamma(n=100,shape=0.25,rate=0.25)

## simulate DNA sequences with rate heterogeneity
dna<-genSeq(mammal.tree,l=100,rate=gg)
```

geo.legend	<i>Adds a geological (or other temporal) legend to a plotted tree</i>
------------	---

Description

Adds a geological legend to a plotted tree.

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.

Details

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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Examples

```
## load tree
data(anoletree)

## rescale tree to 50 ma total depth
anoletree<-rescale(anoletree,model="depth",depth=50)

## plot phylogeny
plotTree(anoletree,ylim=c(-0.16,1)*Ntip(anoletree),
         ftype="i",lwd=1,fsiz=0.5)

## add geological color legend
obj<-geo.legend()

## reset par to default
par(mar=c(5.1,4.1,4.1,2.1))
```

get.treepos	<i>Get position or node of a plotted tree interactively</i>
-------------	---

Description

Return the phylogenetic position of a mouse click on a plotted tree.

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 get.treepos and getnode are primarily meant for internal use in other **phytools** functions.

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.

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

getCladesofSize	<i>Get all subtrees larger than or equal to a specified size</i>
-----------------	--

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[extract.clade](#), [getDescendants](#)

getDescendants	<i>Get descendant node numbers</i>
----------------	------------------------------------

Description

Returns the descendants or parent of a specified node.

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.

Details

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).

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[Descendants](#), [paintSubTree](#)

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

Description

Computes the set of extant or extinct tips from a phylogenetic tree.

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.

Details

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. These tips are presumed to be "extant."
`getExtinct` returns the complement.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[nodeHeights](#)

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

Description

Computes the sister taxon or node.

Usage

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

Arguments

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

Details

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

Value

If `mode="number"` this function returns an integer or vector containing the node number or 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 or 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[getDescendants](#), [Siblings](#)

getStates	<i>Get the states at nodes or tips from a mapped tree</i>
-----------	---

Description

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[describe.simmap](#), [make.simmap](#), [read.simmap](#), [sim.history](#)

labelnodes	<i>Function to interactively label nodes of a plotted tree</i>
------------	--

Description

Adds node labels to a plotted object of class "phylo".

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.

Details

The nodes to be labels can be selected interactively by the user (i.e., by clicking on the corresponding nodes of the plotted tree).

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[cladelabels](#), [nodelabels](#)

ladderize.simmap

Ladderize a tree with a mapped discrete character

Description

Ladderizes an object of class "simmap".

Usage

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

Arguments

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

Details

This function 'ladderizes' an object of class "simmap" with a mapped discrete character.

For more information see [ladderize](#).

Value

A ladderized object of class "simmap".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[make.simmap](#), [ladderize](#)

Examples

```
## load "simmap" object
data(anoletree)

## ladderize object
anoletree_ladderized<-ladderize.simmap(anoletree)

## plot unladderized & ladderized
par(mfrow=c(2,1))
plot(anoletree,ftype="off",direction="upwards",
     mar=c(1.1,2.1,2.1,1.1))
mtext("a) unladderized tree",adj=0)
plot(anoletree_ladderized,ftype="off",
     direction="upwards",mar=c(1.1,2.1,2.1,1.1))
mtext("b) ladderized tree",adj=0)

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

lambda.transform

λ transformation of matrix

Description

Internal function for [phyl.pca](#) and others.

Usage

lambda.transform(lambda, C)

Arguments

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

Details

Multiplies the off-diagonals of a square matrix by lambda and returns the result.

Value

Typically an among-species phylogenetic variance covariance matrix (e.g., [vcv.phylo](#)) in which the off-diagonal elements have been multiplied by lambda.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

likMlambda	<i>Likelihood for joint λ</i>
------------	--

Description

Likelihood function for joint estimation 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 $n \times n$ 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

linklabels	<i>Function to add tip labels to a plotted tree with linking lines</i>
------------	--

Description

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[cladelabels](#), [nodelabels](#), [tiplabels](#)

locate.fossil

Locate a fossil lineage in a tree using continuous characters

Description

Uses ML to place a fossil lineage into a tree using continuous traits following Revell et al. (2015).

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

Uses ML (or REML) to place a recently extinct, cryptic, or missing taxon on an ultrametric (i.e., time-calibrated) phylogeny following Revell et al. (2015).

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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.

Usage

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

Arguments

tree	phylogeny.
D	distance matrix.

Details

Generally intended as a function to be used internally by [optim.phylo.ls](#).

Value

A tree with branch lengths.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

Description

Computes and visualizes a lineage through time (LTT) plot, and related measures.

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

<code>tree</code>	an object of class "phylo", "multiPhylo", "simmap", or "multiSimmap".
<code>plot</code>	a logical value indicating whether or not to create LTT plot.
<code>drop.extinct</code>	logical value indicating whether or not to drop extinct tips from the tree.
<code>log.lineages</code>	logical value indicating whether LTT plot should be on log-linear (default for "phylo" and "multiPhylo" objects) or linear-linear scale.
<code>gamma</code>	logical value indicating whether or not to compute γ from Pybus & Harvey (2000; <i>Proc. Roy. Soc. B</i>).
<code>n</code>	for gtt the number of time intervals to use to track γ through time.
<code>obj</code>	for mccr an object of class "ltt".
<code>rho</code>	for mccr sampling fraction.
<code>nsim</code>	for mccr number of simulations to use for the MCCR test.
<code>...</code>	other arguments to be passed to plotting methods. See plot.default .

Details

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](#).

For the case in which `tree` is an object of class "simmap" or "multiSimmap" then the object will contain the number of lineages through time (for each tree, in the case of "multiSimmap" objects) separated by mapped regimes.

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.

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 normally includes the following elements:

times	a vector of branching times.
ltt	a vector of lineages, or a matrix of lineages in each state over time for objects of class "simmap" and "multiSimmap".
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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[gammatest](#), [ltt95](#)

Examples

```
## set seed
set.seed(99)

## simulate set of random trees
trees<-pbtree(n=100,scale=100,nsim=10)

## create "multiLTT" object
obj<-ltt(trees,plot=FALSE)

## plot "multiLTT" object
plot(obj,log="y",log.lineages=FALSE,
      bty="l")
title(main="LTT plots for 10 pure-birth trees",
      font.main=3)

## simulate single tree
tree<-pbtree(b=1,d=0.25,t=4)
```

```
## graph LTT with superimposed tree
obj<-ltt(tree,gamma=FALSE,show.tree=TRUE,
        bty="l",lwd=2)
title(main="LTT plot with superimposed tree",
      font.main=3)
obj

## load dataset
data(anoletree)

## create gamma-through-time plot
anole.gtt<-gtt(anoletree,n=40)
plot(anole.gtt)
```

ltt95

Creates a $(1-\alpha)\%$ CI for a set of LTTs

Description

This function computes LTT plots for a set of trees & plots a $(1-\alpha)\%$ CI by various methods. (See [ltt](#) for more details.)

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 <i>all</i> 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 $\text{res}=100$); 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 $(1-\alpha)\%$ 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[ltt](#)

make.era.map

Create "era" map on a phylogenetic tree

Description

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

Examples

```
## simulate tree with total depth of 100
tree<-pbtree(n=1000,scale=100)

## subdivide into mapped eras
tree<-make.era.map(tree,c(0,25,50,75))

## create graph of mapped tree
plot(tree,ftype="off",lwd=1)

## reset margins to default
par(mar=c(5.1,4.1,4.1,2.1))
```

make.simmap	<i>Simulate stochastic character maps on a phylogenetic tree or trees</i>
-------------	---

Description

Performs stochastic character mapping (Huelsenbeck et al., 2003) using several different alternative methods.

Usage

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

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 <i>per</i> input tree.

...	optional arguments. So far, <code>pi</code> gives the prior distribution on the root node of the tree. Acceptable values for <code>pi</code> are "equal", "estimated", or a vector with the frequencies. If <code>pi="estimated"</code> then the stationary distribution is estimated by numerically solving $\pi \cdot Q = 0$ for π, and this is used as a prior on the root. If <code>pi="fitzjohn"</code>, then the Fitzjohn et al. (2009) root prior is used. Finally, if <code>pi</code> is a numeric vector then the root state will be sampled from this vector. The function defaults to <code>pi="equal"</code> which results in the root node being sampled from the conditional scaled likelihood distribution at the root. <code>message</code> tells whether or not to print a message containing the rate matrix, Q and state frequencies. <code>message</code> defaults to TRUE. For optional argument <code>Q="mcmc"</code> (see below) the mean value of Q from the posterior sample is printed. <code>tol</code> gives the tolerance for zero elements in Q. (Elements less than <code>tol</code> will be reset to <code>tol</code>). Optional argument <code>Q</code> can be a string ("empirical" or "mcmc"), or a fixed value of the transition matrix, Q. If "empirical" then a single value of Q, the most likely value, is used for all simulations. If "mcmc", then <code>nsim</code> 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 <code>vQ</code> can consist of a single numeric value or a vector containing the variances of the (normal) proposal distributions for the MCMC. The order of <code>vQ</code> is assumed to be in the order of the <code>index.matrix</code> in <code>ace</code> for the chosen model. <code>prior</code> 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 <code>vQ</code>, the order of <code>prior</code> is assumed to correspond with the order of <code>index.matrix</code> as in <code>ace</code>. <code>prior</code> can also be given the optional logical value <code>use.empirical</code> which tells the function whether or not to give the prior distribution the empirical mean for Q. If TRUE then only <code>prior\$beta</code> is used and <code>prior\$alpha</code> is set equal to <code>prior\$beta</code> times the empirical mean of Q. <code>burnin</code> and <code>samplefreq</code> are burn-in and sample frequency for the MCMC, respectively.
object	for generic <code>simmap</code> method, object of various classes: for instance, an object of class "fitMk" from <code>fitMk</code> .

Details

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.

`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** $\geq 0.2-33$ `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** $\leq 0.2-26$. 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:

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

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. (2024) `phytools 2.0`: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.
- Revell, L. J. and L. J. Harmon (2022) *Phylogenetic Comparative Methods in R*. Princeton University Press.

See Also

[ancr](#), [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))

## fit model
er_model<-fitMk(sunfish.tree,fmode,model="ER",
  pi="fitzjohn")

## do stochastic mapping
sunfish_smap<-simmap(er_model)

## print a summary of the stochastic mapping
summary(sunfish_smap)

## plot a posterior probabilities of ancestral states
cols<-setNames(c("blue","red"),levels(fmode))
plot(summary(sunfish_smap),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),las=1)

## plot posterior density on the number of changes
plot(density(sunfish_smap),bty="l")
title(main="Posterior distribution of changes of each type",
  font.main=3)
## reset par to default
par(las=0)
## End(Not run)
```

map.overlap

Proportional overlap between two mapped character histories on a tree

Description

Calculates the similarity of two different stochastically mapped character histories.

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.

Details

map.overlap computes a single quantity giving the overall similarity of the maps, consequently this measure only makes sense if some or all of the states are shared between the two mapped trees.

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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

map.to.singleton	<i>Converts a tree without singletons to a tree with singleton nodes</i>
------------------	--

Description

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.

Usage

```
map.to.singleton(tree)
plotTree.singletons(tree)
## S3 method for class 'singleton'
drop.tip(phy, 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).
phy	for drop.tip.singleton, an object of class "singleton" or "phylo".
tip	for drop.tip.singleton, a tip label or vector of tip labels.
...	optional arguments for drop.tip.singleton.

Details

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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[collapse.singles](#), [drop.tip](#), [make.simmap](#)

<code>mapped.states</code>	<i>Returns a vector, matrix, or list of the mapped states on a tree or set of trees</i>
----------------------------	---

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

<code>tree</code>	a single tree or a set of trees as an object of class "simmap" or "multiSimmap", respectively.
<code>...</code>	optional arguments.

