Package 'phylolm'

July 23, 2025

Version 2.6.5

Date 2024-09-24

Title Phylogenetic Linear Regression

Depends R (>= 3.0), ape

Imports future.apply

Description Provides functions for fitting phylogenetic linear models and phylogenetic generalized linear models. The computation uses an algorithm that is linear in the number of tips in the tree. The package also provides functions for simulating continuous or binary traits along the tree. Other tools include functions to test the adequacy of a population tree.

License GPL (>= 2) | file LICENSE

URL https://github.com/lamho86/phylolm

BugReports https://github.com/lamho86/phylolm/issues

Encoding UTF-8

NeedsCompilation yes

Suggests testthat, nlme

Author Lam Si Tung Ho [aut, cre],
Cecile Ane [aut],
Robert Lachlan [ctb],
Kelsey Tarpinian [ctb],
Rachel Feldman [ctb],
Qing Yu [ctb],
Wouter van der Bijl [ctb],
Joan Maspons [ctb],
Rutger Vos [ctb],
Paul Bastide [ctb]

Maintainer Lam Si Tung Ho <lamho86@gmail.com>

Repository CRAN

Date/Publication 2024-09-30 19:00:02 UTC

Contents

`phylolm-package` *Phylogenetic Linear Regression*

Description

The package provides functions for fitting phylogenetic linear models and phylogenetic generalized linear models. The computation uses an algorithm that is linear in the number of tips in the tree. The package also provides functions for simulating continuous and binary traits along the tree. Other tools include functions to test the adequacy of a population tree.

Details

Package:	phylolm
Type:	Package
Version:	2.6.5
Date:	2024-09-24
License:	GPL (>= 2)

Author(s)

Lam Si Tung Ho, Cecile Ane, Robert Lachlan, Kelsey Tarpinia, Rachel Feldman, Qing Yu, Wouter van der Bijl, Joan Maspons, Rutger Vos, Paul Bastide

Maintainer: Lam Si Tung Ho <lamho86@gmail.com>

---

flowerSize *Flower size of 25 Euphorbiaceae species*

Description

Names, flower diameters (mm) and log-transformed diameter (mm) of 25 plant species.

Usage

```
data(flowerSize)
```

Format

A data frame with 25 rows and 3 columns.

References

Davis, C.C., Latvis, M., Nickrent, D.L., Wurdack, K.J. and Baum, D.A. 2007. "Floral gigantism in Rafflesiaceae". Science **315**:1812.

---

flowerTree *Phylogenetic tree of 25 Euphorbiaceae species*

Description

A phylogenetic tree with 25 tips and 24 internal nodes.

Usage

```
data(flowerTree)
```

Format

A data frame of class phylo.

References

Davis, C.C., Latvis, M., Nickrent, D.L., Wurdack, K.J. and Baum, D.A. 2007. "Floral gigantism in Rafflesiaceae". Science **315**:1812.

guidetree *Binary population tree within Arabidopsis thaliana*

Description

Binary population tree for 29 A. thaliana accessions and A. lyrata, obtained from chromosome 4 using MDL to delimit loci, BUCKy to estimate quartet concordance factors (CFs) and Quartet Max Cut to estimate the tree topology.

Usage

```
data(guidetree)
```

Format

tree object of class phylo. Branch lengths are in coalescent units.

References

Stenz, Noah W. M., Bret Larget, David A. Baum and Cécile Ané (2015). Exploring tree-like and non-tree-like patterns using genome sequences: An example using the inbreeding plant species *Arabidopsis thaliana* (L.) Heynh. Systematic Biology, **64**(5):809-823.

TICR pipeline: github.com/nstenz/TICR

mace *Multi-task learning for ancestral state estimation.*

Description

Estimate the ancestral states of multiple traits simultaneously using a regularized maximum likelihood objective.

Usage

```
mace(trait, phy, lambda = NULL)
```

Arguments

trait	a matrix of trait values. Each row is one species and each column is a trait.
phy	a phylogenetic tree of type phylo with branch lengths.
lambda	regularizer parameter.

Details

Traits are assumed to evolve according to the Brownian motion model.

Value

a numeric vector of estimated ancestral states.

