

# Package ‘motmot’

October 13, 2022

**Type** Package

**Title** Models of Trait Macroevolution on Trees

**Version** 2.1.3

**Depends** R (>= 2.10.0), ape (>= 3.0-7)

**Date** 2019-11-21

**Author** Mark Puttick [aut, cre, cph], Gavin Thomas [aut, cph], Rob Freckleton [aut, cph], Magnus Clarke [ctb], Travis Ingram [ctb], David Orme [ctb], Emmanuel Paradis [ctb]

**Maintainer** Mark Puttick <marknputtick@gmail.com>

**Description** Functions for fitting models of trait evolution on phylogenies for continuous traits. The majority of functions described in Thomas and Freckleton (2012) <doi:10.1111/j.2041-210X.2011.00132.x> and include functions that allow for tests of variation in the rates of trait evolution.

**License** GPL (>= 2)

**Repository** CRAN

**URL** <https://puttickbiology.wordpress.com/motmot/>

**RoxygenNote** 7.0.1

**LazyData** true

**Suggests** knitr, rmarkdown, formatR

**VignetteBuilder** knitr

**LinkingTo** Rcpp

**Imports** Rcpp, coda, ks, mvtnorm, caper, methods

**Encoding** UTF-8

**SystemRequirements** C++11

**NeedsCompilation** yes

**Date/Publication** 2019-11-25 17:10:07 UTC

**R topics documented:**

motmot-package	3
addFossilToPhy	3
allopatric.data	4
anolis.data	5
anolis.tree	6
as.rateData	6
as.rateMatrix	8
blomberg.k	9
calcCutOff	10
chr.disp.lrt	11
chr.disp.param	12
chr.disp.sim	13
contemporaryPhy	15
dropTipPartial	16
fairProportions	17
finch.data	18
finch.tree	19
likRatePhylo	19
likTraitPhylo	21
mammals	22
mcmc.plot	23
ML.RatePhylo	24
node.descendants	26
nodeTimes	27
optim.likRatePhylo	28
phyloCovar	29
phyloMean	30
phyloVar	32
pic.pgls	33
plot.timeSlice.ML	35
plot.traitMedusa.model	37
RatePhylo.allCI	38
sampleShid	40
sortTraitData	41
summary.traitMedusa	42
traitData.plot	44
transformPhylo	45
transformPhylo.ll	49
transformPhylo.MCMC	54
transformPhylo.ML	57
transformPhylo.sim	63
transformRateMatrix	66

---

`motmot-package`*Models Of Trait Macroevolution On Trees*

---

**Description**

Functions for fitting models of trait evolution on phylogenies for continuous traits. The majority of functions described in Thomas and Freckleton (2012) and include functions that allow for tests of variation in the rates of trait evolution.

**Author(s)**

Mark Puttick <marknputtick@gmail.com>.

Gavin Thomas.

Travis Ingram.

Magnus Clarke.

Rob Freckleton.

David Orme.

Emmanuel Paradis.

**References**

Thomas GH, & Freckleton R. 2012. Body size diversification in Anolis: novel environments and island effects. MOTMOT: models of trait macroevolution on trees 3, 145-151.

---

`addFossilToPhy`*add a fossil to an interior branch of a time-scaled phylogeny*

---

**Description**

the function takes a time-scaled phylogeny and adds a fossil at a branch in the past

**Usage**

```
addFossilToPhy(  
  phy,  
  inGroup,  
  fossil,  
  fossilAge,  
  minLength = 0.1,  
  maxLength = NULL  
)
```

**Arguments**

phy	Input phylogeny an object of class phylo (see <b>ape</b> )
inGroup	vector of tip labels defined as the ingroup - the fossil(s) will be placed on the stem branch leading to the 'inGroups' most recent common ancestor
fossil	tip labels for the new fossil
fossilAge	age of the fossil
minLength	minimum length leading to the fossil
maxLength	maximum length leading to the fossil. If NULL (default) then the maximum bound is half the length of the branch leading to the crown node

**Value**

the the time-scaled phylogeny with the fossil attached

**Author(s)**

Mark Puttick

**References**

Puttick, M. N., Kriwet, J., Wen, W., Hu, S., Thomas, G. H., & Benton, M. J. (2017). Body length of bony fishes was not a selective factor during the biggest mass extinction of all time. *Palaeontology*, 60, 727-741.

**Examples**

```
data(anolis.tree)
plot(anolis.tree)
nodelabels(214, 214)
# add fossil to node 214
in.groups <- node.descendants(x=214, phy=anolis.tree, tip.labels=TRUE)[[2]]
fossilPhy <- addFossilToPhy(anolis.tree, in.groups, fossil="fakeFossil", fossilAge=60)
plot(fossilPhy)
```

---

allopatric.data

*Finch allopatric data*

---

**Description**

Finch data from Clarke et al. 2017

**Usage**

```
data(finches)
```

**Format**

An object of class "data.frame".

**References**

Clarke M, Thomas GH, Freckleton RP. 2017. Trait evolution in adaptive radiations: modelling and measuring interspecific competition on phylogenies. *The American Naturalist*. 189, 121-137.

**Examples**

```
data(finches)
head(allopatric.data)
```

---

anolis.data

*Anolis phenotype data*

---

**Description**

Data on anolis phenotype data from Thomas et al. 2009

**Usage**

```
data(anolis.data)
```

**Format**

An object of class "data.frame".

**References**

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. *Evolution* 63, 2017-2030.

**Examples**

```
data(anolis.data)
head(anolis.data)
```

---

anolis.tree	<i>Anolis phylogeny</i>
-------------	-------------------------

---

**Description**

Data on anolis phylogeny from Thomas et al. 2009

**Usage**

```
data(anolis.tree)
```

**Format**

An object of class "phylo".

**References**

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. *Evolution* 63, 2017-2030.

**Examples**

```
data(anolis.tree)
anolis.tree
```

---

as.rateData	<i>Conversion among data and phylogeny objects</i>
-------------	--

---

**Description**

Function to generate a "rateData" object containing the discrete explanatory variable, continuous response variable and set of variance co-variance matrices. It loads the trait data and removes species with missing data from the data and vcv matrix. `as.rateData` requires either a set of matrices in `rateMatrix` format created using `as.rateMatrix` or, if no `rateMatrix` object is input then it requires a phylogeny in "phylo" format. If a "phylo" object is used `as.rateData` will call `as.rateMatrix` internally. `as.rateMatrix` calls the "ape" function `vcv.phylo` multiple times and this can be slow for large phylogenies. It will often be more efficient to use `as.rateMatrix` first to create a "rateMatrix" object to pass to `as.rateData`, particularly if there are many response traits of interest to be fitted to the same phylogeny and set of reconstructed ancestral states.

**Usage**

```
as.rateData(
  y,
  x,
  rateMatrix = NULL,
  phy = NULL,
  data,
  meserr.col = NULL,
  meserr.propn = NULL,
  log.y = FALSE,
  report_prune = FALSE
)
```

**Arguments**

y	The response variable - typically a continuous trait. Specified as a column name or index
x	The explanatory (discrete) variable used to define the hypothesised rate categories. Specified as a column name or index.
rateMatrix	A "rateMatrix" object or NULL
phy	An object of class "phylo" (see <b>ape</b> ).
data	A data frame containing (minimally) the x and y variables as columns with species names as rownames.
meserr.col	Column name or index containing measurement error for species means.
meserr.propn	Single value specifying the proportional measurement to be applied across all species.
log.y	Logical, natural log transform response variable.
report_prune	Logical. Prints a list of dropped species if TRUE

**Value**

rateData An object of class "rateData" which is a list containing the response (y) and explanatory (x) variable along with a list of variance-covariance matrices.

**Author(s)**

Gavin Thomas

**References**

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. *Evolution* 63, 2017-2030.

**Examples**

```
## Read in phylogeny and data from Thomas et al. (2009)
data(anolis.tree)
data(anolis.data)

## Convert data to class rateData with a rateMatrix object as input
anolis.rateMatrix <- as.rateMatrix(phy=anolis.tree, x="geo_ecomorph", data=anolis.data)

anolis.rateData <- as.rateData(y="Female_SVL", x="geo_ecomorph",
rateMatrix = anolis.rateMatrix, phy=NULL, data=anolis.data, log.y=TRUE)

## Convert data to class rateData with a phylo object as input
anolis.rateData <- as.rateData(y="Female_SVL", x="geo_ecomorph",
rateMatrix = NULL, phy=anolis.tree, data=anolis.data, log.y=TRUE)
```

---

as.rateMatrix

*Conversion among data and phylogeny objects*


---

**Description**

Function to generate a rateMatrix object containing a set of variance covariance matrices. Note that as.rateMatrix calls the CAIC function vcv.array multiple times and this can be slow for large phylogenies (though faster than using the **ape** equivalent vcv.phylo).

**Usage**

```
as.rateMatrix(phy, x, data)
```

**Arguments**

phy	An object of class "phylo" (see <b>ape</b> ).
x	The explanatory (discrete) variable used to define the hypothesised rate categories. Can be specified as a column number or column name
data	The explanatory (discrete) variable used to define the hypothesised rate categories. Can be specified as a column number or column name

**Value**

rateMatrix An object of class "rateMatrix" - a list of matrices describing the expected variances and covariances of between species. Each matrix refers to the variances and covariances for a given state of x (see Thomas et al. 2006).

**Author(s)**

Gavin Thomas



## References

Thomas GH, Freckleton RP, & Szekely T. 2006. Comparative analyses of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society B* 273, 1619-1624.

## Examples

```
## Read in phylogeny and data from Thomas et al. (2009)
data(anolis.tree)
data(anolis.data)

## Convert data to class rateMatrix
anolis.rateMatrix <- as.rateMatrix(phy=anolis.tree, x="geo_ecomorph", data=anolis.data)
```

---

blomberg.k

*Blomberg's K Estimate Blomberg's K (Blomberg et al. 2003)*

---

## Description

Blomberg's K Estimate Blomberg's K (Blomberg et al. 2003)

## Usage

```
blomberg.k(phy, y)
```

## Arguments

phy            An object of class `phylo` (see **ape**).

y              A matrix of trait values.

## Value

The estimate of the K statistic

## References

Blomberg SP, Garland T, & Ives AR. 2003. Testing for phylogenetic signal in comparative data: behavioral traits are more labile. *Evolution* 57, 717-745.

## See Also

[transformPhylo.ML](#), the Picante package

---

calcCutOff                      *Calculate multiple-test cut-off*

---

### Description

Calculate the log-likelihood, AIC, or AICc cut-off necessary for type-one error to reach acceptable levels

### Usage

```
calcCutOff(
  phy,
  n = 1000,
  mc.cores = 1,
  model,
  measure = "AICc",
  alpha.error = 0.05,
  ...
)
```

### Arguments

phy	An object of class phylo (see <b>ape</b> ).
n	Number of simulations
mc.cores	Number of cores for parallel processing for linux-type systems (not applicable to Windows)
model	Evolutionary model, typically "tm1", "tm2", or "timeSlice", which is used to test the empirical data
measure	Measure used to summarise the model. One of "lnL" (log-likelihood), "AIC", or "AICc"
alpha.error	Target for the desired type-one error rate for the model (default 0.05)
...	Arguments to be passed to <a href="#">transformPhylo.ML</a> which should be identical to the model applied to empirical data

### Value

The cut-off required to produce an type-one error rate equal to `quantile.cut.off` (default = 0.05) when data are simulated under Brownian motion, and these data are analysed under the appropriate model.

### Author(s)

Mark Puttick

### See Also

[transformPhylo.ML](#), [transformPhylo.ll](#), [transformPhylo](#), [transformPhylo.MCMC](#)

**Examples**

```

data(anolis.tree)
set.seed(393)
# calculated necessary AICc cut-off to reduce type-one error to 5%
# for a timeSlice model with a split at 30Ma (only 5 simulations used,
# it's recommend to use 1000 for analyses)
calcCutOff(anolis.tree, n=5, model="timeSlice", splitTime=30)

```

chr.disp.lrt

*Character displacement likelihood ratio test***Description**

Conducts a likelihood ratio test between empirical data (phylogeny and trait data), and simulations from the function `chr.disp.sim` using an approximate Bayesian computation (ABC) approach (Clarke et al. 2017)

**Usage**

```
chr.disp.lrt(emp.tree, emp.data, param.out, posteriorSize = 500)
```

**Arguments**

<code>emp.tree</code>	An empirical phylogeny - a object of class <code>phylo</code> (see <b>ape</b> ).
<code>emp.data</code>	Continuous trait data matrix
<code>param.out</code>	simulated data from the function <code>chr.disp.sim</code>
<code>posteriorSize</code>	The number of samples to use in the likelihood-ratio test

**Value**

List containing element of 'estimates' with the estimates of sigma and a, with the Brownian motion ( $a = 0$ ) summarised in column one and the character displacement ( $a > 0$ ) in column two. 'likelihood' contains the likelihood of the Brownian motion model and the character displacement model, and the likelihood ratio test estimate. If used, there is an estimate of Blomberg's K for the empirical and simulated data.

**Author(s)**

Magnus Clarke and Mark Puttick

**References**

Clarke M, Thomas GH, Freckleton RP. 2017. Trait evolution in adaptive radiations: modelling and measuring interspecific competition on phylogenies. *The American Naturalist*. 189, 121-137.