Value

A vector, matrix, or list.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

markChanges	<i>Add marked changes to a plotted tree with mapped discrete character</i>
-------------	--

Description

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[plotSimmap](#)

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 model
er_model<-fitMk(sunfish.tree,fmode,model="ER",
  pi="fitzjohn")

## generate single stochastic map
sunfish_smap<-simmap(er_model,nsim=1)

## plot stochastic map & mark changes
cols<-setNames(c("blue","red"),levels(fmode))
plot(sunfish_smap,cols,ftype="i")
markChanges(sunfish_smap,colors=cols,lwd=6)

## reset par to default
par(mar=c(5.1,4.1,4.1,2.1))
```

matchNodes	<i>Matches nodes between two trees</i>
------------	--

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.

Usage

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

Arguments

<code>tr1</code>	first tree.
<code>tr2</code>	second tree.
<code>method</code>	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 unambiguous shortening of "descendants" or "distances" is also permitted.
<code>...</code>	optional arguments which may or may not be used depending on the value of <code>method</code> . <code>tol</code> is a tolerance value for the difference from exact matching that is allowed for <code>method="distances"</code> . <code>corr</code> , which is <code>FALSE</code> by default, indicates whether to match nodes under <code>method="distances"</code> using the correlation (<code>corr=TRUE</code>) or the absolute similarity of distances.

Details

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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

mergeMappedStates	<i>Merge two or more mapped states into one state</i>
-------------------	---

Description

Merges two or mapped states on the tree to get one new state.

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.

Details

mergeMappedStates can be used to merge two or more mapped states into a single, new state. For instance, one could merge the states "C", "G", and "T" and define the new the state "not-A".

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

midpoint_root	<i>Midpoint root a phylogeny</i>
---------------	----------------------------------

Description

These functions midpoint root a rooted or unrooted tree (Farris 1972).

Usage

```
midpoint_root(tree)
## S3 method for class 'root'
midpoint(tree, node.labels="support", ...)
```

Arguments

tree	an object of class "phylo".
node.labels	argument to be passed to midpoint .
...	optional arguments to be passed to midpoint .

Details

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

The function `midpoint_root` performs the same operation as [midpoint](#) in the **phangorn** package, but uses no **phangorn** (Schliep, 2011) code internally.

The function `midpoint.root` is a pseudo S3 method for the object class "root" that exists because when [midpoint](#) was created in **phangorn** it was not defined as a generic method. This function merely points to [midpoint](#) and is being deprecated out.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

See Also

[midpoint](#), [reroot](#), [root](#)

minRotate	<i>Rotates all nodes of the tree to minimize the difference in order with a vector</i>
-----------	--

Description

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 minRotate and tipRotate 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

minSplit	<i>Finding the minimum (median) split in the posterior sample</i>
----------	---

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 elements:

- 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

Computes modified Grafen (1989) edge lengths.

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.

Details

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.

Value

An object of class "phylo" with edge lengths.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

- Grafen, A. (1989) The phylogenetic regression. *Philosophical Transactions of the Royal Society of London. Series B. Biological Sciences*, **326**, 119-157.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[compute.brLen](#)

mrp.supertree	<i>Matrix representation parsimony supertree estimation</i>
---------------	---

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 <code>compute.mr</code> , 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

See Also

[exhaustiveMP](#), [optim.parsimony](#), [pratchet](#)

multi.mantel	<i>Multiple matrix regression (partial Mantel test)</i>
--------------	---

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	the 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

Description

Computes phylogenetic covariance matrices from a "simmap" object.

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.

Details

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.
Used internally by multiple **phytools** functions, such as [brownie.lite](#).

Value

A list of matrices.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

multirateBM	<i>Function to fit a multi-rate Brownian evolution model</i>
-------------	--

Description

Fits a flexible multi-rate Brownian motion evolution model using penalized likelihood.

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.

Details

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.

Value

An object of class "multirateBM".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

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

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

Examples

```
## Not run:
## load data
data(sunfish.tree)
data(sunfish.data)

## convert from "simmap" to "phylo"
sunfish.tree<-as.phylo(sunfish.tree)

## 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)

## reset par
par(mar=c(5.1,4.1,4.1,2.1))
## 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 argument 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

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

nodeHeights	<i>Compute the heights above the root of each node</i>
-------------	--

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 highest 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, whereas 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[vcvPhylo](#)

Examples

```
## load tree
data(vertebrate.tree)

## compute height of all nodes
H<-nodeHeights(vertebrate.tree)
print(H)

## compute total tree depth
max(H)
```

nodelabels.cophylo	<i>Add labels to a plotted "cophylo" object</i>
--------------------	---

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[cophylo](#), [edgelabels](#), [nodelabels](#), [tiplabels](#)

optim.phylo.ls	<i>Phylogeny inference using the least squares method</i>
----------------	---

Description

Phylogenetic inference using the method of least-squares (Cavalli-Sforza & Edwards, 1967).

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("Q-score")`.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Cavalli-Sforza, L. L., and A. W. F. Edwards. (1967) Phylogenetic analysis: Models 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

See Also

[exhaustiveMP](#), [nni](#)

orderMappedEdge	<i>Order the columns of mapped.edge to match across trees</i>
-----------------	---

Description

Orders the levels of a mapped character to match across trees in a "multiSimmap" object.

Usage

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

Arguments

- | | |
|----------|---|
| trees | object of class "simmap" or "multiSimmap". |
| 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")). |

Details

This function takes a an object of class "multiSimmap" with a mapped discrete character (e.g., see [make.simmap](#) and sorts the columns of each tree\$mapped.edge element 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.

Value

An object of class "simmap" or (normally) "multiSimmap".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

paintSubTree	<i>Paint sub-trees with a discrete character</i>
--------------	--

Description

Paints regimes on a tree to create an object of class "simmap" with mapped regimes.

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.

Details

These functions map or "paint" arbitrary (i.e., user-specified) discrete character histories on the tree. paintSubTree paints the clade downstream of node with a particular state; whereas paintBranches paints only a specified branch.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

paste.tree	<i>Paste two trees together</i>
------------	---------------------------------

Description

Internal function for [posterior.evolrate](#).

Usage

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

Arguments

tr1	receptor tree.
tr2	donor clade.

Details

Primarily designed as an internal function for [posterior.evolrate](#); however, can be used to graft a clade onto a receptor tree at the "sticky tip" labeled with "NA".

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Examples

```
## create two random trees
tr1<-rtree(10)
tr2<-rtree(10)

## make one tip "sticky" by renaming it "NA"
tr1$tip.label[sample(1:10,1)]<-"NA"
tr2$root.edge<-0

## paste the two trees together
tr3<-paste.tree(tr1,tr2)

## plot the results
layout(matrix(c(1,1,2,2,4,3,3,4),2,4,byrow=TRUE))
plotTree(tr1,direction="upwards",
  mar=c(1.1,1.1,3.1,1.1))
add.arrow(tree=tr1,tip=which(tr1$tip.label=="NA"),
  hedl=0.2,arrr=0.2,offset=1)
mtext("a) recipient tree",adj=0)
plotTree(tr2,direction="upwards",plot=FALSE)
pp<-get("last_plot.phylo",envir=.PlotPhyloEnv)
plotTree(tr2,direction="upwards",
  mar=c(1.1,1.1,3.1,1.1),
  ylim=c(-0.2,pp$y.lim[2]),add=TRUE)
arrows(pp$xx[11],pp$yy[11],pp$xx[11],-0.2,
  length=0.1,lwd=2)
mtext("b) donor tree",adj=0)
plotTree(tr3,direction="upwards",
  mar=c(1.1,1.1,3.1,1.1))
mtext("c) grafted tree",adj=0)

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

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 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 continuous-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

<i>b</i>	birth rate or speciation rate for <code>type="continuous"</code> ; the probability of speciating per time-step for <code>type="discrete"</code> .
<i>d</i>	death rate or extinction rate for <code>type="continuous"</code> ; the probability of going extinct per time-step for <code>type="discrete"</code> .
<i>n</i>	desired number of species (i.e., taxa-stop criterion).
<i>t</i>	total time for simulation (i.e., time-stop criterion).
<i>scale</i>	if set, rescales tree to have total length scale.
<i>nsim</i>	number of simulated trees to return.
<i>type</i>	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 <code>ape</code> , a logical value indicating whether to return nodes in a 'ape' compatible ordering (default is TRUE); <code>extant.only</code> a logical value indicating whether or not to return only extant species (defaults to FALSE); <code>max.count</code> a numeric value indicating the maximum number of iterations to run is sampling conditioned on both <i>n</i> and <i>t</i> (defaults to 1e5); <code>method</code> gives the method used for simultaneously conditioning on <i>n</i> and <i>t</i> - options are "rejection" and "direct"; <code>tip.label</code> , 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`).

Details

Simulate stochastic birth-death trees.

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

Phylogenetic regression with within-species sampling error following Ives et al. (2007).

Usage

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

pgls.Ives 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.

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 elements:

<code>beta</code>	a vector or matrix of regression coefficients.
<code>sig2x</code>	fitted BM rate for X.
<code>sig2y</code>	fitted BM rate for y.
<code>a</code>	fitted ancestral states for X and y.
<code>logL</code>	log-likelihood.
<code>convergence</code>	a value for convergence. <code>convergence=0</code> is good; see optim for more details.
<code>message</code>	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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

phenogram	<i>Plot traitgram (phenogram)</i>
-----------	-----------------------------------

Description

Plots a phylogenetic traitgram (Evans et al., 2009).

Usage

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

Arguments

<code>tree</code>	an object of class "phylo", with or without a mapped discrete character.
<code>x</code>	a vector containing the states at the tips <i>or</i> the states at all the tips and the internal nodes of the tree.
<code>fsize</code>	relative font size for tip labels.
<code>ftype</code>	font type - options are "reg", "i" (italics), "b" (bold), or "bi" (bold-italics).
<code>colors</code>	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.
<code>axes</code>	list of axis dimensions. Items are time and trait.
<code>add</code>	optional logical value indicating whether to add to an open plot. If TRUE, then new axes will not be plotted.
<code>...</code>	optional arguments including <code>xlim</code> , <code>ylim</code> , <code>log</code> , <code>main</code> , <code>sub</code> , <code>xlab</code> , <code>ylab</code> , <code>asp</code> , <code>type</code> , <code>lty</code> , <code>lwd</code> , <code>offset</code> , and <code>digits</code> are as in <code>plot.default</code> or <code>par</code> . Note that <code>axes</code> overrides <code>xlim</code> and <code>ylim</code> . <code>spread.labels</code> is a logical value indicating whether or not to minimize tip label overlap (default is TRUE); <code>spread.cost</code> is a numeric vector indicating the relative penalty to be used for label overlap and deviance, respectively (if <code>spread.labels=TRUE</code>); <code>spread.range</code> 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, <code>link</code> is a numeric value by which to offset the tip labels, linking them to the tips with a dashed line (default is 0, if <code>spread.labels=FALSE</code> , or 10% of the total tree length otherwise). The optional argument <code>offsetFudge</code> "fudges" the computation of label offset in scaling <code>xlim</code> . It is 1.37, which is the correct fudge in the Windows R GUI, but this may need to be changed in other systems. <code>hold</code> indicates whether (or not) the output to the graphical device should be held using <code>dev.hold</code> before plotting (defaults to <code>hold=TRUE</code>). <code>quiet</code> suppresses some system messages if set to <code>quiet=TRUE</code> .

Details

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.

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

Revell, L. J. (2014) Graphical methods for visualizing comparative data on phylogenies. Chapter 4 in *Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice* (L. Z. Garamszegi ed.), pp. 77-103.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Revell, L. J., K. Schliep, E. Valderrama, and J. E. Richardson (2018) Graphs in phylogenetic comparative analysis: Anscombe’s quartet revisited. *Methods in Ecology and Evolution*, **9**, 2145-2154.

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,color=palette()[4],
  xlab="time (ma)",ylab="log(body mass)",las=1)
```

phyl.cca	<i>Phylogenetic canonical correlation analysis</i>
----------	--

Description

Phylogenetic canonical correlation analysis, following Revell & Harrison (2008).

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 elements:

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>i</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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[phyl.pca](#)

phyl.pairedttest	<i>Phylogenetic paired t-test</i>
------------------	-----------------------------------

Description

Phylogenetic paired *t*-test following Lindenfors et al. (2010).

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

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

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.