Note

The default choice for `lambda` was proposed by Ho et al. (2019).

Author(s)

Lam Si Tung Ho

References

Ho, Lam Si Tung, Vu Dinh, and Cuong V. Nguyen. "Multi-task learning improves ancestral state reconstruction." Theoretical Population Biology 126 (2019): 33-39.

Examples

```
m = 3
anc = c(0, 8, 16)
sig2 = c(1, 1, 2)
tree = rtree(50)

trait = rTrait(n = 1, phy = tree, model = "BM",
               parameters=list(ancestral.state = anc[1], sigma2 = sig2[1]))
for (i in 2:m) {
  trait = cbind(trait,rTrait(n = 1, phy = tree, model = "BM",
                             parameters=list(ancestral.state = anc[i], sigma2 = sig2[i])))
}
res = mace(trait, tree)
print(res)
```

`OU1d.loglik` *Log likelihood of an one-dimensional Ornstein-Uhlenbeck model*

Description

computes log likelihood of an one-dimensional Ornstein-Uhlenbeck model with an algorithm that is linear in the number of tips in the tree.

Usage

```
OU1d.loglik(trait, phy, model = c("OUrandomRoot", "OUfixedRoot"), parameters = NULL)
```

Arguments

trait	a vector of trait values.
phy	a phylogenetic tree of type phylo with branch lengths.
model	an Ornstein-Uhlenbeck model.
parameters	List of parameters for the model

Author(s)

Lam Si Tung Ho

Examples

```
tr = rtree(100)
alpha = 1
sigma2 = 1
sigma2_error = 0.5
ancestral.state = 0
optimal.value = 1

trait = rTrait(n = 1, tr, model = "OU",
              parameters = list(ancestral.state=ancestral.state, alpha=alpha,
                                sigma2=sigma2,sigma2_error=sigma2_error,
                                optimal.value=optimal.value))
OU1d.loglik(trait=trait, phy=tr, model="OUfixedRoot",
            parameters=list(ancestral.state=ancestral.state, alpha=alpha,sigma2=sigma2,
                            sigma2_error=sigma2_error,optimal.value=optimal.value))
```

OUshifts *Detections of shifts in the OU process along a phylogeny.*

Description

Trait data is fitted to a phylogeny using an Ornstein-Uhlenbeck (OU) process, such that the mean (or selection optimum) of the process may change in one or more edges in the tree. The number and location of changes, or shifts, is estimated using an information criterion.

Usage

```
OUshifts(y, phy, method = c("mbic", "aic", "bic", "saic", "sbic"),
         nmax, check.pruningwise = TRUE)
```

Arguments

y	values for the trait data.
phy	a phylogenetic tree of type phylo with branch lengths.
method	a method for model selection (see details below).

`nmax` maximum allowed number of shifts.

`check.pruningwise`
if TRUE, the algorithm checks if the ordering of the edges in phy are in pruningwise order.

Details

This function does not accept multivariate data (yet): `y` should be a vector named with species labels. The data `y` and the tree `phy` need to contain the same species. The user can choose among various information criteria. Each criterion seeks to minimize the value of $-2\log[\text{likelihood}(y, M)] + \text{penalty}(M)$, where M is an OU model with m shifts, placed on various edges along the phylogeny. All models use $3+m$ parameters: α, σ^2, and $m+1$ parameters to describe the expected trait values in each of the $m+1$ regimes. The AIC penalty is $2*(3+m)$. The BIC penalty is $(3+m)\log(n)$ where n is the numer of species. If one considers the position of the m shifts in the phylogeny as parameters (even though they are discrete parameters), we get the sAIC penalty $2*(3+2m)$ (used in SURFACE), and the sBIC penalty $(3+2*m)*\log(n)$. The default penalty (model = 'mbic') is defined as $3*\log(n) + (2m-1)\log(n) + \sum_{i=0}^{m}(\log(n_i))$. A lower value of `nmax` will make the search faster, but if the estimated number of shifts is found equal to nmax, then the output model is probably not optimal. Re-running with a larger `nmax` would take longer, but would likely return a more complex model with a better score.