**See Also**

[chr.disp.sim](#), [chr.disp.param](#)

## Examples

```
## import finch data form Clarke et al. (2017)
data(finches)
## simulate small amount of data
## (example only - many more datasets are required for accuracy)
param.simulation <- chr.disp.param(finch.tree, n.sim = 100, n.steps=100,
max.sigma = 8, max.a = 8, ntraits=1,
allopatry=as.matrix(allopatric.data), mc.cores = 1)
chr.disp.lrt(finch.tree, finch.data, param.simulation, 50)
```

---

chr.disp.param                      *Simulate character displacement data wrapper*

---

## Description

Simulates phylogenetic trait data under a character displacement model (Clarke et al. 2017) in which traits interact inter-specifically, with competition between sympatric lineages driving trait values apart

## Usage

```
chr.disp.param(
  phy,
  n.sim = 100,
  n.steps = 1000,
  max.sigma = 8,
  max.a = 8,
  est.blomberg.k = FALSE,
  ntraits = 1,
  sympatry = NA,
  allopatry = NA,
  trait.lim = NA,
  mc.cores = 1
)
```

## Arguments

phy	An object of class phylo (see <b>ape</b> ).
n.sim	Number of replications to simulate data
n.steps	Number of time steps the for the simulation (default = 1000 time steps).
max.sigma	The maximum value of Brownian variance in the simulation sampled from a U(0, max.sigma) distribution for each iteration
max.a	The maximum value of the strength of competition between inter-specific lineages sampled from a U(0, max.a) distribution for each iteration
est.blomberg.k	Logical. If TRUE, Blomberg's K is simultaneously estimated

ntraits	Number of traits to be simulated
sympatry	an optional matrix giving the time that each pair of species starts to interact
allopatry	an optional matrix giving the times when species stop interacting
trait.lim	an optional parameter that puts limits on the available trait-space, preventing trait values with magnitude greater than the value of lim
mc.cores	Numeric. The number of parallel cores to be used in simulations. Only applicable on Linux and Mac systems

**Value**

List containing the simulated data 'simulated.param': a matrix with each row represented an iteration, the sigma (Brownian variance) used in the iteration, the 'a' value used in each iteration, the mean and standard deviation between neighbouring trait values. The 'input.arguments' from the model, the 'input.phy' from the model, and the input 'sympatry' and 'allopatry' matrices.

**Author(s)**

Magnus Clarke and Mark Puttick

**References**

Clarke M, Thomas GH, & Freckleton RP. 2017. Trait evolution in adaptive radiations: modeling and measuring interspecific competition on phylogenies. *The American Naturalist* 189, 121-137.

**See Also**

[chr.disp.sim](#), [chr.disp.lrt](#)

**Examples**

```
## import finch data from Clarke et al. (2017)
data(finches)
## simulate small amount of data
## (example only - many more datasets are required for accuracy)
param.simulation <- chr.disp.param(finch.tree, n.sim = 3, n.steps=100,
max.sigma = 8, max.a = 8, ntraits=1,
allopatry=as.matrix(allopatric.data), mc.cores = 1)
```

---

chr.disp.sim

*Simulate character displacement data*

---

**Description**

Simulates data under a Brownian motion or character displacement model

**Usage**

```
chr.disp.sim(  
  phy,  
  n.steps = 1000,  
  sigma = 1,  
  a = 0,  
  ntraits = 1,  
  sympatry = NA,  
  allopatry = NA,  
  trait.lim = NA  
)
```

**Arguments**

phy	An object of class <code>phylo</code> (see <b>ape</b> ).
n.steps	Number of time steps the for the simulation (default = 1000 time steps).
sigma	The value of Brownian variance in the simulation
a	The strength of competition between inter-specific lineages
ntraits	Number of traits to be simulated
sympatry	an optional matrix giving the time that each pair of species starts to interact
allopatry	an optional matrix giving the times when species stop interacting
trait.lim	an optional parameter that puts limits on the available trait-space, preventing trait values with magnitude greater than the value of lim

**Value**

A list containing the the simulated data (`tval`) showing the `sigma`, `a`, mean gap and gap standard deviation. Additionally, if used, the user input `sympatry` (`symp`) and/or `allopatry` (`allo`) matrices

**Author(s)**

Magnus Clarke and Mark Puttick

**References**

Clarke M, Thomas GH, Freckleton RP. 2017. Trait evolution in adaptive radiations: modelling and measuring interspecific competition on phylogenies. *The American Naturalist*. 189, 121-137.

**See Also**

[chr.disp.param](#), [chr.disp.lrt](#)

**Examples**

```
## import finch data form Clarke et al. (2017)
data(finches)
emp.tree <- finch.tree
emp.data <- finch.data
## simulate small amount of data
## (example only - many more datasets are required for accuracy)
sim.data <- chr.disp.sim(emp.tree, n.steps=100,
sigma=1, a=2, ntraits=1, sympatry=NA, allopatry=NA, trait.lim=NA)
```

---

contemporaryPhy      *prune tree and data to lineages present in a time bin in the past*

---

**Description**

the function takes a full tree and returns a pruned phylogeny with only tips and lineages found within a time bin preserved. If trait data are supplied the function will return tip states based either on the original tips found in the bin, or tip states inferred from ancestral states

**Usage**

```
contemporaryPhy(
  phy,
  maxBin,
  minBin,
  reScale = 0,
  allTraits,
  closest.min = TRUE,
  traits.from.tip = TRUE
)
```

**Arguments**

phy	An object of class phylo (see <b>ape</b> ).
maxBin	the start age (older time in myr from present) of the time bin in which lineages are preserved
minBin	the final age (younger time in myr from present) of the time bin in which lineages are preserved
reScale	if the most recent tip is not from the present, the age needed to add so the phylogeny is in 'real time'
allTraits	a vector of trait data corresponding to the phy\$edge object. The trait data represent tip and internal node data for the phylogeny
closest.min	Logical. Should new tip values for lineages that span the bin be taken from the node nearest the minBin age (closest.min=TRUE, default) or the maxBin age (closest.min=FALSE)
traits.from.tip	Logical. Should tip values for pendant edges in the bin be taken from the original tip value or the reconstructed node value (if it is closer than the tip value)

**Value**

the pruned phylogeny. The object descendants refers to the lineages the branch in the time bin gave rise to before it was pruned. If traits are included a vector of trait values representing species at the tips.

**Author(s)**

Mark Puttick

**References**

Puttick, M. N., Kriwet, J., Wen, W., Hu, S., Thomas, G. H., & Benton, M. J. (2017). Body length of bony fishes was not a selective factor during the biggest mass extinction of all time. *Palaeontology*, 60, 727-741.

**Examples**

```
## prune a random tree to taxa present between 4 and 2 units before present
# generate tree
set.seed(20)
tree <- rtree(20)
# generate traits
traits <- rnorm(20)
# plot tree and timeframe
plot(tree)
max.age <- nodeTimes(tree)[1,1]
abline(v=max.age - c(4, 2))
# prune tree to timeframe
cont.tree <- contemporaryPhy(phy=tree, maxBin=4, minBin=2, allTraits=traits)
plot(cont.tree$phy)
```

---

dropTipPartial

*Drop tips from a phylogenetic tree while preserving deleted nodes*

---

**Description**

Wrapper for the ape function `drop.tip` that preserves the number of nodes affecting each branch. For use with the `psi` and `multipsi` models.

**Usage**

```
dropTipPartial(phy, tip)
```

**Arguments**

phy	Phylogenetic tree in phylo format
tip	A vector of mode numeric or character specifying the tips to delete, to be passed to <code>drop.tip</code>



**Value**

Phylogenetic tree in phylo format, with an added element `Shid`, a vector of numbers of observed but "missing" speciation events per branch, in the same order as the branches in the phylo object

**Author(s)**

Travis Ingram

**References**

Ingram, T. 2011. Speciation along a depth gradient in a marine adaptive radiation. *Proc. R. Soc. B* 278: 613-618.

**See Also**

[transformPhylo.ML](#)

**Examples**

```
## Read in phylogeny and data from Thomas et al. (2009)
data(anolis.tree)
data(anolis.data)
## identify tips to drop
tips.to.go <- anolis.tree$tip.label[1:30]
dropTipPartial(phy=anolis.tree, tip=tips.to.go)
```

---

fairProportions

*Calculate fair proportions phylogenetic diversity metric*

---

**Description**

Calculate fair proportions phylogenetic diversity metric Note that `as.rateMatrix` calls the CAIC function `vcv.array` multiple times and this can be slow for large phylogenies (though faster than using the "ape" equivalent `vcv.phylo`).

**Usage**

```
fairProportions(phy, nodeCount = FALSE)
```

**Arguments**

`phy` An object of class `phylo` (see **ape**).

`nodeCount` Logical - should root to tip node counts be returned (default is FALSE)

**Value**

Returns a matrix of fair proportion for all tips in phylogeny and node counts if selected.

**Author(s)**

Gavin Thomas

**References**

Redding, D.W. and Mooers, A.O. (2006). Incorporating evolutionary measures into conservation prioritisation. *Conservation Biology*, 20, 1670-1678.

Isaac, N.J.B., Turvey, S.T., Collen, B., Waterman, C. and Baillie, J.E.M. (2007). Mammals on the EDGE: conservation priorities based on threat and phylogeny. *PLoS ONE*, 2, e296.

**Examples**

```
data(anolis.tree)

fp <- fairProportions(anolis.tree)
fpNodes <- fairProportions(anolis.tree, nodeCount=TRUE)
```

---

finch.data

*Finch phenotype data*

---

**Description**

Finch data from Clarke et al. 2017

**Usage**

```
data(finches)
```

**Format**

An object of class "data.frame".

**References**

Clarke M, Thomas GH, Freckleton RP. 2017. Trait evolution in adaptive radiations: modelling and measuring interspecific competition on phylogenies. *The American Naturalist*. 189, 121-137.

**Examples**

```
data(finches)
head(finch.data)
```

---

finch.tree	<i>Finch tree</i>
------------	-------------------

---

**Description**

Finch data from Clarke et al. 2017

**Usage**

```
data(finches)
```

**Format**

An object of class "phylo"

**References**

Clarke M, Thomas GH, Freckleton RP. 2017. Trait evolution in adaptive radiations: modelling and measuring interspecific competition on phylogenies. *The American Naturalist*. 189, 121-137.

**Examples**

```
data(finches)
head(finch.tree)
```

---

likRatePhylo	<i>Log-likelihood rate estimation for traits and phylogenies</i>
--------------	--

---

**Description**

This function calculates the log-likelihood, phylogenetic mean, and Brownian variance for a trait and a phylogeny transformed according to variation in relative rates.

**Usage**

```
likRatePhylo(
  rateData,
  rate = NULL,
  common.mean = FALSE,
  lambda.est = TRUE,
  lambda = 1,
  meserr = FALSE,
  sigmaScale = NULL
)
```

**Arguments**

rateData	an object of class rateData
rate	a vector of relative rate parameters. The length of the vector is equal to the number of rates being estimated. If rate=NULL then rates are equal.
common.mean	a logical specifying whether each rate category should have its own mean (common.mean=FALSE) or all categories should have the same mean (common.mean=TRUE). See Thomas et al. (2009) for a discussion on the impact of assumptions about mean on rate estimates.#'
lambda.est	Logical. Fit Pagel's lambda.
lambda	Logical. Numeric value for lambda from 0-1.
meserr	Logical. Logical. Include measurement error.
sigmaScale	Logical. Scalar for measurement error relative to tree.

**Value**

ll log-likelihood of the model  
 mu phylogenetically corrected mean(s)  
 s2 Brownian variance

**Note**

The means are output as treatment contrasts.

**Author(s)**

Gavin Thomas, Rob Freckleton

**References**

Thomas GH, Freckleton RP, & Szekely T. 2006. Comparative analyses of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society B* 273, 1619-1624.

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in *Anolis*: novel environments and island effects. *Evolution* 63, 2017-2030.

**Examples**

```
data(anolis.tree)
data(anolis.data)

## Convert data to class rateData with a rateMatrix object as input
anolis.rateMatrix <- as.rateMatrix(phy=anolis.tree, x="geo_ecomorphy", data=anolis.data)

anolis.rateData <- as.rateData(y="Female_SVL", x="geo_ecomorphy",
rateMatrix = anolis.rateMatrix, phy=NULL, data=anolis.data, log.y=TRUE)

## Calculate phylogenetic mean, variance, log likelihood for a model where the first
```

```
# mean only
phyloMean(rateData=anolis.rateData, rate = c(1,2,0.1,1), common.mean = FALSE)

# variance only
phyloVar(rateData=anolis.rateData, rate = c(1,2,0.1,1), common.mean = FALSE)

# mean, variance and log-likelihood
likRatePhylo(rateData=anolis.rateData, rate = c(1,2,0.1,1), common.mean = FALSE)
```

---

likTraitPhylo                      *Log-likelihood estimation for traits and phylogenies*

---

## Description

This function calculates the log-likelihood and Brownian (co)variance for a trait(s) and a phylogeny using phylogenetically independent contrasts. Note that `as.rateMatrix` calls the CAIC function `vcv.array` multiple times and this can be slow for large phylogenies (though faster than using the "ape" equivalent `vcv.phylo`).

## Usage

```
likTraitPhylo(y, phy, covPIC = TRUE, brCov = NULL)
```

## Arguments

<code>y</code>	A matrix of trait data. Rownames must be included and match the taxon names in the phylogeny. Can accept single traits (calculates variance) or multiple traits (calculates variance-covariance matrix).
<code>phy</code>	An object of class "phylo" (see <b>ape</b> ).
<code>covPIC</code>	Logical - allow for covariance between multivariate traits (TRUE), or assume not covariance (FALSE). Only applicable to multivariate traits
<code>brCov</code>	If NULL (the default), Brownian covariance is analytically estimated. If a user-supplied numerical value is supplied the likelihood is calculate given this value

## Details

The `phylo` object must be rooted and fully dichotomous

## Value

`brownianVariance` Brownian variance (or covariance for multiple traits) given the data and phylogeny

`logLikelihood` The log-likelihood of the model and data

## Author(s)

Gavin Thomas, Rob Freckleton

**References**

Felsenstein J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous characters. *Am. J. Hum. Genet.* 25, 471-492.

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

Freckleton RP & Jetz W. 2009. Space versus phylogeny: disentangling phylogenetic and spatial signals in comparative data. *Proc. Roy. Soc. B* 276, 21-30.