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 elements:

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 $((\text{dbar}-h0)/\text{se})$ 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Description

This function performs phylogenetic PCA following Revell (2009).

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, ...)
as.prcomp(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, as.princomp, and as.prcomp 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".

S3 methods as.prcomp and as.princomp convert the object of class "phyl.pca" to objects of class "prcomp" and "princomp", respectively.

Value

An object of class "phyl.pca" consisting of a list with some or all of the following elements:

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[phyl.cca](#), [phyl.resid](#), [prcomp](#), [princomp](#)

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

Computes the residuals from the phylogenetic regression of multiple y variables (in a matrix) on a single x.

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

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).

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 elements:

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

phyl.RMA	<i>Phylogenetic reduced major axis (RMA) regression</i>
----------	---

Description

Phylogenetic reduced major axis (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).

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.

Note that some statistician think there is *never* a condition in which a reduced-major-axis regression is appropriate.

Value

An object of class "phyl.RMA" consisting of a list with the following elements:

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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)

## plot fitted RMA
par(las=1,bty="n")
plot(fitted.rma,las=1,bty="n")
```

phyl.vcv

Compute evolutionary VCV matrix for a tree & dataset

Description

Internal function for [phyl.pca](#).

Usage

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

Arguments

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

Details

Primarily designed as an internal function for [phyl.pca](#); `phyl.vcv` 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`).

Note that prior to **phytools** 2.1-0 the matrix X was not sorted to match the rows of C since that was assumed to have been done in the function calling `phyl.vcv` internally; however, I recently discovered that this had caused the function to be used incorrectly resulting in a paper correction. This is now fixed such that X is checked for row names and (if present) C is sorted to match the rows of X. Hopefully this does not cause any problems for other functions using `phyl.vcv`!

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

phylANOVA

Phylogenetic ANOVA and post-hoc tests

Description

Simulation based phylogenetic ANOVA following Garland et al. (1993), with post-hoc tests.

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 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).

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

phylo.heatmap	<i>Creates a phylogenetic heat map</i>
---------------	--

Description

Multivariate phylogenetic [heatmap](#) plot.

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., <code>heat.colors(n=200)[200:1]</code>).
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: <code>legend</code> , a logical value indicating whether or not to plot a figure legend (defaults to <code>legend=TRUE</code>); <code>labels</code> , a logical value indicating whether or not to plot trait labels (defaults to <code>labels=TRUE</code>); <code>split</code> , a numeric vector indicating the fraction of the horizontal dimension to use for the tree & heatmap, respectively (defaults to <code>split=c(0.5,0.5)</code>); <code>xlim</code> , <code>ylim</code> , & <code>mar</code> , defined as in par ; and <code>ftype</code> , <code>lwd</code> , and <code>pts</code> as defined in plotSimmap .

Value

Function creates a plot.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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,pts=FALSE)

## reset par to default
par(mar=c(5.1,4.1,4.1,2.1))
```

phylo.impute

Phylogenetic imputation for multivariate continuous character data

Description

This function performs phylogenetic multiple 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 occurred 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 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

`evol.vcv`

phylo.to.map

Plot tree with tips linked to geographic coordinates

Description

Project a phylogeny 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

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

Details

`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.

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. (2014) Graphical methods for visualizing comparative data on phylogenies. Chapter 4 in *Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice* (L. Z. Garamszegi ed.), pp. 77-103.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Examples

```
## load data
data(tortoise.tree)
data(tortoise.geog)

## create "phylo.to.map" object
tortoise.phymap<-phylo.to.map(tortoise.tree,
  tortoise.geog,plot=FALSE,direction="rightwards",
  regions="Ecuador")

## plot object
plot(tortoise.phymap,direction="rightwards",pts=FALSE,
  xlim=c(-92.25,-89.25),ylim=c(-1.8,0.75),ftype="i",
  fsize=0.8,lty="dashed",map.bg="lightgreen",
  colors="slategrey")

## reset margins
par(mar=c(5.1,4.1,4.1,2.1))

## note: generally recommend using higher resolution map
## e.g., from mapdata package
```

phylo.toBackbone	<i>Converts tree to backbone or vice versa</i>
------------------	--

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[plot.backbonePhylo](#)

phyloDesign	<i>Compute design matrix for least squares analyses</i>
-------------	---

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

phylomorphospace	<i>Creates phylomorphospace plot</i>
------------------	--------------------------------------

Description

Project a phylogeny into morphospace.

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 traits 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 <i>vice versa</i> .
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 phylomorphospace to the current plot.

Details

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.

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. (2014) Graphical methods for visualizing comparative data on phylogenies. Chapter 4 in *Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice* (L. Z. Garamszegi ed.), pp. 77-103.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.
- 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))
```

```
## graph phylomorphospace
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	<i>Creates three-dimensional phylomorphospace plot</i>
--------------------	--

Description

Creates a phylomorphospace plot in three dimensions.

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 traits 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.

Details

This function creates a phylomorphospace plot for three characters using the 3D visualization package, **rgl** (if available) or statically, by simulating 3D on a flat graphical device.

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. (2014) Graphical methods for visualizing comparative data on phylogenies. Chapter 4 in *Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice* (L. Z. Garamszegi ed.), pp. 77-103.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

```
## load data
data(anoletree)
data(anole.data)

## run phylogenetic PCA
anole.pca<-phyl.pca(anoletree,anole.data)

## Not run:
## create dynamic plot using rgl
phylomorphospace3d(anoletree,scores(anole.pca)[,1:3],
  control=list(spin=FALSE))
## End(Not run)

## create static plot
par(cex=0.5) ## adjust font size
phylomorphospace3d(anoletree,scores(anole.pca)[,1:3],
  method="static",angle=-30)
par(cex=1) ## reset to default
```

phylosig

*Compute phylogenetic signal with two methods***Description**

Calculate phylogenetic signal using two different methods (Pagel, 1999; Blomberg et al., 2003).

Usage

```
phylosig(tree, x, method="K", test=FALSE, nsim=1000, se=NULL, start=NULL,
         control=list(), niter=10)
## 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.
niter	number of iterations for likelihood optimization of λ (if se!=NULL), or the number of <i>intervals</i> between 0 and the maximum possible value of λ for univariate optimization of λ (if se=NULL).
...	optional arguments for plot method.

Details

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).

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 elements:

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 elements:

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., and 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, and 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

## here's what trans looks like
print(trans)

## convert
obj<-phylo.toBackbone(tree,trans)

## plot
plot(obj)

## reset par to default
par(mar=c(5.1,4.1,4.1,2.1))
```

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.

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 then the second element of cex will be passed internally to the function add.color.bar .

Details

Unlike most other tree plotting functions in **phytools**, this function calls [plot.phylo](#) (not plotSimmap) internally.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

plotSimmap	<i>Plot stochastic character mapped tree</i>
------------	--

Description

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=if(type=="arc") 0.5 else 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, underscore=FALSE, arc_height=2)
## S3 method for class 'simmap'
plot(x, ...)
## S3 method for class 'multiSimmap'
plot(x, ...)
```

Arguments

tree	an object of class "simmap" or "multiSimmap" 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 .)
underscore	logical value indicating whether to plot the underscore character, "_", or substitute for a space. Defaults to underscore=FALSE.
arc_height	for type="arc" trees, the height of the arc in units of total tree depth.
x	for S3 plotting method, object of class "simmap" or "multiSimmap".
...	for S3 plotting method, other arguments to be passed to plotSimmap .

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. (2014) Graphical methods for visualizing comparative data on phylogenies. Chapter 4 in *Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice* (L. Z. Garamszegi ed.), pp. 77-103.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

Examples

```
## load tree with mapped character
data(anoletree)

## set colors for regims
cols<-setNames(c("green", "#E4D96F", "darkgreen",
  "brown", "black", "darkgrey"),
  c("CG", "GB", "TC", "TG", "Tr", "Tw"))

## create plot
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)

## reset margins to default
par(mar=c(5.1, 4.1, 4.1, 2.1))
```

plotThresh

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

Description

Plots estimated posterior probabilities at nodes under the threshold model.

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% 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.

Details

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](#).

Value

Plots a tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

- Revell, L. J. (2014) Ancestral character estimation under the threshold model from quantitative genetics. *Evolution*, **68**, 743-759.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

`plotTree`*Plots rooted phylogenetic tree*

Description

Essentially a wrapper for `plotSimmap`. 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

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

Details

Plots a rooted phylogram or cladogram.

Value

This function plots a rooted phylogram or cladogram.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2014) Graphical methods for visualizing comparative data on phylogenies. Chapter 4 in *Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice* (L. Z. Garamszegi ed.), pp. 77-103.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

Examples

```
## load tree
data(vertebrate.tree)

## create plot
plotTree(vertebrate.tree, fsize=1.2, ftype="i")

## reset margins to default
par(mar=c(5.1, 4.1, 4.1, 2.1))
```

plotTree.datamatrix	<i>Plot a tree with a discrete (or continuous) character data matrix at the tips</i>
---------------------	--

Description

plotTree.datamatrix plots a phylogeny next to a matrix of discrete characters. plotFanTree.wTraits plots an arc or fan style tree with discrete or continuous data at the tips.

Usage

```
plotTree.datamatrix(tree, X, ...)
plotFanTree.wTraits(tree, X, type=c("arc", "fan"), ...)
```

Arguments

tree	an object of class "phylo".
X	a data frame with columns as factors for plotTree.datamatrix, or a matrix of data frame containing continuous and/or discrete characters for plotFanTree.wTraits.
type	plot type for plotFanTree.wTraits.
...	optional arguments.

Value

plotTree.datamatrix 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.

plotFanTree.wTraits invisibly returns the color palette used for plotting.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[dotTree](#), [phylo.heatmap](#)

Examples

```
## load data
data(liolaemid.data)
data(liolaemid.tree)

## subsample data to species in the tree
liolaemid.data<-liolaemid.data[liolaemid.tree$tip.label,]

## set colors for plotting
colors<-list(
  c("blue","white","red"),
  terrain.colors(n=10),
  setNames(c("#F0EAD6","#DF536B"),c("O","V")))

## plot fan tree with traits
cols<-plotFanTree.wTraits(liolaemid.tree,
  liolaemid.data[,3:1],lwd=12,colors=colors,ftype="off",
  spacer=0)

## add legends
legend(x=0,y=0.7*max(nodeHeights(liolaemid.tree)),
  names(colors[[3]]),lwd=8,col=colors[[3]],
  title="parity mode",bty="n",xjust=0.5,yjust=0.5)
add.color.bar(1.5*max(nodeHeights(liolaemid.tree)),cols[[2]],
  title="maximum altitude (m)",
  lims=range(liolaemid.data[,2]),digits=2,prompt=FALSE,
  x=-0.75*max(nodeHeights(liolaemid.tree)),
  y=0.2*max(nodeHeights(liolaemid.tree)),subtitle="",
  lwd=8,outline=FALSE)
add.color.bar(1.5*max(nodeHeights(liolaemid.tree)),cols[[1]],
  title="environmental temp.",
  lims=range(liolaemid.data[,3]),digits=2,prompt=FALSE,
  x=-0.75*max(nodeHeights(liolaemid.tree)),
  y=-0.15*max(nodeHeights(liolaemid.tree)),subtitle="",
  lwd=8,outline=FALSE)

## reset margins to default
par(mar=c(5.1,4.1,4.1,2.1))
```

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[ltt](#), [plotTree](#)

```
plotTree.lollipop
```

Plot a tree with one or more matched lollipop plots

Description

Plots an upward facing phylogenetic tree with one or more matched lollipop plots. Inspired by a plotting style used by Title et al. (2024).

Usage

```
plotTree.lollipop(tree, x, args.plotTree=list(), args.lollipop=list(), ...)
```

Arguments

tree	an object of class "phylo".
x	a matrix, data frame, or vector of numeric values.
args.plotTree	list of arguments to be passed to plotTree .
args.lollipop	list of arguments for the lollipop plot(s).
...	optional arguments.

Value

Creates a plot.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Title, P. O., Singhal, S., Grundler, M. C., Costa, G. C., Pyron, R. A., Colston, T. J., Grundler, M. R., Prates, I., Stepanova, N., Jones, M. E. H., Cavalcanti, L. B. Q., Colli, G. R., Di-Poi, N., Donnellan, S. C., Moritz, C., Mesquita, D. O., Pianka, E. R., Smith, S. A., Vitt, L. J., and Rabosky, D. L. (2024) The macroevolutionary singularity of snakes. *Science*, **383**, 918-923.

See Also

[phylo.heatmap](#), [plotTree.barplot](#)

Examples

```
## load tree
data(anoletree)

## convert to simple "phylo" object
anole_tree<-as.phylo(anoletree)

## load trait data
data(anole.data)

## create new data frame with size &
## residuals of other traits
anole_data<-cbind(phylo.resid(anole_tree,
  x=as.matrix(anole.data[, "SVL", drop=FALSE]),
  Y=as.matrix(anole.data[, c(6,4,2)]))$resid,
  exp(anole.data[, "SVL", drop=FALSE]))

## create plot
plotTree.lollipop(anole_tree, anole_data,
  ylab=c("relative (TL)", "relative (FLL)",
```

```

"relative (HL)","SVL"))

## reset margins to default
par(mar=c(5.1,4.1,4.1,2.1))

```

plotTree.wBars	<i>Plot a tree with bars at the tips</i>
----------------	--

Description

Plots a phylogenetic tree with adjacent boxplot or barplot.

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, args.boxplot, or args.barplot because y axis limits must match exactly between the two plots.

Details

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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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=4) ## change sci-notation
par(cex.axis=0.8)
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),cex.axis=1)
```

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

Description

Analyzes posterior sample from [evol.rate.mcmc](#).