Value

`y`	the input trait.
`phy`	the input tree.
`score`	the information criterion value of the optimal model.
`nmax`	maximum allowed number of shifts.
`nshift`	estimated number of shifts.
`alpha`	estimated selection strength of the optimal model.
`sigma2`	estimated variance of the optimal model.
`mean`	estimated the expected value of the trait in lineages without shift.
`pshift`	positions of shifts, i.e. indicies of edges where the estimated shifts occurred. The same ordering of edges is used as in phy.
`shift`	estimated shifts in the expected value of the trait.

Note

The tip labels in the tree are matched to the data names. The default choice for the parameters are as follows: `method = "mbic"`, `check.pruningwise = TRUE`

Due to unidentifiability, the parameters are the expected value of the trait and their shifts instead of the ancestral trait, the optimal values and shifts in optimal values.

Author(s)

Lam Si Tung Ho

References

Ho, L. S. T. and Ane, C. 2014. "Intrinsic inference difficulties for trait evolution with Ornstein-Uhlenbeck models". Methods in Ecology and Evolution. **5**(11):1133-1146.

Ingram, T. and Mahler, D.L. 2013. "SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with step-wise Akaike information criterion". Methods in Ecology and Evolution **4**:416-425.

Zhang, N.R. and Siegmund, D.O. 2007. "A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data". Biometrics **63**:22-32.

Examples

```
data(flowerSize)
data(flowerTree)
result <- OUshifts(flowerSize$log_transformed_size, flowerTree,
                   method = "mbic", nmax = 1)
plot.OUshifts(result,show.tip.label=FALSE)
```

OUshifts-methods *Methods for class 'OUshifts'.*

Description

These are method functions for class 'OUshifts'.

Usage

```
## S3 method for class 'OUshifts'
plot(x, show.data = TRUE, digits=3, ...)
```

Arguments

x	an object of class "OUshifts".
show.data	if TRUE, visualizes a bar plot of the data side-by-side with the phylogeny.
digits	number of digits to show in the bar plot.
...	further arguments passed to plot.phylo to plot the tree.

Author(s)

Lam Si Tung Ho, Kelsey Tarpinian

See Also

OUshifts

phyloglm *Phylogenetic Generalized Linear Model*

Description

Fits the phylogenetic logistic regression described in Ives and Garland (2010) and the Poisson regression described in Paradis and Claude (2002). The computation uses an algorithm that is linear in the number of tips in the tree.

Usage

```
phyloglm(formula, data, phy, method = c("logistic_MPLE","logistic_IG10", "logistic_MLE",
         "poisson_GEE"), btol = 10, log.alpha.bound = 4,
         start.beta=NULL, start.alpha=NULL,
         boot = 0, full.matrix = TRUE, save = FALSE)
```

Arguments

formula	a model formula.
data	a data frame containing variables in the model. If not found in data, the variables are taken from the current environment.
phy	a phylogenetic tree of type phylo with branch lengths.
method	The "logistic_IG10" method optimizes a GEE approximation to the penalized likelihood of the logistic regression. "logistic_MPLE" maximizes the penalized likelihood of the logistic regression. In both cases, the penalty is Firth's correction. The "poisson_GEE" method solves the generalized estimating equations (GEE) for Poisson regression.
btol	(logistic regression only) bound on the linear predictor to bound the searching space.
log.alpha.bound	(logistic regression only) bound for the log of the parameter alpha.
start.beta	starting values for beta coefficients.
start.alpha	(logistic regression only) starting values for alpha (phylogenetic correlation).
boot	number of independent bootstrap replicates, 0 means no bootstrap.
full.matrix	if TRUE, the full matrix of bootstrap estimates (coefficients and alpha) will be returned.
save	if TRUE, the simulated bootstrap data will be returned.

Details

This function uses an algorithm that is linear in the number of tips in the tree.

Bootstrapping can be parallelized using the future package on any future compatible back-end. For example, run library(future); plan(multiprocess)), after which bootstrapping will automatically occur in parallel. See plan for options.