**Examples**

```
data(anolis.tree)
data(anolis.data)
## calculate Brownian variance log-likelihood of female SVL
female.svl <- matrix(anolis.data[, "Female_SVL"],
dimnames=list(rownames(anolis.data)))
input.data <- sortTraitData(phy=anolis.tree, y=female.svl, log.trait=TRUE)
likTraitPhylo(phy = input.data$phy, y=input.data$trait)
```

---

mammals

*Mammal data*


---

**Description**

Mammal data from Slater 2013

**Usage**

```
data(mammals)
```

**Format**

An list containing mass and errors of mammal body mass and mammal phylogeny of class "phylo"

**References**

Phylogenetic evidence for a shift in the mode of mammalian body size evolution at the Cretaceous-Palaeogene boundary. *Methods in Ecology and Evolution.* 4: 734-744.

**Examples**

```
data(mammals)
```

---

`mcmc.plot`*plot the output from transformPhylo.MCMC*

---

### Description

Plots a histogram of the estimated parameter and a trace of the results

### Usage

```
mcmc.plot(  
  mcmc.input,  
  y.limit = NULL,  
  x.limit = NULL,  
  label.text = NULL,  
  cex.axis = 1,  
  cex.labels = 0.7,  
  col.hist = "green4",  
  col.trace = "navy"  
)
```

### Arguments

<code>mcmc.input</code>	an object of class "mcmc.motmot" output from <a href="#">transformPhylo.MCMC</a>
<code>y.limit</code>	the limits for the y axes for the plots
<code>x.limit</code>	the limits for the x axes for the plots
<code>label.text</code>	the labels for the two plots defaults to '(a)' etc., for the histogram and '(b)' etc., for the trace plot
<code>cex.axis</code>	character expansion for the plot axis labels
<code>cex.labels</code>	character expansion for the plot axis names
<code>col.hist</code>	colour for the histogram bars
<code>col.trace</code>	colour for the trace plot

### Value

Two plots showing the histogram of the estimated parameter value and a trace of the MCMC estimation

### Author(s)

Mark Puttick

**Examples**

```

library(motmot)
data(anolis.tree)
data(anolis.data)
attach(anolis.data)
male.length <- matrix(Male_SVL, dimnames=list(rownames(anolis.data)))
sortedData <- sortTraitData(anolis.tree, male.length)
phy <- sortedData$phy
male.length <- sortedData$trait
phy.clade <- extract.clade(phy, 182)
male.length.clade <- as.matrix(male.length[match(phy.clade$tip.label, rownames(male.length)),])
## not run
# please note, this model will be need to run for longer to achieve convergence
# lambda.mcmc <- transformPhylo.MCMC(y=male.length.clade, phy=phy.clade,
# model="lambda", mcmc.iteration=100, burn.in=0.1)
# mcmc.plot(lambda.mcmc)

```

---

ML.RatePhylo

*Maximum likelihood rate estimation for traits and phylogenies*


---

**Description**

Full function for maximum likelihood estimation of rate parameters and comparison to a single rate model.

**Usage**

```

ML.RatePhylo(
  rateData,
  rate = NULL,
  fixed = NULL,
  pretty = TRUE,
  rateMIN = 0.001,
  rateMAX = 50,
  common.mean = FALSE,
  lambda.est = TRUE,
  est.CI = FALSE,
  meserr = FALSE,
  file = NULL
)

```

**Arguments**

`rateData` an object of class `rateData`

`rate` a vector of relative rate parameters. The length of the vector is equal to the number of rates being estimated. If `rate=NULL` then rates are equal.



<code>fixed</code>	A vector stating whether each parameter should be allowed to vary (either FALSE which results in a start value of 1, or a numeric start value) or should be fixed (TRUE).
<code>pretty</code>	Display the output nicely ( <code>pretty=TRUE</code> ) or as a list ( <code>pretty=FALSE</code> )
<code>rateMIN</code>	Minimum value for the rate parameters.
<code>rateMAX</code>	Maximum value for the rate parameters
<code>common.mean</code>	a logical specifying whether each rate category should have its own mean ( <code>common.mean=FALSE</code> ) or all categories should have the same mean ( <code>common.mean=TRUE</code> ). See Thomas et al. (2009) for a discussion on the impact of assumptions about mean on rate estimates.
<code>lambda.est</code>	Logical. Estimate Pagel's lambda.
<code>est.CI</code>	Logical. Estimate approximate confidence intervals for rate parameters.
<code>meserr</code>	Logical. Incorporate measurement error.
<code>file</code>	File string for output. Only used if <code>pretty=TRUE</code> .

### Value

If `pretty=FALSE`, returns a list containing:

- MLRate Maximum likelihood estimates of the rate parameters
- Lambda Maximum likelihood estimate of lambda
- LCI Approximate lower confidence intervals for rate
- UCI Approximate upper confidence intervals for rate parameters
- means Means for each category
- nParam Number of parameters in the model (how many means and rate categories)
- Max.lik Maximum (log) likelihood
- AIC for maximum likelihood model
- AICc for maximum likelihood model
- LambdaSingle Maximum likelihood estimate of lambda for the single rate model
- Lik1 Likelihood of the equivalent single rate model
- Likelihood ratio statistic of "Max.lik" vs "Lik1"
- P P values for the LR statistic
- df Degrees of freedom for the LR statistic
- AIC.rate1 AIC for single rate model
- AICc.rate1 AICc for single rate model

If `pretty=TRUE`, prints a nice version of the list to screen. If `file` is specified the pretty output will be sent to file, not the console.

### Note

Unlike `phyloMean` and `likRatePhylo` (that use treatment contrasts), the means reported here are the actual values

**Author(s)**

Gavin Thomas, Rob Freckleton

**References**

Thomas GH, Freckleton RP, & Szekely T. 2006. Comparative analyses of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society B* 273, 1619-1624.

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. *Evolution* 63, 2017-2030.

**Examples**

```
## Read in phylogeny and data from Thomas et al. (2009)
data(anolis.tree)
data(anolis.data)
## Convert data to class rateData with a rateMatrix object as input
anolis.rateMatrix <- as.rateMatrix(phy=anolis.tree, x="geo_ecomorph",
data=anolis.data)
anolis.rateData <- as.rateData(y="Female_SVL", x="geo_ecomorph",
rateMatrix = anolis.rateMatrix, phy=NULL, data=anolis.data, log.y=TRUE)
# A model with a different rate in one of the four groups. The 'fixed' command is used to determine
# whether a particular rate is to be constrained or not. Use '1' to fix a group and 'FALSE' to show
# that the parameter is not fixed and should be estimated. The values should be entered in the same
# order as the ranking of the groups. That is, group 0 (small islands) takes position one in the
# fixed vector, group 1 (large island trunk crown and trunk ground) takes position 2 and so on.
# The default is to allow each group to take a different mean.
ML.RatePhylo(anolis.rateData, fixed=c(1, FALSE, FALSE, FALSE), pretty=TRUE, lambda.est = FALSE)
```

---

node.descendants	<i>Identify nodes and tips descended from a node</i>
------------------	--

---

**Description**

Obtains a vector of the tips and nodes subtending from a node in a phylogeny.

**Usage**

```
node.descendants(x, phy, tip.labels = FALSE)
```

**Arguments**

x	A positive integer
phy	An object of class phylo (see <b>ape</b> ).
tip.labels	Logical - output tip.labels?

**Details**

This function is stolen from the `clade.members` function in the `CAIC` package but returns both node and tip id's.

**Value**

Returns a vector of node and tip ids descended from the tip(s) "x". If `tip.labels=TRUE` then returns a list of node ids and tip labels.

**Note**

`as.rateMatrix` calls the `CAIC` function `vcv.array` multiple times and this can be slow for large phylogenies (though faster than using the **ape** equivalent `vcv.phylo`).

**Author(s)**

Gavin Thomas, David Orme

---

nodeTimes	<i>Get times for nodes and tips</i>
-----------	-------------------------------------

---

**Description**

Produces branching and tip times for ultrametric and non-ultrametric trees

**Usage**

```
nodeTimes(phy)
```

**Arguments**

`phy` An object of class `phylo` (see **ape**).

**Value**

Returns a matrix corresponding the `phy` "edge" matrix showing internal and external node times

**Note**

`nodeTimes` is essentially a re-written version of the **ape** `branching.times`.

**Author(s)**

Mark Puttick, Emmanuel Paradis

**Examples**

```
## Read in phylogeny from Thomas et al. (2009)
data(anolis.tree)
anolis.node.times <- nodeTimes(phy=anolis.tree)
```

---

optim.likRatePhylo      *Maximum likelihood rate estimation for traits and phylogenies*

---

### Description

Function for the maximum likelihood estimation of rate parameters on a trait and phylogeny.

### Usage

```
optim.likRatePhylo(
  rateData,
  rate = NULL,
  fixed = NULL,
  rateMIN = 0.001,
  rateMAX = 50,
  common.mean = FALSE,
  lambda.est = TRUE,
  meserr = FALSE
)
```

### Arguments

rateData	an object of class rateData
rate	a vector of relative rate parameters. The length of the vector is equal to the number of rates being estimated. If rate=NULL then rates are equal.
fixed	A vector stating whether each parameter should be allowed to vary (either FALSE which results in a start value of 1, or a numeric start value) or should be fixed (TRUE).
rateMIN	Minimum value for the rate parameters
rateMAX	Maximum value for the rate parameters
common.mean	a logical specifying whether each rate category should have its own mean (common.mean=FALSE) or all categories should have the same mean (common.mean=TRUE). See Thomas et al. (2009) for a discussion on the impact of assumptions about mean on rate estimates.
lambda.est	Logical. Fit Pagel's lambda.
meserr	Logical. Include measurement error.

### Value

MLRate Maximum likelihood estimates of the rate parameters  
 Max.lik Maximum (log) likelihood  
 AIC AIC for maximum likelihood model  
 AICc AICc for maximum likelihood model  
 convergence convergence value from optim  
 n.parameters Number of parameters in the model (how many means and rate categories)

**Author(s)**

Gavin Thomas

**References**

Thomas GH, Freckleton RP, & Szekely T. 2006. Comparative analyses of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society B* 273, 1619-1624.

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. *Evolution* 63, 2017-2030.

**Examples**

```
data(anolis.tree)
data(anolis.data)

## Convert data to class rateData with a rateMatrix object as input
anolis.rateMatrix <- as.rateMatrix(phy=anolis.tree, x="geo_ecomorph", data=anolis.data)

anolis.rateData <- as.rateData(y="Female_SVL", x="geo_ecomorph",
rateMatrix = anolis.rateMatrix, phy=NULL, data=anolis.data, log.y=TRUE)

# A model with a different rate in each of the four groups. The 'fixed' command is used to determine
# whether a particular rate is to be constrained or not. Use '1' to fix a group and 'FALSE' to show
# that the parameter is not fixed and should be estimated. The values should be entered in the same
# order as the ranking of the groups. That is, group 0 (small islands) takes position one in the
# fixed vector, group 1 (large island trunk crown and trunk ground) takes position 2 and so on.
# The default is to allow each group to take a different mean.

optim.likRatePhylo(anolis.rateData, rate=c(1,1,1,1), common.mean=TRUE, lambda.est=FALSE)
```

---

phyloCovar

---

*Calculation of Brownian (co)variance using independent contrasts.*


---

**Description**

Calculates the Brownian variance (single trait) or variance-covariance matrix (multiple traits) using phylogenetically independent contrasts.

**Usage**

```
phyloCovar(x, phy, estimator = "unbiased")
```

**Arguments**

x	A continuous trait
phy	An object of class phylo (see <b>ape</b> ).
estimator	Should Brownian variance (or covariance) be based on the unbiased ("unbiased" - default) or maximum likelihood ("ML") estimator.

**Value**

brownianVariance Brownian variance (or covariance for multiple traits) given the data and phylogeny

**Author(s)**

Gavin Thomas, Rob Freckleton

**References**

Felsenstein J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous characters. *Am. J. Hum. Genet.* 25, 471-492.

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

Freckleton RP & Jetz W. 2009. Space versus phylogeny: disentangling phylogenetic and spatial signals in comparative data. *Proc. Roy. Soc. B* 276, 21-30.

**Examples**

```
data(anolis.tree)
data(anolis.data)
## calculate Brownian variance of female SVL
female.svl <- matrix(anolis.data[, "Female_SVL"],
dimnames=list(rownames(anolis.data)))
input.data <- sortTraitData(phy=anolis.tree, y=female.svl, log.trait=TRUE)
phyloCovar(x=input.data$trait, phy = input.data$phy)
```

---

phyloMean

*Calculation of phylogenetically corrected mean.*

---

**Description**

This function calculates the phylogenetic mean of the data given the tree and model of evolution

**Usage**

```
phyloMean(
  rateData,
  rate = NULL,
  common.mean = FALSE,
  lambda.est = TRUE,
  lambda = 1,
  meserr = FALSE
)
```

**Arguments**

rateData	an object of class rateData
rate	a vector of relative rate parameters. The length of the vector is equal to the number of rates being estimated. If rate=NULL then rates are equal.
common.mean	a logical specifying whether each rate category should have its own mean (common.mean=TRUE) or all categories should have the same mean (common.mean=FALSE). See Thomas et al. (2009) for a discussion on the impact of assumptions about mean on rate estimates.
lambda.est	Logical. Fit Pagel's lambda.
lambda	Numeric value for lambda from 0-1.
meserr	Logical. Include measurement error.

**Value**

mu phylogenetically corrected mean

**Note**

The means are output as treatment contrasts.

**Author(s)**

Gavin Thomas, Rob Freckleton

**References**

Thomas GH, Freckleton RP, & Szekely T. 2006. Comparative analyses of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society B* 273, 1619-1624.

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. *Evolution* 63, 2017-2030.