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 <code>evol.rate.mcmc</code> (probably with burn-in excluded).
tips	list of tips in state σ_1^2 for each sampled generation of MCMC.
showTree	optional logical value indicating whether or not to plot the stretched and shrunk tree generated by the pre-processing algorithm implemented in this function (default is FALSE).

Details

This function 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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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	<i>Generic post-hoc test</i>
---------	------------------------------

Description

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 of the appropriate class containing the results of a posthoc test.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[phylo.toBackbone](#)

pscore	<i>Compute the parsimony score</i>
--------	------------------------------------

Description

Calculates the parsimony score using the Fitch algorithm.

Usage

```
pscore(tree, x, ...)
```

Arguments

tree	object of class "phylo".
x	vector (e.g., factor vector), matrix, or data frame. Should contain names or row names.
...	optional arguments.

Details

Mostly for diagnostic purposes. Users interested in using Maximum Parsimony for phylogeny inference or ancestral state reconstruction should refer to the **phangorn** package.

Value

A numerical value or vector of values.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

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

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[fitMk](#), [sim.Mk](#)

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))

## compute the parsimony score
pscore(sunfish.tree,fmode)
```

ratebystate	<i>Method for investigating the rate of one trait as a function of the state of another</i>
-------------	---

Description

Statistical test of whether the rate of a continuous character might be influenced by the state of another.

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.

Details

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.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[fastAnc](#), [pic](#)

ratebytree	<i>Likelihood test for rate variation among trees, clades, or traits</i>
------------	--

Description

Multiple methods for comparing the rate or process of evolution between trees.

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

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 Mk 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").

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 parameter 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-speciation", 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.
- Revell, L. J. (2018) Comparing the rates of speciation and extinction between phylogenetic trees. *Ecology and Evolution*, **8**, 5303-5312.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.
- 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.
- Stadler, T. (2012) How can we improve the accuracy of macroevolutionary rate estimates? *Systematic Biology*, **62**, 321-329.

See Also

[brownie.light](#), [fitMk](#)

rateshift

Find the temporal position of one or more rate shifts

Description

Fits a model with one or more temporal rate shifts for a continuous trait on the tree.

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.

Details

rateshift attempts to find the location of one or more rate shifts. This model is quite easy to compute the likelihood for, but quite difficult to optimize as the likelihood surface is often rugged. In addition, the model of rateshift is sometimes non-identifiable, i.e., more than one solution may have the same (maximum) likelihood, depending on the data, the tree, and the number of shifts in our model.

likSurface.rateshift plots the likelihood surface.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Revell, L. J. and L. J. Harmon (2022) *Phylogenetic Comparative Methods in R*. Princeton University Press.

See Also

[brownie.lite](#)

read.newick	<i>Newick or Nexus style tree reader</i>
-------------	--

Description

Reads a phylogenetic tree from file.

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` reads a simple Newick style tree from file. This function is now almost completely redundant with [read.tree](#). At the time of development, however, 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` reads a Nexus formatted tree, optionally with bootstrap values as node labels. This function can read a simple 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 singletons (see [collapse.singles](#)); or an object of class "multiPhylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[read.nexus](#), [read.tree](#)

Examples

```
## create text Newick string
tree<-"((Human:7,Chimp:7):3,Gorilla:10):15,Monkey:25);"

## parse string
phy<-read.newick(text=tree)

## create plot
plotTree(phy, fsize=1.2, lwd=5, direction="downwards",
  mar=c(2.1, 5.1, 2.1, 1.1), ylim=c(25, -3.8))
axis(2)
title(ylab="time before present (ma)")
```

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 containing the mapped history of a discrete character.
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).

The function 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

An object of class "simmap" (or list of such objects with class "multiSimmap"), consisting of a modified object of class "phylo" with at least 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

`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

<code>x</code>	an object of class "backbonePhylo".
<code>order</code>	order. See reorder.phylo for possible orderings.
<code>...</code>	optional arguments.

Value

An object of class "backbonePhylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[phylo.toBackbone](#)

reorderSimmap	<i>Reorder edges of a "simmap" tree</i>
---------------	---

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

<code>tree</code>	a modified object of class "phylo".
<code>order</code>	"cladewise", "pruningwise", or any other allowable order permitted by the ape function reorder.phylo .
<code>index.only</code>	logical value indicating whether only an index should be returned.
<code>...</code>	other arguments.

Value

A modified object of class "phylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

`rep.phylo`*Replicate a tree or set of trees*

Description

`rep` method for object of class "phylo" or "multiPhylo".

Usage

```
## S3 method for class 'phylo'
rep(x, ...)
## S3 method for class 'multiPhylo'
rep(x, ...)
repPhylo(tree, times)
```

Arguments

<code>tree</code>	object of class "phylo".
<code>times</code>	number of times to replicate tree.
<code>x</code>	for S3 method an object of class "phylo" or "multiPhylo".
<code>...</code>	other arguments for <code>rep</code> (specifically, <code>times</code>).

Details

`repPhylo` is just an alias for `rep.phylo` and `rep.multiPhylo`.

Value

An object of class "multiPhylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

Examples

```
## create a random tree
tree<-pbtree(n=100)

## create a "multiPhylo" object with 100 copies
trees<-rep(tree,100)
```

reroot	<i>Re-root a tree along an edge</i>
--------	-------------------------------------

Description

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 <i>at</i> 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[paste.tree](#), [root](#), [splitTree](#)

rerootingMethod	<i>Get marginal ancestral state reconstructions by re-rooting</i>
-----------------	---

Description

Computes marginal ancestral states for a discrete character using the re-rooting method of Yang et al. (1995).

In general, this function is **redundant** with [ancr](#) for circumstances in which it is valid (i.e., symmetric **Q** matrices), and improper otherwise. In general [ancr](#) should be preferred.

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 reversible 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 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.

`rerootingMethod` calls `fitMk` internally. `fitMk` uses some code adapted from `ace` in the **ape** package.

Value

An object of class "rerootingMethod" containing at least the following elements:

<code>loglik</code>	the log-likelihood.
<code>Q</code>	the fitted transition matrix between states.
<code>marginal.anc</code>	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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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](#), [ancr](#), [fitMk](#), [make.simap](#)

rescale

Rescale phylogenetic objects of different types

Description

Generic method for rescaling different types of phylogenetic trees.

Usage

```
rescale(x, ...)
```

Arguments

`x` phylogenetic tree object to be rescaled: e.g., object of class "phylo" or "simmap".

`...` other arguments to be used in rescaling, depending on the object class. (E.g., see [rescale.phylo](#) in **geiger** and [rescale.simmap](#).)

Details

See [rescale.phylo](#) in **geiger** and [rescale.simmap](#) for details.

Value

A rescaled phylogenetic tree object.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Pennell, M.W., J. M. Eastman, G. J. Slater, J. W. Brown, J. C. Uyeda, R. G. FitzJohn, M. E. Alfaro, and L. J. Harmon (2014) geiger v2.0: an expanded suite of methods for fitting macroevolutionary models to phylogenetic trees. *Bioinformatics*, **30**, 2216-2218.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[rescale.phylo](#), [rescale.simmap](#)

rescale.simmap	<i>Rescale object of class "simmap"</i>
----------------	---

Description

Scales a tree with a mapped discrete character ("simmap" object), or a set of such trees, to an arbitrary total height, preserving the relative time spent in each state along each edge.

Usage

```
## S3 method for class 'simmap'
rescale(x, model="depth", ...)
## S3 method for class 'multiSimmap'
rescale(x, model="depth", ...)
rescaleSimmap(tree, ...)
```

Arguments

<code>x</code>	object of class "simmap" or "multiSimmap" to be rescaled.
<code>model</code>	model to use to rescale the tree. Currently the only option is "depth".
<code>...</code>	parameter of the model to use in rescaling. Currently the only parameter is depth for model="depth".
<code>tree</code>	for rescaleSimmap, object of class "simmap" to be rescaled.

Details

Replaces `rescaleTree` (now `rescale.phylo`) in the **geiger** package for the "simmap" object class. `rescaleSimmap` is now a redundant alias for the method `rescale.simmap`.

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

Examples

```
## load anoletree
data(anoletree)

## rescale to have total depth of 50
rescaled_anoletree<-rescale(anoletree,depth=50)

## plot rescaled tree
plot(rescaled_anoletree,ftype="i",fsize=0.6,
      mar=c(5.1,1.1,1.1,1.1))
axis(1,at=seq(0,50,by=10))

## reset margin to default
par(mar=c(5.1,4.1,4.1,2.1))
```

resolveNode	<i>Compute all possible resolutions of a node or all nodes in a multifurcating tree</i>
-------------	---

Description

Resolves a single multifurcation or all multifurcations in all possible ways.

Usage

```
resolveNode(tree,node)
resolveAllNodes(tree)
```

Arguments

tree	an object of class "phylo".
node	for resolveNode, the node with a polytomy to resolve.

Details

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.

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" identical to tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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

<code>tree</code>	object of class "phylo".
<code>nodes</code>	either a single node number to rotate, a vector of node numbers, or the string "all".
<code>polytom</code>	a vector of mode numeric and length two specifying the two clades that should be exchanged in a polytomy (see <code>rotate</code>).
<code>node</code>	a single node to rotate (in the case of <code>rotate.multi</code>).
<code>...</code>	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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

roundBranches

Rounds the branch lengths of a tree

Description

Rounds the branch lengths of a phylogenetic tree.

Usage

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

Arguments

tree	an object of class "phylo", "multiPhylo", "simmap", or "multiSimmap".
digits	number of digits for rounding. Passed to round .

Details

This function rounds the branch lengths of a tree or trees to a precision indicated by digits, and reconciles any mappings for objects of class "simmap" or "multiSimmap".

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

roundPhylogram	<i>Plot a round, sigmoidal, or spline phylogram or cladogram</i>
----------------	--

Description

Plots one or multiple round phylograms, a sigmoidal phylogram or cladogram, or a phylogram draw using cubic splines through the nodes.

Usage

```
roundPhylogram(tree, fsize=1.0, ftype="reg", lwd=2, mar=NULL, offset=NULL,
               direction="rightwards", type="phylogram", xlim=NULL, ylim=NULL, ...)
sigmoidPhylogram(tree, ...)
splinePhylogram(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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[plotSimmap](#), [plotTree](#)

Examples

```
## load mammal.tree
data(mammal.tree)

## plot using roundPhylogram
roundPhylogram(compute.brlen(mammal.tree),
  fsize=0.8,ftype="i")

## load anoletree
data(anoletree)

## plot using sigmoidPhylogram
sigmoidPhylogram(anoletree,fsize=0.6,
  ftype="i",direction="upwards")

## reset margin to default
par(mar=c(5.1,4.1,4.1,2.1))
```

rstate

Pick a random state according to a vector of probabilities

Description

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

sampleFrom	<i>Sample from a set of distributions</i>
------------	---

Description

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

setMap

Set color map for various phylogenetic objects of classes

Description

Changes 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 various special types containing a phylogenetic tree. See Description for more details.