Value

coefficients	the named vector of coefficients.
alpha	(logistic regression only) the phylogenetic correlation parameter.
scale	(Poisson regression only) the scale parameter which estimates the overdispersion.
sd	standard deviation for the regression coefficients.
vcov	covariance matrix for the regression coefficients.
logLik	(logistic regression only) log likelihood.
aic	(logistic regression only) AIC.
penlogLik	(logistic regression only) penalized log likelihood, using Firth's penalty for coefficients.
y	response.
n	number of observations (tips in the tree).
d	number of dependent variables.
formula	the model formula.
call	the original call to the function.
method	the estimation method.
convergence	An integer code with '0' for successful optimization. With logistic_MPLE, this is the convergence code from the optim routine.
alphaWarn	(logistic regression only) An interger code with '0' for the estimate of alpha is not near the lower and upper bounds, code with '1' for the estimate of alpha near the lower bound, code with '2' for the estimate of alpha near the upper bound.
X	a design matrix with one row for each tip in the phylogenetic tree.
bootmean	(boot > 0 only) bootstrap means of the parameters estimated.
bootsd	(boot > 0 only) bootstrap standard deviations of the estimated parameters.
bootconfint95	(boot > 0 only) bootstrap 95% confidence interval.
bootmeanAlog	(boot > 0 only) bootstrap mean of the logs of the estimated alphas.
bootsdAlog	(boot > 0 only) bootstrap standard deviation of the logs of the estimated alphas.
bootnumFailed	(boot > 0 only) number of independent bootstrap replicates for which phyloglm failed. These failures may be due to the bootstrap data having too few 0's or too few 1's.
bootstrap	(boot > 0 and full.matrix = TRUE only) matrix of all bootstrap estimates.
bootdata	(boot > 0 and save = TRUE only) matrix of all bootstrap data (each column is a dataset).

Note

The tip labels in the tree are matched to the data names (row names in the data frame). If no data frame is provided through the argument data, taxon labels in the tree are matched to taxon labels in the response variable based on the row names of the response vector, and the taxa are assumed to come in the same order for all variables in the model.

The logistic regression method of Ives and Garland (2010) uses alpha to estimate the level of phylogenetic correlation. The GEE method for Poisson regression does not estimate the level of phylogenetic correlation but takes it from the existing branch lengths in the tree.

The standard deviation and the covariance matrix for the coefficients of logistic regression are conditional on the estimated value of the phylogenetic correlation parameter α.

The default choice `btol=10` constrains the fitted values, i.e. the probability of "1" predicted by the model, to lie within $1/(1+e^{10}) = 0.000045$ and $1/(1+e^{-10}) = 0.999955$.

The log of α is bounded in the interval $-\log(T) \pm \log.\text{alpha.bound}$ where T is the mean of the distances from the root to tips. In other words, αT is constrained to lie within $\exp(\pm \log.\text{alpha.bound})$.

Author(s)

Lam Si Tung Ho, Robert Lachlan, Rachel Feldman and Cecile Ane

References

Ho, L. S. T. and Ane, C. 2014. "A linear-time algorithm for Gaussian and non-Gaussian trait evolution models". Systematic Biology **63**(3):397-408.

Ives, A. R. and T. Garland, Jr. 2010. "Phylogenetic logistic regression for binary dependent variables". Systematic Biology **59**:9-26.

Paradis E. and Claude J. 2002. "Analysis of Comparative Data Using Generalized Estimating Equations". Journal of Theoretical Biology **218**:175-185.

See Also

`compar.gee`.

Examples

```
set.seed(123456)
tre = rtree(50)
x = rTrait(n=1,phy=tre)
X = cbind(rep(1,50),x)
y = rbinTrait(n=1,phy=tre, beta=c(-1,0.5), alpha=1 ,X=X)
dat = data.frame(trait01 = y, predictor = x)
fit = phyloglm(trait01~predictor,phy=tre,data=dat,boot=100)
summary(fit)
coef(fit)
vcov(fit)
```

---

phyloglm-methods *Methods for class 'phyloglm'.*

Description

These are method functions for class 'phyloglm'.