**Examples**

```
## Read in phylogeny and data from Thomas et al. (2009)
data(anolis.tree)
data(anolis.data)

## Convert data to class rateData with a rateMatrix object as input
anolis.rateMatrix <- as.rateMatrix(phy=anolis.tree, x="geo_ecomorph", data=anolis.data)

anolis.rateData <- as.rateData(y="Female_SVL", x="geo_ecomorph",
rateMatrix = anolis.rateMatrix, phy=NULL, data=anolis.data, log.y=TRUE)

# A model with a different rate in each of the four groups. The 'fixed' command is used to determine
# whether a particular rate is to be constrained or not. Use '1' to fix a group and 'FALSE' to show
# that the parameter is not fixed and should be estimated. The values should be entered in the same
# order as the ranking of the groups. That is, group 0 (small islands) takes position one in the
```

```
# fixed vector, group 1 (large island trunk crown and trunk ground) takes position 2 and so on.
# The default is to allow each group to take a different mean.

phyloMean(anolis.rateData, rate=c(1,1,1,1), common.mean=FALSE)
# common mean for all groups
phyloMean(anolis.rateData, rate=c(1,1,1,1), common.mean=TRUE)
```

---

phyloVar

*Calculation of Brownian variance.*


---

### Description

This function calculates the phylogenetic variance (Brownian variance, or rate) of the data given the tree and model of evolution

### Usage

```
phyloVar(
  rateData,
  rate = NULL,
  common.mean = FALSE,
  lambda.est = TRUE,
  lambda = 1,
  meserr = FALSE
)
```

### Arguments

rateData	an object of class rateData
rate	a vector of relative rate parameters. The length of the vector is equal to the number of rates being estimated. If rate=NULL then rates are equal.
common.mean	a logical specifying whether each rate category should have its own mean (common.mean=TRUE) or all categories should have the same mean (common.mean=FALSE). See Thomas et al. (2009) for a discussion on the impact of assumptions about mean on rate estimates.
lambda.est	Logical. Fit Pagel's lambda.
lambda	Numeric value for lambda from 0-1.
meserr	Logical. Include measurement error.

### Value

phylo.var phylogenetic variance (Brownian variance)

### Author(s)

Gavin Thomas, Rob Freckleton



## References

Thomas GH, Freckleton RP, & Szekely T. 2006. Comparative analyses of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society B* 273, 1619-1624. @references Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in *Anolis*: novel environments and island effects. *Evolution* 63, 2017-2030.

## Examples

```
## Read in phylogeny and data from Thomas et al. (2009)
data(anolis.tree)
data(anolis.data)

## Convert data to class rateData with a rateMatrix object as input
anolis.rateMatrix <- as.rateMatrix(phy=anolis.tree, x="geo_ecomorph", data=anolis.data)

anolis.rateData <- as.rateData(y="Female_SVL", x="geo_ecomorph",
rateMatrix = anolis.rateMatrix, phy=NULL, data=anolis.data, log.y=TRUE)

# A model with a different rate in each of the four groups. The 'fixed' command is used to determine
# whether a particular rate is to be constrained or not. Use '1' to fix a group and 'FALSE' to show
# that the parameter is not fixed and should be estimated. The values should be entered in the same
# order as the ranking of the groups. That is, group 0 (small islands) takes position one in the
# fixed vector, group 1 (large island trunk crown and trunk ground) takes position 2 and so on.
# The default is to allow each group to take a different mean.

phyloVar(anolis.rateData, rate=c(1,2,1,1), common.mean=FALSE)
# common mean for all groups
phyloVar(anolis.rateData, rate=c(1,2,1,1), common.mean=TRUE)
```

---

pic.pgls

*Fast PLGS estimation based on contrasts*

---

## Description

Estimates regression parameters for a phylogenetic generalised least-squares analysis using the fast contrasts method (Felsenstein 1973; 1985; Freckleton 2012). This implementation is applicable for continuous traits only and not factors

## Usage

```
pic.pgls(
  formula,
  phy,
  y,
  lambda = "ML",
  return.intercept.stat = FALSE,
  meserr = NULL
)
```

**Arguments**

formula	A model formula with continuous variables only
phy	An object of class phylo (see <b>ape</b> ).
y	A matrix of trait values with rownames corresponding to the phy tip labels, and column names corresponding to the formula variable names
lambda	Default is "ML" meaning the phylogenetic signal of the response variable will be estimated using restricted Maximum likelihood. If a numeric value between 0-1 is provided this will be used in the calculation of regression coefficients
return.intercept.stat	Logical. If TRUE the standard errors, t value, and p value of the estimated Intercept is provided for comparison with output from <b>pgls</b> from <b>caper</b> etc.,. Default is FALSE as this slows the function as it involves constructing and calculating the inverse of the phy variance-covariance matrix, and based on contrasts the design matrix column of ones would have zero contrasts.
meserr	A vector (or matrix) of measurement error for each tip. This is only applicable to univariate analyses.

**Value**

A list containing the model, model summary, intercept, estimate of Lambda, model log-Likelihood, model AIC

**Author(s)**

Mark N Puttick and Rob Freckleton

**See Also**

[pgls](#)

**Examples**

```
# Data and phylogeny
data(anolis.tree)
anolis.tree$node.label <- NULL
lm.data <- transformPhylo.sim(phy=anolis.tree, n=2, model="bm")
dat <- data.frame(x = lm.data[,1], y = lm.data[,2],
names = anolis.tree$tip, row.names = anolis.tree$tip)
picModel <- pic.pgls(formula=y ~ x,
phy=anolis.tree, y = dat, lambda=1,
return.intercept.stat=FALSE)
```

---

plot.timeSlice.ML      *Identify shifts in the rate of trait diversification through time*

---

### Description

Summarises phenotypic rate variation on phylogenies through

### Usage

```
## S3 method for class 'timeSlice.ML'
plot(
  x,
  ...,
  cutoff = 4,
  AICc = TRUE,
  lowerBound = 1e-08,
  upperBound = 1000,
  phylo.plot = TRUE,
  colour.ramp = c("blue", "red"),
  cex.plot = 1,
  model.average = FALSE
)
```

### Arguments

x	Output of a timeSlice analysis in <a href="#">transformPhylo.ML</a>
...	Other functions to pass to <a href="#">plot.phylo</a>
cutoff	Value for differences in AIC scores when comparing models. More complex models with an AIC score more than this number of units lower than simpler models are retained (as per runMedusa in the geiger package).
AICc	If TRUE, AICc is used instead of AIC.
lowerBound	Minimum value for parameter estimates.
upperBound	Maximum value for parameter estimates.
phylo.plot	Logical. If TRUE, the phylogeny is plotted
colour.ramp	The colours signifying different rates from low (first colour) to high (second colour)
cex.plot	Character expansion for the plot of rates through time
model.average	Logical only applicable to "timeSlice" models. Will return the model averaged timeSlice for models in which multiple shifts were considered (i.e, when splitTime is NULL). If TRUE, the function returns a plot showing the relative weights of each shift time and the model-averaged rates through time that are weighted by their relative weights. If TRUE, plot.phylo is ignored.

**Details**

This functions summarises the output of a "timeSlice" model in [transformPhylo.ML](#) (see below). The best overall model is chosen based on AIC (or AICc if AICc=TRUE). The cut-off point for improvement in AIC score between successively more complex models can be defined using cutoff. The default cutoff is 4 but this is somewhat arbitrary and a "good" cut-off may well vary between data sets so it may well be worth exploring different cutoffs.

**Value**

ModelFit Summary of the best optimal rate shift model or the model average of each split time (if model averaging was used).

Rates Summary of the rate parameters from the best rate shift model or the model averaged rates through time.

optimalTree A phylo object with branch lengths scaled relative to rate and a plot of estimated rates through time with their associated CIs.

**Author(s)**

Mark Puttick

**References**

To Add

**See Also**

[transformPhylo.ML](#)

**Examples**

```
data(anolis.tree)
data(anolis.data)
attach(anolis.data)
male.length <- matrix(Male_SVL, dimnames=list(rownames(anolis.data)))
sortedData <- sortTraitData(anolis.tree, male.length)
phy <- sortedData$phy
male.length <- sortedData$trait
phy.clade <- extract.clade(phy, 182)
male.length.clade <- as.matrix(male.length[match(phy.clade$tip.label,
rownames(male.length)),])
timeSlice.10.ml <- transformPhylo.ML(y=male.length.clade, phy=phy.clade, model="timeSlice",
splitTime=c(10))
outputSummary <- plot(timeSlice.10.ml, cutoff=0.001, cex=0.5,
colour.ramp=c("blue", "red"))
```

---

plot.traitMedusa.model

*Tree plotting for rates*


---

## Description

Plots trees with colours based on rates of trait evolution. Also provides simple coloured plotting for trait values using the [ace](#) function in the **ape** library.

## Usage

```
## S3 method for class 'traitMedusa.model'
plot(x, y = NULL, ..., reconType = "rates", palette = "hotspot.colors")
```

## Arguments

x	Output from <a href="#">summary.traitMedusa</a> .
y	A matrix of trait values.
...	Other functions to pass to <a href="#">plot.phylo</a>
reconType	Colour branches according to rate shifts ("rates" - requires traitMedusaObject) or ancestral state reconstruction ("picReconstruction" - requires x).
palette	Defines the colour scheme with four options: hotspot.colors (red to blue), heat.colors (yellow to red), cool.colors (blues), combi.colors (yellows to reds and blues)

## Value

Returns a data frame of colours used in plot along with rate (or ancestral state) range for each colour.

## Author(s)

Gavin Thomas, Mark Puttick

## See Also

[transformPhylo.ML](#), [summary.traitMedusa](#).

## Examples

```
# Data and phylogeny
data(anolis.tree)
data(anolis.data)

# female SVL data
female.svl <- matrix(anolis.data[, "Female_SVL"],
  dimnames=list(rownames(anolis.data)))
input.data <- sortTraitData(phy=anolis.tree, y=female.svl, log.trait=TRUE)
```

```
# arbitrarily reduce data size for speed in this example
phy.clade <- extract.clade(input.data[[1]], 182)
male.length.clade <- as.matrix(input.data[[2]][match(input.data[[1]]$tip.label,
rownames(input.data[[2]])),])
# Identify rate shifts and print and plot results with up to one rate shifts
# and minimum clade size of 10.
anolisSVL_MEDUSA <- transformPhylo.ML(male.length.clade, phy=phy.clade,
model="tm1",minCladeSize=10, nSplits=1)
anolisSVL_MEDUSA_out <- summary(anolisSVL_MEDUSA, cutoff=1, AICc=FALSE)
colours <- plot(x = anolisSVL_MEDUSA_out,
reconType = "rates", type = "fan", cex=0.6, edge.width=3)
```

---

RatePhylo.allCI

*Confidence intervals for rate parameters*


---

### Description

Calculates approximate confidence intervals for all rate parameters. CIs are estimated for one rate parameters while fixing others at a given value (usually the maximum likelihood estimate).

These are reliable (given the asymptotic assumptions of the chi-square distribution) if only two groups are being compared but should be regarded only as a rough approximation for =>3 different rate categories. If the rates are correlated the CIs may be underestimated.

### Usage

```
RatePhylo.allCI(
  rateData,
  MLrate = NULL,
  fixed = NULL,
  rateMIN = 0.001,
  rateMAX = 50,
  common.mean = FALSE,
  lambda.est = TRUE
)
```

### Arguments

rateData	an object of class rateData
MLrate	a vector of relative rate parameters. The length of the vector is equal to the number of rates being estimated. If rate=NULL then rates are equal. Normally these will be the maximum likelihood rate estimates.
fixed	A vector stating whether each parameter should be allowed to vary (either FALSE which results in a start value of 1, or a numeric start value) or should be fixed (TRUE).
rateMIN	Minimum value for the rate parameters
rateMAX	Maximum value for the rate parameters

common.mean	a logical specifying whether each rate category should have its own mean (common.mean=TRUE) or all categories should have the same mean (common.mean=FALSE). See Thomas et al. (2009) for a discussion on the impact of assumptions about mean on rate estimates.
lambda.est	Logical. Estimate Pagel's lambda.

### Value

rateLci Lower confidence interval for rate estimate  
rateUci Upper confidence interval for rate estimate

### Author(s)

Gavin Thomas, Rob Freckleton

### References

Thomas GH, Freckleton RP, & Szekely T. 2006. Comparative analyses of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society B* 273, 1619-1624.

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in *Anolis*: novel environments and island effects. *Evolution* 63, 2017-2030.

### Examples

```
data(anolis.data)
data(anolis.tree)
## Convert data to class rateData with a rateMatrix object as input
anolis.rateMatrix <- as.rateMatrix(phy=anolis.tree, x="geo_ecomorph", data=anolis.data)

anolis.rateData <- as.rateData(y="Female_SVL", x="geo_ecomorph",
rateMatrix = anolis.rateMatrix, phy=NULL, data=anolis.data, log.y=TRUE)

# A model with a different rate in each of the four groups. The 'fixed' command is used to determine
# whether a particular rate is to be constrained or not. Use '1' to fix a group and 'FALSE' to show
# that the parameter is not fixed and should be estimated. The values should be entered in the same
# order as the ranking of the groups. That is, group 0 (small islands) takes position one in the
# fixed vector, group 1 (large island trunk crown and trunk ground) takes position 2 and so on. The
# default is to allow each group to take a different mean.

anole.ML <- optim.likRatePhylo(rateData=anolis.rateData, rate=NULL,
fixed=c(FALSE,FALSE,FALSE, FALSE),
common.mean=FALSE, lambda.est=FALSE)

# Confidence intervals for the first two parameters

RatePhylo.allCI(rateData=anolis.rateData, MLrate = anole.ML$MLRate,
fixed=c(FALSE, TRUE, TRUE, TRUE), rateMIN = 0.001, rateMAX = 50,
common.mean = FALSE)
```

sampleShid

*Sample hidden speciation events along branches of a tree***Description**

Uses estimated speciation and extinction rates to sample the number of speciation events 'hidden' by subsequent extinction on each branch of a tree following Bokma (2008). For use with the `psi` and `multipsi` models.