... 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.
- Revell, L. J. (2014) Graphical methods for visualizing comparative data on phylogenies. Chapter 4 in *Modern phylogenetic comparative methods and their application in evolutionary biology: Concepts and practice* (L. Z. Garamszegi ed.), pp. 77-103.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[contMap](#), [densityMap](#), [multirateBM](#)

sim.corr

Multivariate Brownian simulation with multiple correlations and rates

Description

Simulates multivariate Brownian motion evolution on a tree with multiple evolutionary correlation/covariance matrices.

Usage

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

Arguments

tree	an object of class "phylo" or "simmap".
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.

Details

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.corr will simulate multivariate BM with a single rate matrix.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

sim.history	<i>Simulate character history or a discrete character at the tips of the tree under some model</i>
-------------	--

Description

Simulates discrete character evolution on a phylogenetic 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 <code>sim.multiMk</code> 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 <code>sim.history</code> <i>by default</i> (i.e., when <code>direction="column_to_row"</code> , see below) normally this is the <i>transpose</i> of the matrix produced by <code>fitDiscrete</code> in the geiger package or <code>make.simmap</code> in phytools ; that is to say the transition rate from $i \rightarrow j$ should be given by $Q[j, i]$. However, if your matrix is properly conformed (i.e., rows <i>or</i> columns sum to 0), then <code>sim.history</code> will attempt to transpose your matrix correctly & will return an informative message (if <code>message=TRUE</code> , see below). For <code>sim.Mk</code> and <code>sim.multiMk</code> this matrix has the same conformation as in <code>fitContinuous</code> and <code>make.simmap</code> . For <code>sim.multiMk</code> 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 <code>NULL</code> , a random state will be assigned. <code>anc</code> can be a vector of states, in which one of the states will be chosen randomly for each simulation. For <code>sim.history</code> <code>anc</code> 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. "column_to_row" indicates that the transition rate from $i \rightarrow j$ should be given by $Q[j, i]$, while "row_to_column" indicates the converse.
...	other optional arguments. Currently only <code>internal</code> , a logical value indicating whether or not to return internal node states (defaults to <code>internal=FALSE</code> ; and <code>message</code> , a logical indicating whether or not to turn on informational messages (defaults to <code>message=TRUE</code>).

Details

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.

Value

`sim.history` returns an object of class "simmap" (a tree with a mapped discrete character) or "multiSimmap" for `nsim` 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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

sim.ratebystate	<i>Conduct simulation of state dependent rate variation</i>
-----------------	---

Description

Simulates two characters under a model in which the rate of one depends on the state of the other.

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 <code>x-min(x)==1</code> (for <code>logarithm=FALSE</code>) or <code>x==0</code> (for <code>logarithm=TRUE</code>). |

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 traitgram ("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 <code>logarithm=FALSE</code> or multiplicative function of x . The default is <code>logarithm=FALSE</code> .

Details

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.

Value

This function returns a matrix.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

sim.rates	<i>Brownian or OU simulation with multiple evolutionary regimes</i>
-----------	---

Description

Simulates multi-rate or multi-regime continuous trait evolution on a phylogeny.

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

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. `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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

simBMphylo	<i>Creates a graphical illustration of Brownian motion evolution on a phylogeny</i>
------------	---

Description

Simulates a discrete time phylogeny and Brownian motion trait, and generates a plot.

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".

Details

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.

Value

An object of class "simBMphylo" or a plot.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[bmPlot](#), [fastBM](#), [pbtree](#)

Examples

```
## set seed
set.seed(777)

## create plot
simBMphylo(n=6,t=100,sig2=0.01)

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

skewers

Matrix comparison using the method of random skewers

Description

Performs the random skewers matrix comparison method of Cheverud (1996).

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 <code>genPositiveDefMat</code> for a more robust hypothesis test (see below). Recommended values include "unifcorrmat".

Details

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.

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. and G. Marroig (2007) Comparing covariance matrices: Random skewers method compared to the common principal components model. *Genetics & Molecular Biology*, **30**, 461–469.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

splitEdgeColor	<i>Split edge colors when descendant edges have different mapped states</i>
----------------	---

Description

Splits the vertical linking-line color on a plotted tree to match the daughter edges.

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.

Details

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](#).

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[plotSimmap](#)

<code>splitplotTree</code>	<i>Plots a phylogeny in two columns</i>
----------------------------	---

Description

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

<code>tree</code>	an object of class "phylo".
<code>fsize</code>	relative font size for tip labels.
<code>ftype</code>	font type - options are "reg", "i" (italics), "b" (bold), or "bi" (bold-italics).
<code>lwd</code>	line width for plotting.
<code>split</code>	relative vertical position for splitting the tree (between 0 & 1).
<code>new.window</code>	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.
<code>splits</code>	for <code>plotTree.splits</code> relative positions (from 0 to 1) to split the tree across pages or devices.
<code>file</code>	filename if saving to a PDF file is desired. Otherwise will plot to the default plotting device.
<code>fn</code>	function to be executed on each plotted page. For instance, might be: <code>function() cladelabels()</code> if clade labels are desired.
<code>...</code>	other arguments to be passed to plotTree .

Value

Plots a tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[plotTree](#), [plotSimmap](#)

splitTree

Split tree at a point

Description

Internal function for [posterior.evolrate](#).

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

This function splits the tree at a given point, and returns the two subtrees as an object of class "multiPhylo".

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

starTree	Create star phylogeny
----------	-----------------------

Description

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[stree](#)

strahlerNumber	<i>Computes Strahler number for trees and nodes</i>
----------------	---

Description

Computes the Strahler number of all nodes and tips in a phylogenetic tree.

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 <code>extract.strahlerNumber</code> .
plot	logical value indicating whether to plot the tree with Strahler numbers for node labels.

Details

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`.

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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

threshBayes

*Threshold model using Bayesian MCMC***Description**

Fits the threshold model from quantitative genetics to data for two discrete characters or one discrete and one continuous trait, following Felsenstein (2012).

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

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.

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

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

Examples

```
## Not run:
## load data
data(bonyfish.tree)
data(bonyfish.data)