Usage

```
## S3 method for class 'phyloglm'
print(x, digits = max(3, getOption("digits") - 3), ...)
## S3 method for class 'phyloglm'
summary(object, ...)
## S3 method for class 'phyloglm'
residuals(object,type = c("response"), ...)
## S3 method for class 'phyloglm'
vcov(object, ...)
## S3 method for class 'phyloglm'
nobs(object, ...)
## S3 method for class 'phyloglm'
logLik(object, ...)
## S3 method for class 'phyloglm'
AIC(object, k=2, ...)
## S3 method for class 'phyloglm'
plot(x, ...)
```

Arguments

x	an object of class "phyloglm".
object	an object of class "phyloglm".
digits	number of digits to show in summary method.
type	Currently, only the "response" type is implemented. It returns the raw residuals, that is, the differences between the observed responses and the predicted values. They are phylogenetically correlated.
k	numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC.
...	further arguments to methods.

Author(s)

Lam Si Tung Ho

See Also

phyloglm

---

phyloglmstep *Stepwise model selection for Phylogenetic Generalized Linear Model*

Description

Performs stepwise model selection for phylogenetic generalized linear models, using the criterion -2*log-likelihood + k*npar, where npar is the number of estimated parameters and k=2 for the usual AIC.

Usage

```
phyloglmstep(formula, starting.formula = NULL, data=list(), phy,
     method = c("logistic_MPLE","logistic_IG10", "logistic_MLE", "poisson_GEE"),
       direction = c("both", "backward", "forward"), trace = 2,
       btol = 10, log.alpha.bound = 4, start.beta=NULL,
       start.alpha=NULL, boot = 0, full.matrix = TRUE,
       k=2, ...)
```

Arguments

formula	formula of the full model.
starting.formula	optional formula of the starting model.
data	a data frame containing variables in the model. If not found in data, the variables are taken from current environment.
phy	a phylogenetic tree of type phylo with branch lengths.
method	The "logistic_IG10" method optimizes a GEE approximation to the penalized likelihood of the logistic regression. "logistic_MPLE" maximizes the penalized likelihood of the logistic regression. In both cases, the penalty is Firth's correction.
direction	direction for stepwise search, can be both, forward, and backward.
trace	if positive, information on each searching step is printed. Larger values may give more detailed information.
btol	bound on the linear predictor to bound the searching space.
log.alpha.bound	bound for the log of the parameter alpha.
start.beta	starting values for beta coefficients.
start.alpha	starting values for alpha (phylogenetic correlation).
boot	number of independent bootstrap replicates, 0 means no bootstrap.
full.matrix	if TRUE, the full matrix of bootstrap estimates (coefficients and alpha) will be returned.
k	optional weight for the penalty.
...	further arguments to be passed to the function optim.

Details

The default $k = 2$ corresponds to the usual AIC penalty. Use $k = \log(n)$ for the usual BIC, although it is unclear how BIC should be defined for phylogenetic regression.

See phyloglm for details on the possible phylogenetic methods for the error term, for default bounds on the phylogenetic signal parameters, or for matching tip labels between the tree and the data.

Value

A phyloglm object correponding to the best model is returned.

Author(s)

Rutger Vos

See Also

phyloglm.

Examples

```
set.seed(123456)
tre = rcoal(60)
taxa = sort(tre$tip.label)
b0=0; b1=1;
x1 = rTrait(phy=tre,model="BM",
           parameters=list(ancestral.state=0,sigma2=10))
x2 = rTrait(phy=tre,model="BM",
            parameters=list(ancestral.state=0,sigma2=10))
x3 = rTrait(phy=tre,model="BM",
            parameters=list(ancestral.state=0,sigma2=10))
X = cbind(rep(1,60), x1)
y = rbinTrait(n=1,phy=tre, beta=c(-1,0.5), alpha=1 ,X=X)
dat = data.frame(trait=y[taxa],pred1=x1[taxa],pred2=x2[taxa],pred3=x3[taxa])
fit = phyloglmstep(trait~pred1+pred2+pred3,data=dat,phy=tre,method="logistic_MPLE",direction="both")
summary(fit)
```

---

phylolm *Phylogenetic Linear Model*

Description

Fits a phylogenetic linear regression model. The likelihood is calculated with an algorithm that is linear in the number of tips in the tree.