**Usage**

```
sampleShid(phy, la = NULL, mu = NULL, useMean = FALSE)
```

**Arguments**

<code>phy</code>	An object of class <code>phylo</code> (see <b>ape</b> )
<code>la</code>	Estimate of the rate of speciation "lambda"
<code>mu</code>	Estimate of the rate of extinction "mu"
<code>useMean</code>	A logical indicating whether to output the average or expected number of hidden speciation events per branch, which may be non-integer (if TRUE), or to sample an integer number on each branch from a Poisson distribution (if FALSE, the default)

**Details**

The expected number of hidden speciation events are calculated for each branch given its start and end times, and estimates of lambda and mu which are assumed to be constant across the tree. To properly account for uncertainty in the effect of extinction on the number of nodes affecting each branch of a tree, it may be appropriate to repeat model-fitting on many realizations of Sobs on the tree of interest (similar to evaluating phylogenetic uncertainty)

**Value**

Phylogenetic tree in `phylo` format, with an added element `Sobs`, a vector of numbers of hidden speciation events per branch, in the same order as the branches in the `phylo` object

**Author(s)**

Travis Ingram

**References**

Bokma, F. 2008. Detection of "punctuated equilibrium" by Bayesian estimation of speciation and extinction rates, ancestral character states, and rates of anagenetic and cladogenetic evolution on a molecular phylogeny. *Evolution* 62: 2718-2726.

Ingram, T. 2011. Speciation along a depth gradient in a marine adaptive radiation. *Proc. R. Soc. B* 278: 613-618.



**See Also**[transformPhylo.ML](#)

---

`sortTraitData`*Sort data and remove missing entries for tree and trait data*

---

**Description**

Plots a phylogeny with lines representing the value of a continuous trait

**Usage**

```
sortTraitData(  
  phy,  
  y,  
  data.name = NULL,  
  log.trait = TRUE,  
  pass.ultrametric = FALSE  
)
```

**Arguments**

<code>phy</code>	An object of class <code>phylo</code> or <code>multiPhylo</code> (see <b>ape</b> )
<code>y</code>	A matrix of trait values with taxon names as rownames. Missing values should be NA
<code>data.name</code>	If null the first column of <code>y</code> is assumed as the trait, otherwise if <code>y</code> is a matrix with more than one column either the name of the column or the number of the column must be supplied by <code>data.name</code>
<code>log.trait</code>	Logical. If TRUE, data are log-transformed
<code>pass.ultrametric</code>	Although trees that are believed to be ultrametric to pass the function <code>is.ultrametric</code> in <b>ape</b>

**Value**

`phy` Tree with missing data pruned  
`trait` Rearranged data with missing species removed

**Author(s)**

Mark Puttick

**Examples**

```

data(anolis.tree)
data(anolis.data)
attach(anolis.data)
male.length <- matrix(Male_SVL, dimnames=list(rownames(anolis.data)))
any(is.na(male.length[,1]))
data.sorted <- sortTraitData(anolis.tree, male.length)
phy <- data.sorted[[1]]
male.length <- data.sorted[[2]]

```

---

summary.traitMedusa    *Identify shifts in the rate of trait diversification*

---

**Description**

Summarises phenotypic rate variation on phylogenies.

**Usage**

```

## S3 method for class 'traitMedusa'
summary(
  object,
  ...,
  cutoff = 4,
  AICc = TRUE,
  lowerBound = 1e-08,
  upperBound = 200,
  print.warnings = FALSE
)

```

**Arguments**

object	Output of a traitMedusa analysis in <a href="#">transformPhylo.ML</a> .
...	Additional arguments to be passed to <a href="#">summary</a> .
cutoff	Cutoff value for differences in AIC scores when comparing models. More complex models with an AIC score more than this number of units lower than simpler models are retained (as per runMedusa in the <b>geiger</b> ).
AICc	If true, AICc is used instead of AIC.
lowerBound	Minimum value for parameter estimates.
upperBound	Maximum value for parameter estimates.
print.warnings	Logical. If TRUE, warnings are issued if confidence intervals fall outside upper or lower bounds.

**Details**

This functions summarises the output of a "medusa" model in transformPhylo.ML (see below). The best overall model is chosen based on AIC (or AICc if AICc=TRUE). The cut-off point for improvement in AIC score between successively more complex models can be defined using cutoff. The default cutoff is 4 but this is somewhat arbitrary and a "good" cut-off may well vary between data sets so it may well be worth exploring different cutoffs.

**Value**

ModelFit Summary of the best optimal rate shift model.

Rates Summary of the rate parameters from the best rate shift model.

optimalTree A phylo object with branch lengths scaled relative to rate.

**Author(s)**

Gavin Thomas

**References**

Alfaro ME, Santini F, Brock CD, Alamillo H, Dornburg A, Carnevale G, Rabosky D & Harmon LJ. 2009. Nine exceptional radiations plus high turnover explain species diversity in jawed vertebrates. PNAS 106, 13410-13414.

O'Meara BC, Ane C, Sanderson MJ & Wainwright PC. 2006. Testing for different rates of continuous trait evolution using likelihood. Evolution 60, 922-933

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. Evolution 63, 2017-2030.

**See Also**

[transformPhylo.ML](#)

**Examples**

```
data(anolis.tree)
data(anolis.data)
attach(anolis.data)
male.length <- matrix(Male_SVL, dimnames=list(rownames(anolis.data)))
sortedData <- sortTraitData(anolis.tree, male.length)
phy <- sortedData$phy
male.length <- sortedData$trait
phy.clade <- extract.clade(phy, 182)
male.length.clade <- as.matrix(male.length[match(phy.clade$tip.label, rownames(male.length)),])
tm1 <- transformPhylo.ML(male.length.clade, phy=phy.clade, model="tm1", minCladeSize=10, nSplits=1)
tm1_out <- summary(tm1, cutoff=1)
```

---

traitData.plot	<i>plot a univariate continuous trait data on a phylogeny</i>
----------------	---

---

### Description

Plots a phylogeny with lines representing the value of a continuous trait

### Usage

```
traitData.plot(
  y,
  phy,
  col.label = "red",
  col.tree = "black",
  col.hist = "navy",
  cex.plot = 0.7,
  cex.tips = 0.7,
  show.tips = FALSE,
  include.hist = FALSE,
  n.split = 5,
  lwd.traits = 1,
  show.axis = TRUE,
  axis.text = NULL,
  transform.axis.label = NULL,
  at = NULL,
  labels = NULL,
  axis.text.line = 1,
  offset.bars = 1,
  ...
)
```

### Arguments

<code>y</code>	A matrix of trait values with taxon names as rownames.
<code>phy</code>	An object of class <code>phylo</code> (see <b>ape</b> ).
<code>col.label</code>	colour labels for the traits at the tips and in the histogram
<code>col.tree</code>	colour for the edge labels on the tree
<code>col.hist</code>	colour of the histogram
<code>cex.plot</code>	Numeric. The size of labels for the histogram axis labels
<code>cex.tips</code>	Numeric. The size of the phylogeny tip labels
<code>show.tips</code>	Logical. If <code>FALSE</code> (default), no tip labels are shown. If <code>TRUE</code> , tip labels are shown
<code>include.hist</code>	Logical. Include a histogram alongside the plot of the tree?
<code>n.split</code>	Numeric. The number of splits for the axis labels and shading for the trait values

lwd.traits	Line widths of traits shown on the plot
show.axis	Logical. If TRUE, shows the axis of trait values on the plot.
axis.text	text shown above the trait label axis. If NULL (default), nothing is displayed
transform.axis.label	If the data are provided as logarithms the labels for trait axis can be transformed to their original values by calculating the exponential function of the natural (transform.axis.label="exp") or base 10 logarithm. The default (NULL) leaves the labels un-transformed.
at	Axis tick point locations for if show.axis is TRUE.
labels	Axis labels locations for if show.axis is TRUE.
axis.text.line	The location of the label for the trait axis beside the plot.
offset.bars	The distance of trait plot lines from the phylogeny.
...	further arguments passed to the axis function

**Value**

A plot with the trait values shown at the tips, and a histogram of the trait values

**Author(s)**

Mark Puttick

**Examples**

```
data(anolis.tree)
data(anolis.data)
attach(anolis.data)
male.length <- matrix(Male_SVL, dimnames=list(rownames(anolis.data)))
sortedData <- sortTraitData(anolis.tree, male.length)
phy <- sortedData$phy
male.length <- sortedData$trait
traitData.plot(y=male.length, phy)
```

---

transformPhylo

*Phylogenetic tree transformations*


---

**Description**

Transforms the branch lengths of a phylo object according to a model of trait evolution (see details).

**Usage**

```
transformPhylo(
  phy,
  model = NULL,
  y = NULL,
  meserr = NULL,
  kappa = NULL,
  lambda = NULL,
  delta = NULL,
  alpha = NULL,
  psi = NULL,
  lambda.sp = NULL,
  nodeIDs = NULL,
  rateType = NULL,
  branchRates = NULL,
  cladeRates = NULL,
  splitTime = NULL,
  timeRates = NULL,
  acdcRate = NULL,
  branchLabels = NULL,
  cophenetic.dist = NULL,
  vcv.matrix = NULL,
  mode.order = NULL,
  mode.param = NULL,
  rate.var = NULL
)
```

**Arguments**

phy	An object of class phylo (see <b>ape</b> ).
model	The model of trait evolution (see details).
y	A matrix of trait values.
meserr	A vector (or matrix) of measurement error for each tip. This is only applicable to univariate analyses. Largely untested - please use cautiously
kappa	Value of kappa transform.
lambda	Value of lambda transform.
delta	Value of delta transform.
alpha	Value of alpha (OU) transform.
psi	Value of psi transform.
lambda.sp	Estimate of speciation (lambda) for the psi models
nodeIDs	Integer - ancestral nodes of clades.
rateType	If model="clade", a vector specifying if rate shift occurs in a clade ("clade") or on the single branch leading to a clade ("branch").
branchRates	Numeric vector specifying relative rates for individual branches

cladeRates	Numeric vector specifying relative rates for clades or logical to indicate scalar is included in the 'modeslice' model (the scalar is included in the mode.param argument with the 'modeslice' model).
splitTime	A split time (measured from the present, or most recent species) at which a shift in the rate occurs for the "timeSlice" model
timeRates	The rates (from ancient to recent) for the timeSlice model
acdcRate	Value of ACDC transform.
branchLabels	A vector of length equal to the number of internal branches, signifying the which "multiPsi" class it belongs to
cophenetic.dist	a cophenetic distance matrix showing the absolute distance between taxa - only applicable for OU models run on non-ultrametric trees. If null will be calculated internally, but supplying the data can speed up run time
vcv.matrix	a variance-covariance matrix - only applicable for OU models run on non-ultrametric trees. If null will be calculated internally, but supplying the data can speed up run time
mode.order	The order of modes for the 'modeslice' model. Any combination of 'BM', 'OU', 'acdc', and 'kappa'
mode.param	Parameters for the modes of evolution in the 'modeslice' model
rate.var	Allows rate variation in BM modes in the 'modeslice' model

## Details

Transforms the branch lengths of a phylo object according to one of the following models:

- model="bm" - Brownian motion (constant rates random walk)
- model="kappa" - fits Pagel's kappa by raising all branch lengths to the power kappa. As kappa approaches zero, trait change becomes focused at branching events. For complete phylogenies, if kappa approaches zero this infers speciation trait change.
- model="lambda" - fits Pagel's lambda to estimate phylogenetic signal by multiplying all internal branches of the tree by lambda, leaving tip branches as their original length (root to tip distances are unchanged);
- model="delta" - fits Pagel's delta by raising all node depths to the power delta. If delta < 1, trait evolution is concentrated early in the tree whereas if delta > 1 trait evolution is concentrated towards the tips. Values of delta above one can be difficult to fit reliably.
- model="free" - fits Mooer's et al's (1999) free model where each branch has its own rate of trait evolution. This can be a useful exploratory analysis but it is slow due to the number of parameters, particularly for large trees.
- model="clade" - fits a model where particular clades are a priori hypothesised to have different rates of trait evolution (see O'Meara et al. 2006; Thomas et al. 2006, 2009). Clades are specified using nodeIDs and are defined as the mrca node. Unique rates for each clade are specified using cladeRates. rateType specifies whether the rate shift occurs in the stem clade or on the single branch leading to the clade.
- model="OU" - fits an Ornstein-Uhlenbeck model - a random walk with a central tendency proportional to alpha. High values of alpha can be interpreted as evidence of evolutionary constraints, stabilising selection or weak phylogenetic signal.

- model="psi" - fits a model to assess to the relative contributions of speciation and gradual evolution to a trait's evolutionary rate (Ingram 2010). Note that 'original nodes' from the full phylogeny can be included as an element on the phylogeny (e.g., phy\$orig.node) as well as estimates of 'hidden' speciation (e.g., phy\$hidden.speciation) if estimates of extinction ( $\mu$ ) are  $> 0$ .
- model="multiPsi" fits a model to assess to the relative contributions of speciation and gradual evolution to a trait's evolutionary rate but allows separate values of psi fitted to separate branches (Ingram 2010; Ingram et al. 2016). Note that 'original nodes' from the full phylogeny can be included as an element on the phylogeny (e.g., phy\$orig.node) as well as estimates of 'hidden' speciation (e.g., phy\$hidden.speciation) if estimates of extinction ( $\mu$ ) are  $> 0$ .
- model="ACDC" fits a model to in which rates can exponentially increased or decrease through time (Blomberg et al. 2003). If the upper bound is  $< 0$ , the model is equivalent to the 'Early Burst' model of Harmon et al. 2010. If a nodeIDs is supplied, the model will fit a ACDC model nested within a clade, with a BM fit to the rest of the tree.
- model="timeSlice" A model in which all branch rates change at time(s) in the past.
- model="modeSlice" A model in which all branch modes change at a time or times set a priori by the user.

### Value

phy A phylo object with branch lengths scaled according to the given model and parameters

### Author(s)

Gavin Thomas, Mark Puttick

### References

- Ingram T. 2011. Speciation along a depth gradient in a marine adaptive radiation. *Proc. Roy. Soc. B.* 278, 613-618.
- Ingram T, Harrison AD, Mahler L, Castaneda MdR, Glor RE, Herrel A, Stuart YE, and Losos JB. Comparative tests of the role of dewlap size in Anolis lizard speciation. *Proc. Roy. Soc. B.* 283, 20162199.
- Mooers AO, Vamossi S, & Schluter D. 1999. Using phylogenies to test macroevolutionary models of trait evolution: sexual selection and speciation in Cranes (Gruinae). *American Naturalist* 154, 249-259.
- O'Meara BC, Ane C, Sanderson MJ & Wainwright PC. 2006. Testing for different rates of continuous trait evolution using likelihood. *Evolution* 60, 922-933
- Pagel M. 1997. Inferring evolutionary processes from phylogenies. *Zoologica Scripta* 26, 331-348.
- Pagel M. 1999 Inferring the historical patterns of biological evolution. *Nature* 401, 877-884.
- Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. *Evolution* 63, 2017-2030.