## run MCMC
mcmc<-threshBayes(bonyfish.tree,bonyfish.data,
  ngen=100000,plot=FALSE)

## visualize summary of MCMC
plot(mcmc)

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

## graph posterior density of correlation
plot(density(mcmc,bw=0.1))
## End(Not run)
```

threshDIC

Deviance Information Criterion from the threshold model

Description

Computes 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% 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").

Details

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

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. (2014) Ancestral character estimation under the threshold model from quantitative genetics. *Evolution*, **68**, 743-759.
- Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. Van Der Linde (2002) Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **64**, 583-639.

See Also

[ancThresh](#)

threshState	<i>Computes value for a threshold character from a liability and thresholds</i>
-------------	---

Description

Internal function for [ancThresh](#).

Usage

threshState(x, thresholds)

Arguments

x liability.
thresholds a named vector containing the thresholds.

Details

threshState can also be used to simulate threshold traits.

Value

A discrete character value.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2014) Ancestral character estimation under the threshold model from quantitative genetics. *Evolution*, **68**, 743-759.
Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[ancThresh](#), [threshDIC](#)

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

Description

Creates a binary matrix, normally from a factor.

Usage

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

Arguments

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

Details

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

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

tree.grow*Creates an animation of a tree growing from left-to-right or upwards*

Description

Animates a birth-death tree simulation.

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.)

Details

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.

Value

An object of class "phylo".

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[pbtree](#)

treeSlice	<i>Slices the tree at a particular point and returns all subtrees, or the tree rootward of the point</i>
-----------	--

Description

Cut (or slice) a phylogenetic tree at a particular time point.

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).

Details

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. `treeSlice` can also be used to crop the terminal fraction of a tree for `orientation="rootwards"`.

Value

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

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

[extract.clade](#)

untangle	<i>Attempts to untangle crossing branches for plotting</i>
----------	--

Description

Attempts to "untangle" the branches of a tree that are tangled when 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 .

Details

Generally speaking, this function is wraps several different internal functions that might be use to fix a badly conformed "phylo" or "simmap" object.

Value

An object of class "phylo" or "simmap", depending on the class of tree.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

vcvPhylo	<i>Calculates cophenetic (i.e., phylogenetic VCV) matrix</i>
----------	--

Description

Computes a phylogenetic variance-covariance matrix.

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").

Details

This function returns a so-called *phylogenetic variance covariance matrix* (e.g., see [vcv.phylo](#)), but (optionally) including ancestral nodes, as well as under multiple evolutionary models. vcvPhylo is designed primarily for internal use by other *phytools* functions.

Value

A matrix.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

write.simmap	<i>Write a stochastic character mapped tree to file</i>
--------------	---

Description

Writes one or multiple stochastic character mapped trees to file in several formats (following 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. Nielsen, and J. P. Bollback (2003) Stochastic mapping of morphological characters. *Systematic Biology*, **52**, 131-138.

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

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% 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% 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% confidence intervals.

Value

A file, string, or vector of strings.

Author(s)

Liam Revell <liam.revell@umb.edu>

References

Revell, L. J. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

writeNexus	<i>Write a tree to file in Nexus format</i>
------------	---

Description

Writes one or multiple phylogenetic trees to file in NEXUS format.
Somewhat redundant with [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. (2024) phytools 2.0: an updated R ecosystem for phylogenetic comparative methods (and other things). *PeerJ*, **12**, e16505.

See Also

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

Index

* ancestral states

- anc.Bayes, 15
- anc.ML, 17
- anc.trend, 19
- ancr, 21
- ancThresh, 23
- bounded_bm, 40
- contMap, 52
- countSimmap, 58
- ctt, 59
- density.multiSimmap, 60
- densityMap, 61
- describe.simmap, 64
- edge.widthMap, 74
- edgeProbs, 75
- fastAnc, 88
- fitcontMk, 97
- fitmultiBM, 104
- fitThresh, 108
- make.simmap, 132
- map.overlap, 135
- markChanges, 139
- mergeMappedStates, 141
- phenogram, 162
- phylomorphospace, 180
- phylomorphospace3d, 182
- plotBranchbyTrait, 187
- plotSimmap, 189
- plotThresh, 191
- rerootingMethod, 216

* animation

- branching.diffusion, 42
- phylomorphospace3d, 182

* bayesian

- anc.Bayes, 15
- ancThresh, 23
- ave.rates, 33
- Dtest, 73
- evol.rate.mcmc, 77

- fitBayes, 96

- fitMk, 99

- make.simmap, 132

- markChanges, 139

- mergeMappedStates, 141

- minSplit, 144

- plotThresh, 191

- posterior.evolrate, 200

- rerootingMethod, 216

- threshBayes, 239

- threshDIC, 241

* biogeography

- phylo.to.map, 176

* co-phylogenetics

- cophylo, 54

- cospeciation, 56

* comparative methods

- fit.bd, 94

* comparative method

- add.color.bar, 8

- add.simmap.legend, 11

- anc.Bayes, 15

- anc.ML, 17

- anc.trend, 19

- ancr, 21

- ancThresh, 23

- as.Qmatrix, 32

- ave.rates, 33

- bd, 35

- bounded_bm, 40

- brownie.lite, 43

- brownieREML, 45

- contMap, 52

- ctt, 59

- density.multiSimmap, 60

- densityMap, 61

- describe.simmap, 64

- di2multi.simmap, 66

- Dtest, 73

- edge.widthMap, 74
- estDiversity, 76
- evol.rate.mcmc, 77
- evol.vcv, 79
- evolvcv.lite, 81
- fancyTree, 86
- fastAnc, 88
- fitBayes, 96
- fitcontMk, 97
- fitDiversityModel, 98
- fitMk, 99
- fitmultiBM, 104
- fitPagel, 106
- fitThresh, 108
- gammatest, 111
- lambda.transform, 122
- likMlambda, 123
- make.era.map, 131
- make.simap, 132
- map.overlap, 135
- mapped.states, 138
- mergeMappedStates, 141
- minSplit, 144
- multi.mantel, 147
- multirateBM, 149
- nodeHeights, 152
- paintSubTree, 156
- pgls.Ives, 160
- phenogram, 162
- phyl.cca, 164
- phyl.pairedttest, 165
- phyl.pca, 167
- phyl.resid, 169
- phyl.RMA, 170
- phylANOVA, 173
- phylo.impute, 175
- phylomorphospace, 180
- phylomorphospace3d, 182
- phylosig, 184
- plotBranchbyTrait, 187
- plotSimap, 189
- plotThresh, 191
- plotTree.datamatrix, 194
- plotTree.lollipop, 196
- plotTree.wBars, 198
- posterior.evolrate, 200
- pscore, 203
- ratebystate, 204
- ratebytree, 205
- rateshift, 207
- read.simap, 210
- reorderSimap, 213
- rerootingMethod, 216
- roundPhylogram, 223
- sim.corr, 227
- sim.ratebystate, 229
- sim.rates, 230
- simBMphylo, 232
- skewers, 233
- splitEdgeColor, 234
- threshBayes, 239
- threshDIC, 241
- threshState, 242
- tree.grow, 243
- write.simap, 247
- writeAncestors, 248
- * consensus tree**
 - averageTree, 34
 - consensus.edges, 51
 - ls.tree, 127
 - mrp.supertree, 146
- * continuous characters**
 - ave.rates, 33
- * continuous character**
 - anc.Bayes, 15
 - anc.ML, 17
 - anc.trend, 19
 - bmPlot, 38
 - bounded_bm, 40
 - branching.diffusion, 42
 - brownie.lite, 43
 - brownieREML, 45
 - contMap, 52
 - dotTree, 67
 - edge.widthMap, 74
 - estDiversity, 76
 - evol.rate.mcmc, 77
 - evol.vcv, 79
 - evolvcv.lite, 81
 - fastAnc, 88
 - fastBM, 90
 - fitBayes, 96
 - fitcontMk, 97
 - fitDiversityModel, 98
 - fitmultiBM, 104
 - locate.fossil, 125

- locate.yeti, 126
- multirateBM, 149
- phenogram, 162
- phylo.heatmap, 174
- phylo.impute, 175
- phylomorphospace, 180
- phylomorphospace3d, 182
- phylosig, 184
- plotBranchbyTrait, 187
- plotTree.lollipop, 196
- plotTree.wBars, 198
- posterior.evolrate, 200
- ratebystate, 204
- ratebytree, 205
- rateshift, 207
- sim.corr, 227
- sim.ratebystate, 229
- sim.rates, 230
- simBMphylo, 232
- threshBayes, 239
- * datasets**
 - anoletree, 25
- * discrete character**
 - add.simap.legend, 11
 - ancThresh, 23
 - as.Qmatrix, 32
 - countSimmap, 58
 - ctt, 59
 - density.multiSimmap, 60
 - densityMap, 61
 - describe.simap, 64
 - dotTree, 67
 - Dtest, 73
 - edgeProbs, 75
 - fitMk, 99
 - fitPagel, 106
 - fitThresh, 108
 - genSeq, 113
 - getStates, 120
 - make.simap, 132
 - map.overlap, 135
 - markChanges, 139
 - mergeMappedStates, 141
 - plotSimmap, 189
 - plotThresh, 191
 - plotTree.datamatrix, 194
 - ratebytree, 205
 - read.simap, 210
 - rstate, 224
 - sim.history, 228
 - threshBayes, 239
 - threshState, 242
- * distance matrix**
 - multi.mantel, 147
 - optim.phylo.ls, 154
- * diversification**
 - bd, 35
 - cophylo, 54
 - cospeciation, 56
 - estDiversity, 76
 - fit.bd, 94
 - fitDiversityModel, 98
 - gammatest, 111
 - litt, 127
 - litt95, 130
 - pbtrees, 158
 - tree.grow, 243
- * information criteria**
 - aic.w, 13
 - threshDIC, 241
- * input/output**
 - read.newick, 209
 - read.simap, 210
 - write.simap, 247
 - writeAncestors, 248
 - writeNexus, 249
- * least squares**
 - ls.tree, 127
 - multi.mantel, 147
 - optim.phylo.ls, 154
 - pgls.Ives, 160
 - phyl.cca, 164
 - phyl.pairedttest, 165
 - phyl.resid, 169
 - phylANOVA, 173
 - phyloDesign, 179
- * math**
 - aic.w, 13
 - ave.rates, 33
 - lambda.transform, 122
 - likMLambda, 123
 - rstate, 224
- * maximum likelihood**
 - aic.w, 13
 - anc.ML, 17
 - anc.trend, 19

- ancr, 21
- as.Qmatrix, 32
- bd, 35
- bounded_bm, 40
- brownie.lite, 43
- brownieREML, 45
- contMap, 52
- edge.widthMap, 74
- estDiversity, 76
- evol.vcv, 79
- evolvcv.lite, 81
- fastAnc, 88
- fit.bd, 94
- fitcontMk, 97
- fitDiversityModel, 98
- fitMk, 99
- fitmultiBM, 104
- fitPagel, 106
- likMlambda, 123
- locate.fossil, 125
- locate.yeti, 126
- multirateBM, 149
- pgls.Ives, 160
- phyl.pairedttest, 165
- phyl.pca, 167
- phyl.resid, 169
- phyl.RMA, 170
- phylosig, 184
- pscore, 203
- ratebystate, 204
- ratebytree, 205
- rateshift, 207
- rerootingMethod, 216
- * **package**
 - phytools-package, 6
- * **parsimony**
 - exhaustiveMP, 83