Usage

```
phylolm(formula, data = list(), phy, model = c("BM", "OUrandomRoot",
       "OUfixedRoot", "lambda", "kappa", "delta", "EB", "trend"),
       lower.bound = NULL, upper.bound = NULL,
       starting.value = NULL, measurement_error = FALSE,
       boot=0,full.matrix = TRUE, save = FALSE, REML = FALSE, ...)
```

Arguments

formula	a model formula.
data	a data frame containing variables in the model. If not found in data, the variables are taken from current environment.
phy	a phylogenetic tree of type phylo with branch lengths.

model	a model for the covariance (see Details).
lower.bound	optional lower bound for the optimization of the phylogenetic model parameter.
upper.bound	optional upper bound for the optimization of the phylogenetic model parameter.
starting.value	optional starting value for the optimization of the phylogenetic model parameter.
measurement_error	a logical value indicating whether there is measurement error sigma2_error (see Details).
boot	number of independent bootstrap replicates, 0 means no bootstrap.
full.matrix	if TRUE, the full matrix of bootstrap estimates (coefficients and covariance parameters) will be returned.
save	if TRUE, the simulated bootstrap data will be returned.
REML	Use ML (default) or REML for estimating the parameters.
...	further arguments to be passed to the function optim.

Details

This function uses an algorithm that is linear in the number of tips in the tree to calculate the likelihood. Possible phylogenetic models for the error term are the Brownian motion model (BM), the Ornstein-Uhlenbeck model with an ancestral state to be estimated at the root (OUfixedRoot), the Ornstein-Uhlenbeck model with the ancestral state at the root having the stationary distribution (OUrandomRoot), Pagel's λ model (lambda), Pagel's κ model (kappa), Pagel's δ model (delta), the early burst model (EB), and the Brownian motion model with a trend (trend).

Using measurement error means that the covariance matrix is taken to be $\sigma^2 * V + \sigma^2_{error} * I$ where V is the phylogenetic covariance matrix from the chosen model, I is the identity matrix, and σ^2_{error} is the variance of the measurement error (which could include environmental variability, sampling error on the species mean, etc.).

By default, the bounds on the phylogenetic parameters are $[10^{-7}/T, 50/T]$ for α, $[10^{-7}, 1]$ for λ, $[10^{-6}, 1]$ for κ, $[10^{-5}, 3]$ for δ and $[-3/T, 0]$ for rate, where T is the mean root-to-tip distance. $[10^{-16}, 10^{16}]$ for the ratio sigma2_error/sigma2 (if measurement errors is used).

Bootstrapping can be parallelized using the future package on any future compatible back-end. For example, run library(future); plan(multiprocess)), after which bootstrapping will automatically occur in parallel. See plan for options.

Value

coefficients	the named vector of coefficients.
sigma2	the maximum likelihood estimate of the variance rate σ^2.
sigma2_error	the maximum likelihood estimate of the variance of the measurement errors.
optpar	the optimized value of the phylogenetic correlation parameter (alpha, lambda, kappa, delta or rate).
logLik	the maximum of the log likelihood.
p	the number of all parameters of the model.
aic	AIC value of the model.

vcov	covariance matrix for the regression coefficients, given the phylogenetic correlation parameter (if any).
fitted.values	fitted values
residuals	raw residuals
y	response
X	design matrix
n	number of observations (tips in the tree)
d	number of dependent variables
mean.tip.height	mean root-to-tip distance, which can help choose good starting values for the correlation parameter.
formula	the model formula
call	the original call to the function
model	the phylogenetic model for the covariance
bootmean	(boot > 0 only) bootstrap means of the parameters estimated.
bootsd	(boot > 0 only) bootstrap standard deviations of the estimated parameters.
bootconfint95	(boot > 0 only) bootstrap 95% confidence interval.
bootmeansdLog	(boot > 0 only) bootstrap mean and standard deviation of the logs of the estimated covariance parameters.
bootnumFailed	(boot > 0 only) number of independent bootstrap replicates for which phylolm failed.
bootstrap	(boot > 0 and full.matrix = TRUE only) matrix of all bootstrap estimates.
bootdata	(boot > 0 and save = TRUE only) matrix of all bootstrap data (each column is a dataset).
r.squared	The r^2 for the model.
adj.r.squared	The adjusted r^2 for the model.