### See Also

[transformPhylo.ML](#), [transformPhylo.l1](#), [transformPhylo.MCMC](#)



**Examples**

```
data(anolis.tree)
anolis.treeDelta <- transformPhylo(phy=anolis.tree, model="delta", delta=0.5)
```

---

transformPhylo.l1      *Log-likelihood for models of trait evolution.*

---

**Description**

Fits likelihood models for various models of continuous character evolution.

**Usage**

```
transformPhylo.l1(  
  y = NULL,  
  phy,  
  model = NULL,  
  meserr = NULL,  
  kappa = NULL,  
  lambda = NULL,  
  delta = NULL,  
  alpha = NULL,  
  psi = NULL,  
  lambda.sp = NULL,  
  nodeIDs = NULL,  
  rateType = NULL,  
  branchRates = NULL,  
  cladeRates = NULL,  
  timeRates = NULL,  
  splitTime = NULL,  
  branchLabels = NULL,  
  acdcRate = NULL,  
  covPIC = TRUE,  
  cophenetic.dist = NULL,  
  vcv.matrix = NULL,  
  mode.order = NULL,  
  mode.param = NULL,  
  rate.var = NULL,  
  mu = NULL,  
  sigma.sq = NULL  
)
```

**Arguments**

y                    A matrix of trait values.  
phy                  An object of class phylo (see **ape**).

<code>model</code>	The model of trait evolution (see details).
<code>meserr</code>	A vector (or matrix) of measurement error for each tip. This is only applicable to univariate analyses.
<code>kappa</code>	Value of kappa transform.
<code>lambda</code>	Value of lambda transform.
<code>delta</code>	Value of delta transform.
<code>alpha</code>	Value of alpha (OU) transform.
<code>psi</code>	Value of psi transform.
<code>lambda.sp</code>	Speciation rate estimate for the tree.
<code>nodeIDs</code>	Integer - ancestral nodes of clades.
<code>rateType</code>	If <code>model="clade"</code> , a vector specifying if rate shift occurs in a clade ("clade") or on the single branch leading to a clade ("branch").
<code>branchRates</code>	Numeric vector specifying relative rates for individual branches
<code>cladeRates</code>	Numeric vector specifying relative rates for clades or logical to indicate scalar is included in the 'modeslice' model (the scalar value is included in the <code>mode.param</code> argument with the 'modeslice' model).
<code>timeRates</code>	The rates (from ancient to recent) for the <code>timeSlice</code> model
<code>splitTime</code>	A split time (measured from the present, or most recent species) at which a shift in the rate occurs for the "timeSlice" model
<code>branchLabels</code>	Branches on which different psi parameters are estimated in the "multipsi" model.
<code>acdcRate</code>	Value of ACDC transform
<code>covPIC</code>	Logical. For multivariate analyses, allow for co-variance between traits rates (TRUE) or no covariance in trait rates (FALSE). If FALSE, only the trait variances not co-variances are used.
<code>cophenetic.dist</code>	a cophenetic distance matrix showing the absolute distance between taxa - only applicable for OU models run on non-ultrametric trees. If null will be calculated internally, but supplying the data can speed up run time
<code>vcv.matrix</code>	a variance-covariance matrix - only applicable for OU models run on non-ultrametric trees. If null will be calculated internally, but supplying the data can speed up run time
<code>mode.order</code>	The order of modes for the 'modeslice' model. Any combination of 'BM', 'OU', 'acdc', and 'kappa'
<code>mode.param</code>	Parameters for the modes of evolution in the 'modeslice' model
<code>rate.var</code>	Allows rate variation in BM modes in the 'modeslice' model
<code>mu</code>	Phylogenetic mean estimate, Mainly for internal use when using the variance-covariance method to calculate likelihood for non-ultrametric trees with the OU model
<code>sigma.sq</code>	Brownian variance estimate, mainly for internal use when using the variance-covariance method to calculate likelihood for non-ultrametric trees with the OU model

## Details

This function fits likelihood models (see below) for continuous character evolution where the parameter values are set a priori. The function returns the log-likelihood and the Brownian variance (or variance covariance matrix).

- model="bm" Brownian motion (constant rates random walk).
- model="kappa" fits Pagel's kappa by raising all branch lengths to the power kappa. As kappa approaches zero, trait change becomes focused at branching events. For complete phylogenies, if kappa approaches zero this infers speciation trait change. Default bounds from ~0 - 1.
- model="lambda" fits Pagel's lambda to estimate phylogenetic signal by multiplying all internal branches of the tree by lambda, leaving tip branches as their original length (root to tip distances are unchanged). Default bounds from ~0 - 1.
- model="delta" fits Pagel's delta by raising all node depths to the power delta. If delta < 1, trait evolution is concentrated early in the tree whereas if delta > 1 trait evolution is concentrated towards the tips. Values of delta above one can be difficult to fit reliably. If a nodeIDs is supplied, the model will fit a delta model nested within a clade, with a BM fit to the rest of the tree. Default bounds from ~0 - 5.
- model="OU" fits an Ornstein-Uhlenbeck model - a random walk with a central tendency proportional to alpha. High values of alpha can be interpreted as evidence of evolutionary constraints, stabilising selection or weak phylogenetic signal. It is often difficult to distinguish among these possibilities. If a nodeIDs is supplied, the model will fit a OU model nested within a clade, with a BM fit to the rest of the tree. For OU models, alternative optimisation are performed with different starting values (1e-8, 0.01, 0.1, 1, 5). Default bounds from ~0 - 10.
- model="ACDC" fits a model to in which rates can exponentially increased or decrease through time (Blomberg et al. 2003). If the upper bound is < 0, the model is equivalent to the 'Early Burst' model of Harmon et al. 2010. If a nodeIDs is supplied, the model will fit a ACDC model nested within a clade, with a BM fit to the rest of the tree. Default rate parameter bounds from  $\ln(1e-10) \sim \ln(20)$  divided by the root age. Note this process starts on the stem branch leading to the MRCA of the common node, unlike the other methods that start at the common node.
- model="trend" fits a model in which the expected mean change through time is non-zero, signifying a directional evolution to a larger or smaller trait value. This model is only applicable to non-ultrametric trees.
- model="psi" fits a model to assess to the relative contributions of speciation and gradual evolution to a trait's evolutionary rate (Ingram 2010). Note that the algorithm will automatically estimate speciation and extinction estimates, and will incorporate estimates of 'hidden' speciation if death estimates are greater than 0.
- model="multipsi" fits a model to assess to the relative contributions of speciation and gradual evolution to a trait's evolutionary rate but allows separate values of psi fitted to separate branches (Ingram 2010; Ingram et al. 2016). Note that the algorithm will automatically estimate speciation and extinction estimates, and will incorporate estimates of 'hidden' speciation if death estimates are greater than 0.
- model="free" fits Mooers et al's free model where each branch has its own rate of trait evolution. This can be a useful exploratory analysis but it is slow due to the number of parameters, particularly for large trees. Default rate parameter bounds from ~0 - 200.

- `model="clade"` fits a model where particular clades are a priori hypothesised to have different rates of trait evolution (see O'Meara et al. 2006; Thomas et al. 2006, 2009). Clades are specified using nodeIDs and are defined as the mrca node. Default rate parameter bounds from  $\sim 0$  - 200.
- `model="tm1"` fits "clade" models without any a priori assertion of the location of phenotypic diversification rate shifts. It uses the same AIC approach as the `runMedusa` function in the `geiger` package (`runMedusa` tests for shifts in the rate of lineage diversification). The algorithm first fits a constant-rate Brownian model to the data, it then works iteratively through the phylogeny fitting a two-rate model at each node in turn. Each two-rate model is compared to the constant rate model and the best two-rate model is retained. Keeping the location of this rate shift intact, it then repeats the procedure for a three-rate model and so on. The maximum number of rate shifts can be specified a priori using `nSplits`. Limits can be applied to the size (species richness) of clades on which to infer new rate shifts using `minCladeSize`. This can be useful to enable large trees to be handled but should be used cautiously since specifying a large minimum clade size may result in biologically interesting nested rate shifts being missed. Equally, very small clade sizes may provide poor estimates of rate that may not be informative. Limits on the search can also be placed using `restrictNode`. This requires a list where each element of the list is a vector of tip names that define monophyletic groups. Rate shifts will not be searched for within any of the defined groups. Default rate parameter bounds from  $\sim 0$  - 1000.
- `model="tm2"` this model is similar to "tm1", however, at each node it assesses the fit of two models. The first model is exactly as per "tm1". The second model infers a rate shift on the single branch descending directly from a node but not on any of the descending branches thereafter. Only the best fitting single-branch or whole clade model is retained for the next iteration. If a single-branch shift is favoured, this infers either that there was a rapid shift in trait value along the stem leading to the crown group, or that the members of the clade have undergone parallel shifts. In either case, this can be considered as a change in mean, though separating a single early shift from a clade-parallel shift is not possible with this method.
- `model="timeSlice"` A model in which all branch rates change at a time or times set a priori by the user. If Default rate parameter bounds from  $\sim 0$  - 1000. If `splitTime=NULL`, all 1 Ma (as defined by `test Age`) intervals from the root of the tree - 10 and the youngest tip + 10 will be included in the search. The  $\pm 10$  Ma age can be modified using the argument `boundaryAge`. At each stage the best fitting model will be stored, and the search will continue until `n` shifts, with `n` shifts defined by `nSplits`. If a single value or vector is used for `splitTime`, only these ages are included in the search.
- `model="modeslice"` A model in which all branch modes change at a time or times set a priori by the user.

### Value

`brownianVariance` Brownian variance (or covariance for multiple traits) given the data and phylogeny

`logLikelihood` The log-likelihood of the model and data

### Author(s)

Gavin Thomas, Mark Puttick

## References

- Felsenstein J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous characters. *Am. J. Hum. Genet.* 25, 471-492. Felsenstein J. 1985. Phylogenies and the comparative method. *American Naturalist* 125, 1-15. Freckleton RP & Jetz W. 2009. Space versus phylogeny: disentangling phylogenetic and spatial signals in comparative data. *Proc. Roy. Soc. B* 276, 21-30.
- Ingram T. 2011. Speciation along a depth gradient in a marine adaptive radiation. *Proc. Roy. Soc. B* 278, 613-618.
- Ingram T, Harrison AD, Mahler L, Castaneda MdR, Glor RE, Herrel A, Stuart YE, and Losos JB. 2016. Comparative tests of the role of dewlap size in *Anolis* lizard speciation. *Proc. Roy. Soc. B* 283, 20162199. # Mooers AO, Vamosi S, & Schluter D. 1999. Using phylogenies to test macroevolutionary models of trait evolution: sexual selection and speciation in Cranes (Gruinae). *American Naturalist* 154, 249-259. O'Meara BC, Ane C, Sanderson MJ & Wainwright PC. 2006. Testing for different rates of continuous trait evolution using likelihood. *Evolution* 60, 922-933. Pagel M. 1997. Inferring evolutionary processes from phylogenies. *Zoologica Scripta* 26, 331-348. Pagel M. 1999. Inferring the historical patterns of biological evolution. *Nature* 401, 877-884. Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in *Anolis*: novel environments and island effects. *Evolution* 63, 2017-2030.

## See Also

[transformPhylo.ML](#), [transformPhylo.MCMC](#), [transformPhylo.sim](#)

## Examples

```
# Data and phylogeny
data(anolis.tree)
data(anolis.data)

# anolis.data is not matrix and contains missing data so put together matrix of
# relevant traits (here female and male snout vent lengths) and remove species
# with missing data from the matrix and phylogeny
sorted.traits <- sortTraitData(anolis.tree, anolis.data,
c("Female_SVL", "Male_SVL"), log.trait=TRUE, pass.ultrametric=TRUE)

tree <- sorted.traits$phy
traits <- sorted.traits$trait

# log likelihood of kappa = 0.1 or 1
transformPhylo.ll(traits, phy=tree, model="kappa", kappa=0.1)
transformPhylo.ll(traits, phy=tree, model="kappa", kappa=1)

# log likelihood of lambda = 0.01 or 1
transformPhylo.ll(traits, phy=tree, model="lambda", lambda=0.01)
transformPhylo.ll(traits, phy=tree, model="lambda", lambda=1)

# log likelihood of delta = 1.5 or 1
transformPhylo.ll(traits, phy=tree, model="delta", delta=1.5)
transformPhylo.ll(traits, phy=tree, model="delta", delta=1)

# log likelihood of alpha = 0.001 or 2
```

```
transformPhylo.ll(traits, phy=tree, model="OU", alpha=0.001)
transformPhylo.ll(traits, phy=tree, model="OU", alpha=2)
```

---

transformPhylo.MCMC    *Bayesian MCMC for models of trait evolution*

---

## Description

Fits Bayesian models for various models of continuous character evolution using a Metropolis-Hastings Markov Chain Monte Carlo (MCMC) approach

## Usage

```
transformPhylo.MCMC(
  y,
  phy,
  model,
  mcmc.iteration = 1000,
  burn.in = 0.1,
  hiddenSpeciation = FALSE,
  full.phy = NULL,
  lowerBound = NULL,
  upperBound = NULL,
  useMean = FALSE,
  random.start = TRUE,
  meserr = NULL,
  covPIC = TRUE,
  lambdaEst = FALSE,
  nodeIDs = NULL,
  branchLabels = NULL,
  acdcScalar = FALSE,
  sample.every = 10
)
```

## Arguments

y	A matrix of trait values.
phy	An object of class phylo (see <b>ape</b> ).
model	The model of trait evolution (see details).
mcmc.iteration	Integer - the number of generations for which to run the MCMC chain
burn.in	The proportion of the chain (as given by mcmc.iteration) which to discard as 'burn-in'
hiddenSpeciation	Logical. If TRUE the psi model will include nodes that are on the 'full.phy' but not the tree pruned of trait data

full.phy	The full phylogeny containing the species that do not contain trait data so are not included in 'phy'
lowerBound	Minimum value for parameter estimates
upperBound	Maximum value for parameter estimates
useMean	Logical. Use the branch-based estimates of extinction of mean (TRUE, default) for the "psi" and "multispi" models only applicable if "hiddenSpeciation" = TRUE
random.start	Use a random starting value for the MCMC run (TRUE), or use the environment set.seed() value
meserr	A vector (or matrix) of measurement error for each tip. This is only applicable to univariate analyses
covPIC	Logical. For multivariate analyses, allow for co-variance between traits rates (TRUE) or no covariance in trait rates (FALSE). If FALSE, only the trait variances not co-variances are used.
lambdaEst	Logical. Estimate lambda alongside parameter estimates to reduce data noise. Only applicable for models "kappa", "delta", "OU", "psi", "multispi", and "ACDC". Default=FALSE.
nodeIDs	Integer - ancestral nodes of clades applicable to rate heterogenous and nested models of evolution (see details)
branchLabels	Branches on which different psi parameters are estimated in the "multispi" model
acdcScalar	Logical. For nested EB rate model, simultaneously estimated a rate scalar alongside EB model. Default=FALSE.
sample.every	Number specifying the every nth that is sampled in the MCMC chain (default = 1).