 - mrp.supertree, 146
- * **phylogenetics**
 - add.arrow, 6
 - add.color.bar, 8
 - add.everywhere, 9
 - add.random, 10
 - add.simmap.legend, 11
 - add.species.to.genus, 12
 - allFurcTrees, 14
 - anc.Bayes, 15
 - anc.ML, 17
 - anc.trend, 19
 - ancr, 21
 - ancThresh, 23
 - ansi.phylo, 29
 - applyBranchLengths, 30
 - as.multiPhylo, 31
 - as.Qmatrix, 32
 - ave.rates, 33
 - averageTree, 34
 - bd, 35
 - bind.tip, 36
 - bind.tree.simmap, 37
 - bmPlot, 38
 - bounded_bm, 40
 - branching.diffusion, 42
 - brownie.lite, 43
 - brownieREML, 45
 - cladelabels, 47
 - collapse.to.star, 48
 - collapseTree, 49
 - compare.chronograms, 50
 - consensus.edges, 51
 - contMap, 52
 - cophylo, 54
 - cospeciation, 56
 - countSimmap, 58
 - ctt, 59
 - density.multiSimmap, 60
 - densityMap, 61
 - densityTree, 63
 - describe.simmap, 64
 - di2multi.simmap, 66
 - dotTree, 67
 - drop.clade, 69
 - drop.leaves, 69
 - drop.tip.contMap, 70
 - drop.tip.multiSimmap, 71
 - drop.tip.simmap, 72
 - Dtest, 73
 - edge.widthMap, 74
 - edgeProbs, 75
 - estDiversity, 76
 - evol.rate.mcmc, 77
 - evol.vcv, 79
 - evolvcv.lite, 81
 - exhaustiveMP, 83
 - expand.clade, 84
 - export.as.xml, 85

fancyTree, 86
fastAnc, 88
fastBM, 90
fastMRCA, 91
findMRCA, 92
fit.bd, 94
fitBayes, 96
fitcontMk, 97
fitDiversityModel, 98
fitMk, 99
fitmultiBM, 104
fitPagel, 106
fitThresh, 108
gammatest, 111
genSeq, 113
geo.legend, 114
getCladesofSize, 116
getDescendants, 117
getExtant, 118
getSisters, 119
getStates, 120
ladderize.simap, 121
lambda.transform, 122
likMlambda, 123
locate.fossil, 125
locate.yeti, 126
ls.tree, 127
ltt, 127
ltt95, 130
make.era.map, 131
make.simap, 132
map.overlap, 135
map.to.singleton, 137
mapped.states, 138
markChanges, 139
matchNodes, 140
mergeMappedStates, 141
midpoint_root, 142
minRotate, 143
minSplit, 144
modified.Grafen, 145
mrp.supertree, 146
multiC, 148
multirateBM, 149
multiRF, 151
nodeHeights, 152
nodelabels.cophylo, 153
optim.phylo.ls, 154
orderMappedEdge, 155
paintSubTree, 156
paste.tree, 157
pbtree, 158
pgls.Ives, 160
phenogram, 162
phyl.cca, 164
phyl.pairedttest, 165
phyl.pca, 167
phyl.resid, 169
phyl.RMA, 170
phyl.vcv, 172
phylANOVA, 173
phylo.heatmap, 174
phylo.impute, 175
phylo.to.map, 176
phylo.toBackbone, 178
phyloDesign, 179
phylomorphospace, 180
phylomorphospace3d, 182
phylosig, 184
plot.backbonePhylo, 186
plotBranchbyTrait, 187
plotSimap, 189
plotThresh, 191
plotTree, 193
plotTree.datamatrix, 194
plotTree.errorbars, 195
plotTree.lollipop, 196
plotTree.wBars, 198
posterior.evolrate, 200
print.backbonePhylo, 202
pscore, 203
ratebystate, 204
ratebytree, 205
rateshift, 207
read.newick, 209
read.simap, 210
reorder.backbonePhylo, 212
reorderSimap, 213
rep.phylo, 214
reroot, 215
rerootingMethod, 216
rescale, 217
rescale.simap, 218
resolveNode, 220
rotateNodes, 221
roundBranches, 222

- roundPhylogram, 223
- sampleFrom, 225
- setMap, 226
- sim.corr, 227
- sim.history, 228
- sim.ratebystate, 229
- sim.rates, 230
- simBMphylo, 232
- splitEdgeColor, 234
- splitplotTree, 235
- splitTree, 236
- starTree, 237
- strahlerNumber, 238
- threshBayes, 239
- threshDIC, 241
- threshState, 242
- tree.grow, 243
- treeSlice, 244
- untangle, 245
- vcvPhylo, 246
- write.simap, 247
- writeAncestors, 248
- writeNexus, 249
- * phylogeny inference**
 - allFurcTrees, 14
 - consensus.edges, 51
 - exhaustiveMP, 83
 - locate.fossil, 125
 - locate.yeti, 126
 - ls.tree, 127
 - midpoint_root, 142
 - mrp.supertree, 146
 - multiRF, 151
 - optim.phylo.ls, 154
 - phyloDesign, 179
- * plotting**
 - add.arrow, 6
 - add.color.bar, 8
 - add.simap.legend, 11
 - ansi_phylo, 29
 - bmPlot, 38
 - branching.diffusion, 42
 - collapseTree, 49
 - compare.chronograms, 50
 - contMap, 52
 - cophylo, 54
 - cospeciation, 56
 - ctt, 59
 - densityMap, 61
 - densityTree, 63
 - dotTree, 67
 - edge.widthMap, 74
 - edgeProbs, 75
 - expand.clade, 84
 - fancyTree, 86
 - geo.legend, 114
 - get.treepos, 115
 - labelnodes, 120
 - linklabels, 124
 - lft, 127
 - lft95, 130
 - map.to.singleton, 137
 - markChanges, 139
 - phenogram, 162
 - phylo.heatmap, 174
 - phylo.to.map, 176
 - phylo.toBackbone, 178
 - phylomorphospace, 180
 - phylomorphospace3d, 182
 - plot.backbonePhylo, 186
 - plotBranchbyTrait, 187
 - plotSimap, 189
 - plotTree, 193
 - plotTree.datamatrix, 194
 - plotTree.errorbars, 195
 - plotTree.lollipop, 196
 - plotTree.wBars, 198
 - print.backbonePhylo, 202
 - reorder.backbonePhylo, 212
 - roundPhylogram, 223
 - setMap, 226
 - simBMphylo, 232
 - splitEdgeColor, 234
 - splitplotTree, 235
 - tree.grow, 243
- * simulation**
 - bmPlot, 38
 - branching.diffusion, 42
 - Dtest, 73
 - fastBM, 90
 - genSeq, 113
 - make.simap, 132
 - mergeMappedStates, 141
 - pbtrees, 158
 - phylANOVA, 173
 - phylosig, 184

- rstate, 224
- sim.corr, 227
- sim.history, 228
- sim.ratebystate, 229
- sim.rates, 230
- threshState, 242
- tree.grow, 243
- * **statistics**
 - aic.w, 13
 - force.ultrametric, 110
 - multi.mantel, 147
 - pgls.Ives, 160
 - phyl.cca, 164
 - phyl.pairedttest, 165
 - phyl.pca, 167
 - phyl.resid, 169
 - phyl.RMA, 170
 - phyl.vcv, 172
 - phylANOVA, 173
 - phylo.impute, 175
 - posthoc, 201
 - rstate, 224
 - sampleFrom, 225
 - skewers, 233
 - threshDIC, 241
 - vcvPhylo, 246
- * **supertree**
 - mrp.supertree, 146
- * **utilities**
 - add.arrow, 6
 - add.color.bar, 8
 - add.everywhere, 9
 - add.random, 10
 - add.simap.legend, 11
 - add.species.to.genus, 12
 - aic.w, 13
 - applyBranchLengths, 30
 - as.multiPhylo, 31
 - as.Qmatrix, 32
 - bd, 35
 - bind.tip, 36
 - bind.tree.simap, 37
 - cladelabels, 47
 - collapse.to.star, 48
 - collapseTree, 49
 - consensus.edges, 51
 - countSimap, 58
 - density.multiSimap, 60
 - describe.simap, 64
 - di2multi.simap, 66
 - drop.clade, 69
 - drop.leaves, 69
 - drop.tip.contMap, 70
 - drop.tip.multiSimap, 71
 - drop.tip.simap, 72
 - expand.clade, 84
 - export.as.xml, 85
 - fastMRCA, 91
 - findMRCA, 92
 - force.ultrametric, 110
 - gamma_pruning, 112
 - get.treepos, 115
 - getCladesofSize, 116
 - getDescendants, 117
 - getExtant, 118
 - getSisters, 119
 - getStates, 120
 - labelnodes, 120
 - ladderize.simap, 121
 - lambda.transform, 122
 - likMLambda, 123
 - linklabels, 124
 - map.to.singleton, 137
 - mapped.states, 138
 - matchNodes, 140
 - mergeMappedStates, 141
 - midpoint_root, 142
 - minRotate, 143
 - modified.Grafen, 145
 - multiC, 148
 - multiRF, 151
 - nodeHeights, 152
 - nodelabels.cophylo, 153
 - orderMappedEdge, 155
 - paintSubTree, 156
 - paste.tree, 157
 - phyl.vcv, 172
 - reorder.backbonePhylo, 212
 - reorderSimap, 213
 - rep.phylo, 214
 - reroot, 215
 - rescale, 217
 - rescale.simap, 218
 - resolveNode, 220
 - rotateNodes, 221
 - roundBranches, 222

- rstater, 224
 - sampleFrom, 225
 - splitTree, 236
 - starTree, 237
 - strahlerNumber, 238
 - threshState, 242
 - to.matrix, 243
 - treeSlice, 244
 - untangle, 245
 - vcvPhylo, 246
- ace, 16, 18, 20, 76, 89, 102, 103, 106, 107, 132, 133, 206, 217, 248, 249
- add.arrow, 6
- add.color.bar, 8, 188
- add.everywhere, 9, 11, 15
- add.random, 10, 13
- add.simmap.legend, 11
- add.species.to.genus, 12
- aic.w, 13
- all.equal.phylo, 65
- allFurcTrees, 10, 11, 14
- allRotations (rotateNodes), 221
- anc.Bayes, 15, 18, 20, 24, 79, 89, 97, 240
- anc.ML, 16, 17, 18, 20, 54, 87, 89, 182
- anc.trend, 16, 19
- ancr, 16, 18, 21, 89, 135, 216, 217
- ancThresh, 23, 109, 192, 241, 242
- anole.data (anoletree), 25
- anoletree, 25
- anova, 174
- ansi_phylo, 29
- ant.geog (anoletree), 25
- ant.tree (anoletree), 25
- applyBranchLengths, 30
- arc.cladelabels (cladelabels), 47
- as.multiPhylo, 31
- as.prcomp (phyl.pca), 167
- as.princomp (phyl.pca), 167
- as.Qmatrix, 32
- ave.rates, 33
- averageTree, 34
- backbone.toPhylo (phylo.toBackbone), 178
- barplot, 199
- bat.tree (anoletree), 25
- bat_virus.data (anoletree), 25
- bd, 35
- betaCoV.tree (anoletree), 25
- bind.tip, 13, 36
- bind.tree, 36, 38, 159
- bind.tree.simmap, 37
- biplot, 167
- biplot.phyl.pca (phyl.pca), 167
- birthdeath, 35, 95
- bmPlot, 38, 43, 233, 240
- bonyfish.data (anoletree), 25
- bonyfish.tree (anoletree), 25
- bounded_bm, 40, 40, 98, 105, 109
- boxplot, 199
- branching.diffusion, 42
- brownie.lite, 43, 45, 46, 79, 80, 82, 97–99, 105, 135, 149, 150, 162, 206, 207, 209, 212
- brownieREML, 44, 45, 135
- butterfly.data (anoletree), 25
- butterfly.tree (anoletree), 25
- c.phylo, 214
- cladebox (cladelabels), 47
- cladelabels, 47, 121, 125
- coef.phyl.RMA (phyl.RMA), 170
- collapse.singles, 138, 210
- collapse.to.star, 48
- collapseTree, 49
- colorRampPalette, 188, 226
- compare.chronograms, 50
- compute.brLen, 146
- compute.mr (mrp.supertree), 146
- consensus.edges, 51
- contMap, 9, 52, 67, 70, 71, 75, 86, 87, 227
- cophylo, 54, 57, 144, 154
- cophyloplot, 56
- cor, 204
- cordylid.data (anoletree), 25
- cordylid.tree (anoletree), 25
- cospeciation, 56
- cotangleplot (cophylo), 54
- countSimmap, 58, 65, 135
- ctt, 59
- darter.tree (anoletree), 25
- density.anc.Bayes (anc.Bayes), 15
- density.mcmcMk (fitMk), 99
- density.multiSimmap, 60
- densityMap, 9, 54, 60, 61, 61, 67, 70, 71, 86, 87, 191, 227
- densityTree, 63

- Descendants, [117](#)
- describe.simmap, [64](#), [120](#), [135](#)
- dev.hold, [87](#), [163](#)
- di2multi, [66](#), [67](#)
- di2multi.contMap(di2multi.simmap), [66](#)
- di2multi.densityMap(di2multi.simmap), [66](#)
- di2multi.multiSimmap(di2multi.simmap), [66](#)
- di2multi.simmap, [66](#)
- dot.legend(dotTree), [67](#)
- dotTree, [67](#), [195](#), [199](#)
- drop.clade, [69](#)
- drop.leaves, [69](#)
- drop.tip, [70–73](#), [87](#), [138](#)
- drop.tip.contMap, [70](#)
- drop.tip.densityMap(drop.tip.contMap), [70](#)
- drop.tip.multiPhylo, [72](#)
- drop.tip.multiSimmap, [71](#)
- drop.tip.simmap, [70–72](#), [72](#)
- drop.tip.singleton(map.to.singleton), [137](#)
- Dtest, [73](#)
- edge.widthMap, [74](#)
- edgelabels, [153](#), [154](#)
- edgelabels.cophylo
(nodelabels.cophylo), [153](#)
- edgeProbs, [75](#)
- eel.data(anoletree), [25](#)
- eel.tree(anoletree), [25](#)
- elapidae.tree(anoletree), [25](#)
- errorbar.contMap, [9](#)
- errorbar.contMap(contMap), [52](#)
- estDiversity, [76](#), [99](#)
- evol.rate.mcmc, [16](#), [33](#), [46](#), [77](#), [80](#), [97](#), [99](#),
[145](#), [150](#), [200](#), [201](#), [240](#)
- evol.vcv, [44](#), [46](#), [79](#), [79](#), [82](#), [135](#), [176](#), [212](#)
- evolvcv.lite, [81](#), [149](#)
- exhaustiveMP, [10](#), [15](#), [83](#), [147](#), [155](#)
- expand.clade, [84](#)
- export.as.xml, [85](#)
- extract.clade, [72](#), [73](#), [116](#), [245](#)
- extract.clade.simmap(drop.tip.simmap),
[72](#)
- extract.strahlerNumber
(strahlerNumber), [238](#)
- fancyTree, [86](#), [183](#)
- fastAnc, [16](#), [18](#), [53](#), [54](#), [75](#), [88](#), [180](#), [205](#), [248](#),
[249](#)
- fastBM, [40](#), [43](#), [90](#), [204](#), [228](#), [230](#), [231](#), [233](#)
- fastDist(fastMRCA), [91](#)
- fastHeight(fastMRCA), [91](#)
- fastMRCA, [91](#)