Note

The tip labels in the tree are matched to the data names (row names in the data frame). If no data frame is provided through the argument data, taxon labels in the tree are matched to taxon labels in the response variable based on the row names of the response vector, and the taxa are assumed to come in the same order for all variables in the model.

For the delta model, the tree is rescaled back to its original height after each node's distance from the root is raised to the power delta. This is to provide a stable estimate of the variance parameter σ^2. For non-ultrametric trees, the tree is rescaled to maintain the longest distance from the root to its original value.

The trend model can only be used with non-ultrametric trees. For this model, one predictor variable is added to the model whose values are the distances from the root to every tip of the tree. The estimate of the coefficent for this variable forms the trend value.

Pagel's λ model and measurement error cannot be used together: the parameters λ, σ^2 and σ^2_{error} are not distinguishable (identifiable) from each other.

Author(s)

Lam Si Tung Ho

References

Ho, L. S. T. and Ane, C. 2014. "A linear-time algorithm for Gaussian and non-Gaussian trait evolution models". Systematic Biology **63**(3):397-408.

Butler, M. A. and King, A. A. 2004. "Phylogenetic comparative analysis: A modeling approach for adaptive evolution". The American Naturalist **164**:683-695.

Hansen, T. F. 1997. "Stabilizing selection and the comparative analysis of adaptation". Evolution **51**:1341-1351.

Harmon, L. J. et al. 2010. "Early bursts of body size and shape evolution are rare in comparative data". Evolution **64**:2385-2396.

Ho, L. S. T. and Ane, C. 2013. "Asymptotic theory with hierarchical autocorrelation: Ornstein-Uhlenbeck tree models". The Annals of Statistics **41**(2):957-981.

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.

See Also

corBrownian, corMartins, corPagel, fitContinuous, pgls.

Examples

```
set.seed(123456)
tre = rcoal(60)
taxa = sort(tre$tip.label)
b0=0; b1=1;
x <- rTrait(n=1, phy=tre,model="BM",
            parameters=list(ancestral.state=0,sigma2=10))
y <- b0 + b1*x +
     rTrait(n=1,phy=tre,model="lambda",parameters=list(
               ancestral.state=0,sigma2=1,lambda=0.5))
dat = data.frame(trait=y[taxa],pred=x[taxa])
fit = phylolm(trait~pred,data=dat,phy=tre,model="lambda")
summary(fit)

# adding measurement errors and bootstrap
z <- y + rnorm(60,0,1)
dat = data.frame(trait=z[taxa],pred=x[taxa])
fit = phylolm(trait~pred,data=dat,phy=tre,model="BM",measurement_error=TRUE,boot=100)
summary(fit)
```

phylolm-methods *Methods for class 'phylolm'.*

Description

These are method functions for class 'phylolm'.

Usage

```
## S3 method for class 'phylolm'
print(x, digits = max(3, getOption("digits") - 3), ...)
## S3 method for class 'phylolm'
summary(object, ...)
## S3 method for class 'phylolm'
nobs(object, ...)
## S3 method for class 'phylolm'
residuals(object,type = c("response"), ...)
## S3 method for class 'phylolm'
predict(object, newdata = NULL, se.fit = FALSE, ...)
## S3 method for class 'phylolm'
vcov(object, ...)
## S3 method for class 'phylolm'
logLik(object, ...)
## S3 method for class 'phylolm'
AIC(object, k=2, ...)
## S3 method for class 'phylolm'
plot(x, ...)
## S3 method for class 'phylolm'
confint(object, parm, level=0.95, ...)
## S3 method for class 'phylolm'
model.frame(formula, ...)
```

Arguments

x	an object of class "phylolm".
object	an object of class "phylolm".
formula	an object of class "phylolm".
digits	number of digits to show in summary method.
type	Currently, only the "response" type is implemented. It returns the raw residuals, that is, the differences between the observed responses and the predicted values. They are phylogenetically correlated.
newdata	an optional data frame to provide the predictor values at which predictions should be made. If omitted, the fitted values are used. Currently, predictions are made for new species whose placement in the tree is unknown. Only their covariate information is used. The prediction for the trend model is not currently implemented.

se.fit	A switch indicating if standard errors are required.
k	numeric, the penalty per parameter to be used; the default k = 2 is the classical AIC.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required.
...	further arguments to methods.