## Details

The method estimates posterior probabilities using a Metropolis-Hastings MCMC approach that places a prior bounded uniform distribution on all parameters with an independence sampler. These prior distributions can be altered by changing the upperBound and lowerBound arguments. The MCMC model will estimate the posterior probability for the following models:

- model="kappa" fits Pagel's kappa by raising all branch lengths to the power kappa. As kappa approaches zero, trait change becomes focused at branching events. For complete phylogenies, if kappa approaches zero this infers speciation trait change. Default bounds from ~0 - 1.
- model="lambda" fits Pagel's lambda to estimate phylogenetic signal by multiplying all internal branches of the tree by lambda, leaving tip branches as their original length (root to tip distances are unchanged). Default bounds from ~0 - 1.
- model="delta" fits Pagel's delta by raising all node depths to the power delta. If delta < 1, trait evolution is concentrated early in the tree whereas if delta > 1 trait evolution is concentrated towards the tips. Values of delta above one can be difficult to fit reliably. Default bounds from ~0 - 5.

- `model="OU"` fits an Ornstein-Uhlenbeck model - a random walk with a central tendency proportional to alpha. High values of alpha can be interpreted as evidence of evolutionary constraints, stabilising selection or weak phylogenetic signal. It is often difficult to distinguish among these possibilities. Default bounds from  $\sim 0$  - 10.
- `model="psi"` fits a acceleration-deacceleration model to assess to the relative contributions of speciation and gradual evolution to a trait's evolutionary rate (Ingram 2010).
- `model="ACDC"` fits a model to in which rates can exponentially increased or decrease through time (Blomberg et al. 2003). If the upper bound is  $< 0$ , the model is equivalent to the 'Early Burst' model of Harmon et al. 2010. Default rate parameter bounds from  $\ln(1e-10) \sim \ln(20)$  divided by the root age.

### Value

median The median estimate of the posterior for the parameter

95.HPD The 95 percent Highest Posterior Density for the parameter

ESS Effective Sample Size for the posterior

acceptance.rate The ratio for which new proposals were accepted during the MCMC chain

mcmc.chain Full MCMC chain containing all iterations (including burn-in)

### Author(s)

Mark Puttick, Gavin Thomas

### See Also

[transformPhylo.ML](#), [transformPhylo.ll](#), [transformPhylo](#)

### Examples

```
data(anolis.tree)
data(anolis.data)
attach(anolis.data)
male.length <- matrix(Male_SVL, dimnames=list(rownames(anolis.data)))
sortedData <- sortTraitData(anolis.tree, male.length)
phy <- sortedData$phy
male.length <- sortedData$trait
phy.clade <- extract.clade(phy, 182)
male.length.clade <- as.matrix(male.length[match(phy.clade$tip.label, rownames(male.length)),])
## please note, this model will be need to run for longer to achieve convergence
lambda.mcmc <- transformPhylo.MCMC(y=male.length.clade, phy=phy.clade,
model="lambda", mcmc.iteration=100, burn.in=0.1)
```



---

transformPhylo.ML	<i>Maximum likelihood for models of trait evolution</i>
-------------------	---

---

### Description

Fits likelihood models for various models of continuous character evolution. Model fitting is based on maximum-likelihood evaluation using phylogenetically independent contrasts. This is exactly equivalent to, but substantially faster than, GLS approaches.

### Usage

```
transformPhylo.ML(  
  y,  
  phy,  
  model = NULL,  
  modelCIs = TRUE,  
  nodeIDs = NULL,  
  rateType = NULL,  
  minCladeSize = 1,  
  nSplits = 2,  
  splitTime = NULL,  
  boundaryAge = 10,  
  testAge = 1,  
  restrictNode = NULL,  
  lambdaEst = FALSE,  
  acdcScalar = FALSE,  
  branchLabels = NULL,  
  hiddenSpeciation = FALSE,  
  full.phy = NULL,  
  useMean = FALSE,  
  profilePlot = FALSE,  
  lowerBound = NULL,  
  upperBound = NULL,  
  covPIC = TRUE,  
  n.cores = 1,  
  tol = NULL,  
  meserr = NULL,  
  controllist = c(fnscale = -1, maxit = 100, factr = 1e-07, pgtol = 0, type = 2, lmm =  
    5),  
  returnPhy = FALSE,  
  print.warnings = FALSE,  
  mode.order = NULL,  
  rate.var = FALSE,  
  testShiftTimes = NULL,  
  saveAll = TRUE  
)
```

**Arguments**

y	A matrix of trait values.
phy	An object of class phylo (see <b>ape</b> ).
model	The model of trait evolution (see details).
modelCIs	Logical - estimate confidence intervals for parameter estimates.
nodeIDs	Integer - ancestral nodes of clades applicable to rate heterogenous and nested models of evolution (see details)
rateType	If model="clade", a vector specifying if rate shift occurs in a clade ("clade") or on the single branch leading to a clade ("branch").
minCladeSize	Integer - minimum clade size for inferred rate shift (where model="medusa").
nSplits	Integer - number of rate shifts to apply for model="medusa" and "timeSlice".
splitTime	A split time (measured from the present, or most recent species) at which a shift in the rate occurs for the "timeSlice" model. If splitTime=NULL, then all ages between 1 million year intervals from the root age - 10 Ma to the present + 10 Ma will be included in the search. The best model will be retained in each search, and will be used as a fixed age in the next search. The model will calculate the likelihood for the number of shifts defined by 'nSplits'.
boundaryAge	Only applicable if splitTime=NULL, the age distance from the tips and and youngest tip for which to search for rate shifts. For example, if boundaryAge=10, only ages between the root age - 10 and the latest tip + 10 will be included in the search. If one value is given this will be used for upper and lower ages, but if a vector with two ages is provided the first is used for the upper age boundary and the second for the lower age boundary. Set to zero to allow testing of all ages.
testAge	If splitTime=NULL, the interval between ages to be tested. For example, if testAge=1, all 1 Ma ages between the ages defined by 'boundaryAge' will be tested. If you would like to sequentially test specific shift times only, please use the argument "specificShiftTimes".
restrictNode	List defining monophyletic groups within which no further rate shifts are searched.
lambdaEst	Logical.Estimate lambda alongside parameter estimates to reduce data noise. Only applicable for models "kappa", "delta", "OU", "psi", "multispi", and "ACDC". Default=FALSE.
acdcScalar	Logical.For nested EB rate model, simultaneously estimated a rate scalar alongside EB model. Default=FALSE. Only applicable to 'nested mode' and 'modeSlice' models.
branchLabels	Branches on which different psi parameters are estimated in the "multispi" model
hiddenSpeciation	Logical. If TRUE the psi model will include nodes that are on the 'full.phy' but not the tree pruned of trait data
full.phy	The full phylogeny containing the species that do not contain trait data so are not included in 'phy'
useMean	Logical. Use the branch-based estimates of extinction of mean (TRUE, default) for the "psi" and "multispi" models. Only applicable if "hiddenSpeciation"=TRUE. If FALSE, this will generate a single realisation of the numbers of hidden speciation events on each branch

profilePlot	Logical. For the single parameter models "kappa", "lambda", "delta", "OU", "psi", "multipsi", and "ACDC", plot the profile of the likelihood.
lowerBound	Minimum value for parameter estimates
upperBound	Maximum value for parameter estimates
covPIC	Logical. For multivariate analyses, allow for co-variance between traits rates (TRUE) or no covariance in trait rates (FALSE). If FALSE, only the trait variances not co-variances are used.
n.cores	Integer. Set number of computing cores when running model="traitMedusa" (tm1 and tm2 models)
tol	Tolerance (minimum branch length) to exclude branches from trait MEDUSA search. Primarily intended to prevent inference of rate shifts at randomly resolved polytomies.
meserr	A vector (or matrix) of measurement error for each tip. This is only applicable to univariate analyses.
controlList	List. Specify fine-tune parameters for the optim likelihood search
returnPhy	Logical. In TRUE the phylogeny with branch lengths transformed by the ML model parameters is returned
print.warnings	Logical. If TRUE, warnings are issued if confidence intervals fall outside upper or lower bounds
mode.order	The order of modes for the 'modeslice' model. Any combination of 'BM', 'OU', 'acdc', and 'kappa'
rate.var	Allows rate variation in BM modes in the 'modeslice model'
testShiftTimes	A vector of times to be used in the search for split times. For use in the timeSlice model when splitTime=NULL. This allows users to specify ages that are test sequentially, rather than all shifts optimised simultaneously as is done when ages are provided in the argument 'splitTime'.
saveAll	Logical. If TRUE, saves all the outputs from traitMedusa search in timeSlice (i.e, the log-likelihood and rate estimates for all considered shifts, not just the best fitting shift model). This can be used for model averaging with the function <a href="#">plot.timeSlice.ML</a>

## Details

This function finds the maximum likelihood parameter values for continuous character evolution. For "kappa", "delta", "OU", "multipsi", and "ACDC" it is possible to fit a 'nested' model of evolution in which the ancestral rate of BM switches to a different node, as specified by nodeIDs or branchLabels for multipsi. The function returns the maximum-likelihood parameter estimates for the following models.

- model="bm" Brownian motion (constant rates random walk).
- model="kappa" fits Pagel's kappa by raising all branch lengths to the power kappa. As kappa approaches zero, trait change becomes focused at branching events. For complete phylogenies, if kappa approaches zero this infers speciation trait change. Default bounds from -0 - 1.

- model="lambda" fits Pagel's lambda to estimate phylogenetic signal by multiplying all internal branches of the tree by lambda, leaving tip branches as their original length (root to tip distances are unchanged). Default bounds from ~0 - 1.
- model="delta" fits Pagel's delta by raising all node depths to the power delta. If delta <1, trait evolution is concentrated early in the tree whereas if delta >1 trait evolution is concentrated towards the tips. Values of delta above one can be difficult to fit reliably. If a nodeIDs is supplied, the model will fit a delta model nested within a clade, with a BM fit to the rest of the tree. Default bounds from ~0 - 5.
- model="OU" fits an Ornstein-Uhlenbeck model - a random walk with a central tendency proportional to alpha. High values of alpha can be interpreted as evidence of evolutionary constraints, stabilising selection or weak phylogenetic signal. It is often difficult to distinguish among these possibilities. If a nodeIDs is supplied, the model will fit a OU model nested within a clade, with a BM fit to the rest of the tree. For OU models, alternative optimisation are performed with different starting values (1e-8, 0.01, 0.1, 1, 5). Default bounds from ~0 - 10.
- model="ACDC" fits a model to in which rates can exponentially increased or decrease through time (Blomberg et al. 2003). If the upper bound is < 0, the model is equivalent to the 'Early Burst' model of Harmon et al. 2010. If a nodeIDs is supplied, the model will fit a ACDC model nested within a clade, with a BM fit to the rest of the tree. Default rate parameter bounds from  $\ln(1e-10) \sim \ln(20)$  divided by the root age. Note this process starts on the stem branch leading to the MRCA of the common node, unlike the other methods that start at the common node.
- model="trend" fits a model in which the expected mean change through time is non-zero, signifying a directional evolution to a larger or smaller trait value. This model is only applicable to non-ultrametric trees.
- model="psi" fits a model to assess to the relative contributions of speciation and gradual evolution to a trait's evolutionary rate (Ingram 2010). Note that the algorithm will automatically estimate speciation and extinction estimates, and will incorporate estimates of 'hidden' speciation if death estimates are greater than 0.
- model="multipsi" fits a model to assess to the relative contributions of speciation and gradual evolution to a trait's evolutionary rate but allows separate values of psi fitted to separate branches (Ingram 2010; Ingram et al. 2016). Note that the algorithm will automatically estimate speciation and extinction estimates, and will incorporate estimates of 'hidden' speciation if death estimates are greater than 0.
- model="free" fits Mooers et al's free model where each branch has its own rate of trait evolution. This can be a useful exploratory analysis but it is slow due to the number of parameters, particularly for large trees. Default rate parameter bounds from ~0 - 200.
- model="clade" fits a model where particular clades are a priori hypothesised to have different rates of trait evolution (see O'Meara et al. 2006; Thomas et al. 2006, 2009). Clades are specified using nodeIDs and are defined as the mrca node. Default rate parameter bounds from ~0 - 200.
- model="tm1" fits "clade" models without any a priori assertion of the location of phenotypic diversification rate shifts. It uses the same AIC approach as the runMedusa function in the geiger package (runMedusa tests for shifts in the rate of lineage diversification). The algorithm first fits a constant-rate Brownian model to the data, it then works iteratively through the phylogeny fitting a two-rate model at each node in turn. Each two-rate model is compared to

the constant rate model and the best two-rate model is retained. Keeping the location of this rate shift intact, it then repeats the procedure for a three-rate model and so on. The maximum number of rate shifts can be specified a priori using `nSplits`. Limits can be applied to the size (species richness) of clades on which to infer new rate shifts using `minCladeSize`. This can be useful to enable large trees to be handled but should be used cautiously since specifying a large minimum clade size may result in biologically interesting nested rate shifts being missed. Equally, very small clade sizes may provide poor estimates of rate that may not be informative. Limits on the search can also be placed using `restrictNode`. This requires a list where each element of the list is a vector of tip names that define monophyletic groups. Rate shifts will not be searched for within any of the defined groups. Default rate parameter bounds from  $\sim 0$  - 1000.