- findMRCA, [92](#), [92](#), [93](#)
- fit.bd, [35](#), [94](#)
- fit.yule, [35](#)
- fit.yule(fit.bd), [94](#)
- fitBayes, [96](#)
- fitcontMk, [97](#)
- fitDiversityModel, [77](#), [98](#)
- fitfnMk(fitMk), [99](#)
- fitgammaMk(fitMk), [99](#)
- fitHRM(fitMk), [99](#)
- fitMk, [22](#), [33](#), [41](#), [98](#), [99](#), [105](#), [107](#), [133](#), [203](#),
[207](#), [217](#), [229](#)
- fitmultiBM, [41](#), [98](#), [104](#), [109](#)
- fitmultiMk, [229](#)
- fitmultiMk(fitMk), [99](#)
- fitmultiTrend(fitmultiBM), [104](#)
- fitPagel, [98](#), [106](#)
- fitpolyMk(fitMk), [99](#)
- fitsemiThresh(fitThresh), [108](#)
- fitThresh, [41](#), [108](#)
- flatworm.data(anoletree), [25](#)
- flatworm.tree(anoletree), [25](#)
- force.ultrametric, [13](#), [110](#)
- gamma_pruning, [112](#)
- gammatest, [111](#), [129](#)
- genSeq, [113](#)
- genus.to.species.tree
(add.species.to.genus), [12](#)
- geo.legend, [114](#)
- geo.palette(geo.legend), [114](#)
- get.treepos, [115](#)
- getCladesofSize, [116](#)
- getDescendants, [116](#), [117](#), [119](#)
- getExtant, [118](#)
- getExtinct(getExtant), [118](#)
- getMRCA, [92](#), [93](#)
- getNode(get.treepos), [115](#)
- getParent(getDescendants), [117](#)
- getSisters, [119](#)
- getStates, [120](#)
- ginv, [206](#)

- gls, [161](#), [170](#)
- graph.polyMk (fitMk), [99](#)
- gtt (ltt), [127](#)
- heatmap, [174](#)
- hide.hidden (ancr), [21](#)
- image, [174](#)
- is.ultrametric, [110](#), [111](#)
- keep.tip, [70–73](#)
- keep.tip.contMap (drop.tip.contMap), [70](#)
- keep.tip.densityMap (drop.tip.contMap), [70](#)
- keep.tip.multiSimmap (drop.tip.multiSimmap), [71](#)
- keep.tip.simmap (drop.tip.simmap), [72](#)
- labelnodes, [120](#)
- ladderize, [122](#)
- ladderize.simmap, [121](#)
- lambda.transform, [122](#)
- legend, [11](#)
- lik.bd (fit.bd), [94](#)
- likMlambda, [123](#)
- likSurface.rateshift (rateshift), [207](#)
- linklabels, [124](#)
- liolaamid.data (anoletree), [25](#)
- liolaamid.tree (anoletree), [25](#)
- locate.fossil, [125](#)
- locate.yeti, [125](#), [126](#)
- logLik.fitDiversityModel (fitDiversityModel), [98](#)
- ls.consensus (averageTree), [34](#)
- ls.tree, [127](#)
- ltt, [60](#), [111](#), [127](#), [130](#), [131](#), [196](#)
- ltt95, [129](#), [130](#)
- make.era.map, [131](#)
- make.simmap, [43](#), [45](#), [60–62](#), [67](#), [73](#), [74](#), [76](#), [85](#), [103](#), [107](#), [120](#), [122](#), [132](#), [132](#), [136](#), [138](#), [142](#), [155](#), [157](#), [189](#), [191](#), [217](#), [219](#), [224](#), [228](#), [229](#), [231](#), [243](#), [248](#)
- make.transparent (densityTree), [63](#)
- mammal.data (anoletree), [25](#)
- mammal.geog (anoletree), [25](#)
- mammal.tree (anoletree), [25](#)
- map, [177](#)
- Map.Overlap (map.overlap), [135](#)
- map.overlap, [74](#), [135](#)
- map.to.singleton, [137](#)
- mapped.states, [138](#)
- markChanges, [139](#)
- matchLabels (matchNodes), [140](#)
- matchNodes, [140](#)
- mccr, [111](#)
- mccr (ltt), [127](#)
- mcmcMk (fitMk), [99](#)
- mergeMappedStates, [141](#)
- midpoint, [142](#), [143](#)
- midpoint.root (midpoint_root), [142](#)
- midpoint_root, [142](#)
- minRotate, [143](#)
- minSplit, [33](#), [79](#), [144](#), [200](#), [201](#)
- minTreeDist (averageTree), [34](#)
- modified.Grafen, [145](#)
- mrca, [92](#), [93](#)
- mrp.supertree, [84](#), [146](#)
- multi.mantel, [147](#)
- multi2di, [66](#), [67](#), [220](#)
- multi2di.contMap (di2multi.simmap), [66](#)
- multi2di.densityMap (di2multi.simmap), [66](#)
- multi2di.multiSimmap (di2multi.simmap), [66](#)
- multi2di.simmap (di2multi.simmap), [66](#)
- multiC, [148](#)
- multiOU (sim.rates), [230](#)
- multirateBM, [149](#), [227](#)
- multiRF, [151](#)
- nni, [154](#), [155](#)
- nnls.tree, [110](#), [111](#)
- node.paths (modified.Grafen), [145](#)
- nodeheight (nodeHeights), [152](#)
- nodeHeights, [118](#), [152](#)
- nodelabels, [7](#), [21](#), [48](#), [121](#), [125](#), [153](#), [154](#), [190](#)
- nodelabels.cophylo, [153](#)
- optim, [18–20](#), [46](#), [80](#), [149](#), [162](#), [166](#), [184](#), [185](#)
- optim.parsimony, [84](#), [146](#), [147](#)
- optim.phylo.ls, [127](#), [154](#), [179](#)
- optimize, [165](#), [167](#), [169](#), [171](#), [184](#)
- orderMappedEdge, [155](#)
- p.adjust, [173](#)
- paintBranches, [234](#)

- paintBranches (paintSubTree), 156
- paintSubTree, 43, 101, 117, 156, 234
- pairwise.t.test, 173, 174
- palette, 137
- par, 130, 163, 174, 177, 181, 189, 190, 223
- parsimony, 83
- paste.tree, 157, 215, 216
- pbtrees, 40, 158, 233, 244
- pgls.Ives, 160
- pgls.SEy (pgls.Ives), 160
- phenogram, 40, 87, 162, 183
- phenogram95 (fancyTree), 86
- phyDat, 83
- phyl.cca, 164, 168, 171
- phyl.pairedttest, 165
- phyl.pca, 122, 123, 165, 167, 170–172
- phyl.resid, 162, 168, 169, 171
- phyl.RMA, 170
- phyl.vcv, 172
- phylANOVA, 173
- phylo.heatmap, 68, 174, 195, 197
- phylo.impute, 175
- phylo.to.map, 144, 176
- phylo.toBackbone, 178, 187, 202, 212
- phyloDesign, 179
- phylomorphospace, 87, 180, 183
- phylomorphospace3d, 87, 182
- phyloScattergram, 87
- phyloScattergram (fancyTree), 86
- phylosig, 162, 184
- phytools (phytools-package), 6
- phytools-package, 6
- pic, 89, 205
- plot, 16, 128
- plot.anc.Bayes (anc.Bayes), 15
- plot.ancr (ancr), 21
- plot.backbonePhylo, 179, 186
- plot.changesMap (density.multiSimmap), 60
- plot.contMap, 9
- plot.contMap (contMap), 52
- plot.cophylo (cophylo), 54
- plot.cospeciation (cospeciation), 56
- plot.default, 128, 163, 181
- plot.density, 16
- plot.density.mcmcMk (fitMk), 99
- plot.densityMap, 9
- plot.densityMap (densityMap), 61
- plot.describe.simmap (describe.simmap), 64
- plot.edge.widthMap (edge.widthMap), 74
- plot.expand.clade (expand.clade), 84
- plot.fitHRM (fitMk), 99
- plot.fitMk (fitMk), 99
- plot.fitPagel (fitPagel), 106
- plot.fitpolyMk (fitMk), 99
- plot.gfit (fitMk), 99
- plot.ltt95 (ltt95), 130
- plot.mcmcMk (fitMk), 99
- plot.multiSimmap (plotSimmap), 189
- plot.phyl.RMA (phyl.RMA), 170
- plot.phylo, 10, 87, 137, 188, 190, 192, 193, 245
- plot.phylo.to.map (phylo.to.map), 176
- plot.phylosig (phylosig), 184
- plot.Qmatrix (as.Qmatrix), 32
- plot.rateshift (rateshift), 207
- plot.simBMphylo (simBMphylo), 232
- plot.simmap (plotSimmap), 189
- plot.summary.evol.rate.mcmc (evol.rate.mcmc), 77
- plot.window, 190
- plotBranchbyTrait, 9, 187
- plotFanTree.wTraits (plotTree.datamatrix), 194
- plotSimmap, 10, 12, 49, 50, 52, 54–56, 61, 62, 76, 87, 132, 135, 139, 157, 174, 177, 189, 190, 193, 198, 199, 208, 213, 223, 224, 229, 234–236, 245, 248
- plotThresh, 24, 191
- plotTree, 21, 30, 50, 68, 74, 137, 193, 196–199, 223, 224, 235, 236, 245
- plotTree.barplot, 197
- plotTree.barplot (plotTree.wBars), 198
- plotTree.boxplot (plotTree.wBars), 198
- plotTree.datamatrix, 194
- plotTree.errorbars, 195
- plotTree.lollipop, 196
- plotTree.singletons (map.to.singleton), 137
- plotTree.splits (splitplotTree), 235
- plotTree.wBars, 198
- posterior.evolrate, 33, 69, 79, 145, 157, 200, 236
- posthoc, 201
- posthoc.ratebytree (ratebytree), 205

- pratchet, [84](#), [146](#), [147](#)
- prcomp, [168](#)
- primate.data (anoletree), [25](#)
- primate.tree (anoletree), [25](#)
- princomp, [168](#)
- print.backbonePhylo, [202](#)
- print.cospeciation (cospeciation), [56](#)
- print.evol.rate.mcmc (evol.rate.mcmc), [77](#)
- print.fit.bd (fit.bd), [94](#)
- print.fitDiversityModel (fitDiversityModel), [98](#)
- print.Qmatrix (as.Qmatrix), [32](#)
- print.summary.evol.rate.mcmc (evol.rate.mcmc), [77](#)
- project.phylomorphospace (phylomorphospace), [180](#)
- prop.part, [146](#), [151](#)
- pscore, [203](#)
- ratebystate, [204](#), [230](#)
- ratebytree, [44](#), [46](#), [202](#), [205](#)
- rateshift, [207](#)
- read.newick, [209](#)
- read.nexus, [85](#), [209–212](#)
- read.simap, [31](#), [43–45](#), [61](#), [62](#), [67](#), [73](#), [85](#), [120](#), [132](#), [135](#), [136](#), [142](#), [149](#), [157](#), [189](#), [191](#), [210](#), [219](#), [228](#), [229](#), [231](#), [248](#)
- read.tree, [209](#), [210](#), [212](#), [246](#)
- readNexus (read.newick), [209](#)
- reorder.backbonePhylo, [212](#)
- reorder.phylo, [212](#), [213](#), [246](#)
- reorderSimmap, [213](#), [246](#)
- rep, [214](#)
- rep.multiPhylo (rep.phylo), [214](#)
- rep.phylo, [214](#)
- repPhylo (rep.phylo), [214](#)
- reroot, [143](#), [215](#)
- rerootingMethod, [216](#), [243](#)
- rescale, [217](#)
- rescale.multiSimmap (rescale.simap), [218](#)
- rescale.phylo, [218](#)
- rescale.simap, [218](#), [218](#)
- rescaleSimmap (rescale.simap), [218](#)
- resolveAllNodes (resolveNode), [220](#)
- resolveNode, [220](#)
- rmultinom, [224](#)
- root, [143](#), [215](#), [216](#)
- rootedge.to.singleton (map.to.singleton), [137](#)
- rotate, [221](#)
- rotate.multi (rotateNodes), [221](#)
- rotateNodes, [221](#)
- round, [31](#), [222](#)
- roundBranches, [222](#)
- roundPhylogram, [223](#)
- rstate, [224](#)
- salamanders (anoletree), [25](#)
- sampleFrom, [225](#)
- scan, [209](#)
- scatterplot3d, [183](#)
- scores (phyl.pca), [167](#)
- setMap, [53](#), [226](#)
- Siblings, [119](#)
- sigmoidPhylogram (roundPhylogram), [223](#)
- sim.absorbing (bmPlot), [38](#)
- sim.corr, [91](#), [204](#), [227](#)
- sim.ctt (ctt), [59](#)
- sim.history, [73](#), [120](#), [157](#), [228](#), [228](#), [231](#)
- sim.Mk, [113](#), [203](#)
- sim.Mk (sim.history), [228](#)
- sim.multiCtt (ctt), [59](#)
- sim.multiMk (sim.history), [228](#)
- sim.multiTrend (bmPlot), [38](#)
- sim.ratebystate, [229](#)
- sim.rates, [228](#), [229](#), [230](#)
- sim.reflective (bmPlot), [38](#)
- simBMphylo, [232](#)
- simmap, [103](#)
- simmap (make.simap), [132](#)
- skewers, [233](#)
- splinePhylogram, [30](#)
- splinePhylogram (roundPhylogram), [223](#)
- splitEdgeColor, [234](#)
- splitplotTree, [235](#)
- splitTree, [49](#), [215](#), [216](#), [236](#)
- starTree, [49](#), [237](#)
- strahlerNumber, [238](#)
- stree, [237](#)
- summary.evol.rate.mcmc (evol.rate.mcmc), [77](#)
- summary.multiSimmap (describe.simap), [64](#)
- summary.simap (describe.simap), [64](#)
- sunfish.data (anoletree), [25](#)

sunfish.tree (anoletree), 25

threshBayes, 24, 40, 109, 239

threshDIC, 241, 242

threshState, 242

tiplabels, 125, 153, 154

tiplabels.cophylo (nodelabels.cophylo),
153

tipRotate, 55

tipRotate (minRotate), 143

to.matrix, 243

tortoise.geog (anoletree), 25

tortoise.tree (anoletree), 25

tree.grow, 243

treeSlice, 244

tropidurid.data (anoletree), 25

tropidurid.tree (anoletree), 25

untangle, 10, 72, 245

vcv.phylo, 93, 123, 149, 172, 246

vcvPhylo, 149, 153, 246

vertebrate.data (anoletree), 25

vertebrate.tree (anoletree), 25

wasp.data (anoletree), 25

wasp.trees (anoletree), 25

whale.tree (anoletree), 25

write.nexus, 249

write.simmap, 85, 135, 211, 247, 249

write.tree, 249

writeAncestors, 248

writeNexus, 249