- `model="tm2"` this model is similar to "tm1", however, at each node it assesses the fit of two models. The first model is exactly as per "tm1". The second model infers a rate shift on the single branch descending directly from a node but not on any of the descending branches thereafter. Only the best fitting single-branch or whole clade model is retained for the next iteration. If a single-branch shift is favoured, this infers either that there was a rapid shift in trait value along the stem leading to the crown group, or that the members of the clade have undergone parallel shifts. In either case, this can be considered as a change in mean, though separating a single early shift from a clade-parallel shift is not possible with this method.
- `model="timeSlice"` A model in which all branch rates change at a time or times set a priori by the user. If `Default` rate parameter bounds from  $\sim 0$  - 1000. If `splitTime=NULL`, all 1 Ma (as defined by `test Age`) intervals from the root of the tree - 10 and the youngest tip + 10 will be included in the search. The +/- 10 Ma age can be modified using the argument `boundaryAge`. At each stage the best fitting model will be stored, and the search will continue until `n` shifts, with `n` shifts defined by `nSplits`. If a single value or vector is used for `splitTime`, only these ages are included in the search.
- `model="modeslice"` A model in which all branch modes change at a time or times set a priori by the user.

## Value

Returns the maximum log-likelihood and parameter estimates (with 95 percent confidence intervals if specified). If `model="bm"` instead returns the Brownian (co)variance and log-likelihood. Also returned are the root estimate, the AIC, and AICc.

`traitMedusaObject` A list in which the first element contains a matrix summarising the parameter estimates and node ids, log-likelihoods, number of parameters (`k`), AIC and AICc for the best one-rate model, two-rate model, three rate model and so on. The second element is a sub-list where the first element contains all two-rate models, the second element contains all three-rate models and so on. This can be summarised using `traitMedusaSummary`. The third element is the input trait data. The fourth element is the input phylogeny.

## Note

Confidence intervals are based on the assumption of an asymptotic Chi-square distribution. For multi-parameter models (e.g. rate shift models with more than two rates) the confidence intervals are approximate and are calculated for each parameter in turn while holding all other parameters at their maximum likelihood value.

**Author(s)**

Gavin Thomas, Mark Puttick

**References**

- Alfaro ME, Santini F, Brock CD, Alamillo H, Dornburg A, Carnevale G, Rabosky D & Harmon LJ. 2009. Nine exceptional radiations plus high turnover explain species diversity in jawed vertebrates. *PNAS* 106, 13410-13414.
- Blomberg SP, Garland T & Ives AR 2003. Testing for phylogenetic signal in comparative data: behavioral traits are more labile. *Evolution* 57, 717-745.
- Felsenstein J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous characters. *Am. J. Hum. Genet.* 25, 471-492.
- Felsenstein J. 1985. Phylogenies and the comparative method. *American Naturalist* 125, 1-15.
- Freckleton RP & Jetz W. 2009. Space versus phylogeny: disentangling phylogenetic and spatial signals in comparative data. *Proc. Roy. Soc. B* 276, 21-30.
- Harmon LJ et al. 2010. Early bursts of body size and shape evolution are rare in comparative data. *Evolution* 57, 717-745.
- Ingram T. 2011. Speciation along a depth gradient in a marine adaptive radiation. *Proc. Roy. Soc. B.* 278, 613-618.
- Ingram T, Harrison AD, Mahler L, Castaneda MdR, Glor RE, Herrel A, Stuart YE, and Losos JB. 2016. Comparative tests of the role of dewlap size in *Anolis* lizard speciation. *Proc. Roy. Soc. B.* 283, 20162199.
- Mooers AO, Vamossi S, & Schluter D. 1999. Using phylogenies to test macroevolutionary models of trait evolution: sexual selection and speciation in Cranes (Gruinae). *American Naturalist* 154, 249-259.
- O’Meara BC, Ane C, Sanderson MJ & Wainwright PC. 2006. Testing for different rates of continuous trait evolution using likelihood. *Evolution* 60, 922-933
- Pagel M. 1997. Inferring evolutionary processes from phylogenies. *Zoologica Scripta* 26, 331-348.
- Pagel M. 1999 Inferring the historical patterns of biological evolution. *Nature* 401, 877-884.
- Pagel M. 1999 Inferring the historical patterns of biological evolution. *Nature* 401, 877-884.
- Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in *Anolis*: novel environments and island effects. *Evolution* 63, 2017-2030.

**See Also**

[transformPhylo.MCMC](#), [transformPhylo](#), [transformPhylo.ll](#), [blomberg.k](#)

**Examples**

```
# Data and phylogeny
data(anolis.tree)
data(anolis.data)
sortedData <- sortTraitData(anolis.tree, anolis.data,
data.name="Male_SVL", log.trait=TRUE, pass.ultrametric=TRUE)
phy <- sortedData$phy
```

```

male.length <- sortedData$trait
phy.clade <- extract.clade(phy, 182)
male.length.clade <- as.matrix(male.length[match(phy.clade$tip.label, rownames(male.length)),])

# Brownian motion model
transformPhylo.ML(male.length.clade , phy=phy.clade, model="bm")

# Delta
transformPhylo.ML(male.length.clade , phy=phy.clade, model="delta", upperBound=2)

# The upper confidence interval for kappa is outside the bounds so try increasing
# the upper bound

transformPhylo.ML(male.length.clade , phy=phy.clade, model="delta", upperBound=5)

# Test for different rates in different clades - here with 2 hypothesised
# unusual rates compared to the background

# This fits the non-censored model of O'Meara et al. (2006)
phy.clade$node.label[which(phy.clade$node.label == "3")] <- 2
transformPhylo.ML(male.length.clade, phy=phy.clade, model="clade", nodeIDs=c(49, 54))

# Identify rate shifts and print and plot results with upto three rate shifts
# and minimum clade size of 20.
anolisSVL_MEDUSA <- transformPhylo.ML(male.length.clade, phy=phy.clade, model="tm1",
minCladeSize=10, nSplits=2)

```

---

transformPhylo.sim      *Phylogenetic tree transformations*

---

## Description

Simulates trait data on a tree using a specified model of evolution (see details).

## Usage

```

transformPhylo.sim(
  phy,
  n = 1,
  x = NULL,
  model = NULL,
  returnNodes = FALSE,
  kappa = NULL,
  lambda = NULL,
  delta = NULL,
  alpha = NULL,
  psi = NULL,
  acdcRate = NULL,
  lambda.sp = NULL,

```

```

trend = NULL,
trend.anc.state = 0,
nodeIDs = NULL,
rateType = NULL,
cladeRates = NULL,
branchRates = NULL,
rate = NULL,
group.means = NULL,
splitTime = NULL,
timeRates = NULL,
branchLabels = NULL,
rate.var = NULL,
mode.order = NULL
)

```

### Arguments

phy	An object of class phylo (see <b>ape</b> ).
n	Number of simulations
x	Vector, matrix or data.frame (with taxon names as names or rownames) of categories for each species. Only applicable if model="mixedRate"
model	The model of trait evolution (see details).
returnNodes	Logical. If TRUE, alongside the tip values all node values are returned corresponding to APE's edge.matrix for the tree.
kappa	Value of kappa transform.
lambda	Value of lambda transform.
delta	Value of delta transform.
alpha	Value of alpha (OU) transform.
psi	Value of psi transform. Note that 'original nodes' from the full phylogeny can be included as an element on the phylogeny (e.g., phy\$orig.node) as well as estimates of 'hidden' speciation (e.g., phy\$hidden.speciation) if estimates of extinction ( $\mu$ ) are $> 0$ .
acdcRate	Value of ACDC transform.
lambda.sp	Estimate of speciation (lambda) for the psi models
trend	value of the expectation mean change through time
trend.anc.state	the expected ancestral state for the trend model (default is 0)
nodeIDs	Integer - ancestral nodes of clades.
rateType	If model="clade", a vector specifying if rate shift occurs in a clade ("clade") or on the single branch leading to a clade ("branch").
cladeRates	Numeric vector specifying relative rates for clades or logical to indicate scalar is included in the 'modeslice' model (the scalar is included in the mode.param argument with the 'modeslice' model).
branchRates	Numeric vector specifying relative rates for individual branches



rate	a vector of relative rate parameters. The length of the vector is equal to the number of rates being estimated.
group.means	a vector of the relative difference in means between rate categories, expressed as a scalar applied to the expected standard deviation (see Ricklefs 2006)
splitTime	A split time (measured from the present, or most recent species) at which a shift in the rate occurs for the "timeSlice" model
timeRates	The rates (from ancient to recent) for the timeSlice model
branchLabels	Branches on which different psi parameters are estimated in the "multipsi" model.
rate.var	Allows rate variation in BM modes in the 'modeslice' model
mode.order	The order of modes for the 'modeslice' model. Any combination of 'BM', 'OU', 'acdc', and 'kappa'

**Value**

Returns a matrix of simulated dated with taxon names as rownames (number of columns=n).

**Author(s)**

Gavin Thomas, Mark Puttick

**References**

- Ricklefs RE. 2006. Time, species, and the generation of trait variation in clades. *Systematic Biology* 55, 151-159.
- Ricklefs RE. 2006. Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. *Evolution* 63, 2017-2030

**See Also**

[transformPhylo.ML](#), [transformPhylo.ll](#), [transformPhylo](#), [transformPhylo.MCMC](#)

**Examples**

```
data(anolis.tree)
data(anolis.data)

# Simulate 10 sets of data with kappa=0.1 using the anolis tree
sim.dat1 <- transformPhylo.sim(phy=anolis.tree, n=10, model="kappa", kappa=0.1)

# Simulate 10 sets of data where rates and means differ between to the categories defined by "x"
x <- anolis.data$geo_ecomorph
names(x) <- rownames(anolis.data)
sim.dat2 <- transformPhylo.sim(phy=anolis.tree, n=10, x=x, model="mixedRate", rate=c(1,1,2,4),
group.means=c(0,5,0,0))
```

---

transformRateMatrix    *Conversion among data and phylogeny objects*

---

### Description

Transforms the expected variance and covariances among species according to hypotheses of rate variation between lineages.

### Usage

```
transformRateMatrix(rateData, rate = NULL)
```

### Arguments

rateData	an object of class rateData
rate	a vector of relative rate parameters. The length of the vector is equal to the number of rates being estimated. If rate=NULL then rates are equal.

### Value

retMat Rate-transformed variance covariance matrix

### Author(s)

Gavin Thomas

### References

Thomas GH, Freckleton RP, & Szekely T. 2006. Comparative analyses of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society B* 273, 1619-1624.

Thomas GH, Meiri S, & Phillimore AB. 2009. Body size diversification in Anolis: novel environments and island effects. *Evolution* 63, 2017-2030.

### Examples

```
data(anolis.tree)
data(anolis.data)
## Convert data to class rateData with a rateMatrix object as input
anolis.rateMatrix <- as.rateMatrix(phy=anolis.tree, x="geo_ecomorph", data=anolis.data)
anolis.rateData <- as.rateData(y="Female_SVL", x="geo_ecomorph",
rateMatrix = anolis.rateMatrix, phy=NULL, data=anolis.data, log.y=TRUE)

# Tranform the expected variance covariance matrix so that the rates in the first and last
# categories are equal (both 1) whereas the rate in the second category is twice as fast (2) and
# the rate in the third category is ten times slower.

trans.anolis.rateData <- transformRateMatrix(rateData=anolis.rateData, rate = c(1,2,0.1,1))
```

# Index

## \* datasets

- allopatric.data, 4
- anolis.data, 5
- anolis.tree, 6
- finch.data, 18
- finch.tree, 19
- mammals, 22

## \* package

- motmot-package, 3

ace, 37

addFossilToPhy, 3

allopatric.data, 4

anolis.data, 5

anolis.tree, 6

as.rateData, 6

as.rateMatrix, 8

blomberg.k, 9, 62

calcCutOff, 10

chr.disp.lrt, 11, 13, 14

chr.disp.param, 11, 12, 14

chr.disp.sim, 11, 13, 13

contemporaryPhy, 15

dropTipPartial, 16

fairProportions, 17

finch.data, 18

finch.tree, 19

likRatePhylo, 19

likTraitPhylo, 21

mammals, 22

mcmc.plot, 23

ML.RatePhylo, 24

motmot (motmot-package), 3

motmot-package, 3

node.descendants, 26

nodeTimes, 27

optim.likRatePhylo, 28

pgls, 34

phyloCovar, 29

phyloMean, 30

phyloVar, 32

pic.pgls, 33

plot.phylo, 35, 37

plot.timeSlice.ML, 35, 59

plot.traitMedusa.model, 37

RatePhylo.allCI, 38

sampleShid, 40

sortTraitData, 41

summary, 42

summary.traitMedusa, 37, 42

traitData.plot, 44

transformPhylo, 10, 45, 56, 62, 65

transformPhylo.ll, 10, 48, 49, 56, 62, 65

transformPhylo.MCMC, 10, 23, 48, 53, 54, 62, 65

transformPhylo.ML, 9, 10, 17, 35–37, 41–43, 48, 53, 56, 57, 65

transformPhylo.sim, 53, 63

transformRateMatrix, 66