Author(s)

Lam Si Tung Ho

See Also

phylolm

Examples

```
set.seed(321123)
tre = rcoal(50)
y = rTrait(n=1,phy=tre,model="BM")
fit = phylolm(y~1,phy=tre,model="BM")
summary(fit)
vcov(fit)
```

---

phylostep *Stepwise model selection for Phylogenetic Linear Model*

Description

Performs stepwise model selection for phylogenetic linear models, using the criterion -2*log-likelihood + k*npar, where npar is the number of estimated parameters and k=2 for the usual AIC.

Usage

```
phylostep(formula, starting.formula = NULL, keeping.formula = NULL, data = list(),
       phy, model = c("BM", "OUrandomRoot","OUfixedRoot",
       "lambda", "kappa", "delta", "EB", "trend"),
       direction = c("both", "backward", "forward"), trace = 2,
       lower.bound = NULL, upper.bound = NULL,
       starting.value = NULL, k=2, ...)
```

Arguments

formula	formula of the full model.
starting.formula	optional formula of the starting model.
keeping.formula	optional formula of the keeping model. Covariates of the keeping model are always included in the model.
data	a data frame containing variables in the model. If not found in data, the variables are taken from current environment.
phy	a phylogenetic tree of type phylo with branch lengths.
model	a model for the phylogenetic covariance of residuals.
direction	direction for stepwise search, can be both, forward, and backward.
trace	if positive, information on each searching step is printed. Larger values may give more detailed information.
lower.bound	optional lower bound for the optimization of the phylogenetic model parameter.
upper.bound	optional upper bound for the optimization of the phylogenetic model parameter.
starting.value	optional starting value for the optimization of the phylogenetic model parameter.
k	optional weight for the penalty.
...	further arguments to be passed to the function optim.

Details

The default $k = 2$ corresponds to the usual AIC penalty. Use $k = \log(n)$ for the usual BIC, although it is unclear how BIC should be defined for phylogenetic regression.

See phylolm for details on the possible phylogenetic models for the error term, for default bounds on the phylogenetic signal parameters, or for matching tip labels between the tree and the data.

Value

A phylolm object correponding to the best model is returned.

Author(s)

Lam Si Tung Ho and Cecile Ane

See Also

phylolm.

Examples

```
set.seed(123456)
tre = rcoal(60)
taxa = sort(tre$tip.label)
b0=0; b1=1;
x1 = rTrait(phy=tre,model="BM",
```

```
            parameters=list(ancestral.state=0,sigma2=10))
x2 = rTrait(phy=tre,model="BM",
            parameters=list(ancestral.state=0,sigma2=10))
x3 = rTrait(phy=tre,model="BM",
            parameters=list(ancestral.state=0,sigma2=10))
y <- b0 + b1*x1 +
     rTrait(n=1,phy=tre,model="BM",parameters=list(
               ancestral.state=0,sigma2=1))
dat = data.frame(trait=y[taxa],pred1=x1[taxa],pred2=x2[taxa],pred3=x3[taxa])
fit = phylostep(trait~pred1+pred2+pred3,data=dat,phy=tre,model="BM",direction="both")
summary(fit)
```

`pruningwise.branching.times` *Calculates internal node ages in an ultrametric "pruningwise" tree*

Description

Calculates the branching times, or ages, of all internal nodes in an ultrametric tree whose internal representation is in "pruningwise" order.

Usage

```
pruningwise.branching.times(phy)
```

Arguments

phy	an ultrametric phylogenetic tree of type phylo with branch lengths, already in "pruningwise" order.

Value

a vector of node ages, with the original internal node names if those were available in phy, or otherwise named by the node numbers in phy.

Author(s)

Lam Si Tung Ho

See Also

pruningwise.distFromRoot, branching.times.

Examples

```
tre = reorder(rcoal(50),"pruningwise")
pruningwise.branching.times(tre)
```

pruningwise.distFromRoot *Calculates node distance from the root in an "pruningwise" tree*

Description

Calculates the distance from the root to all nodes, in a tree whose internal representation is in "pruningwise" order.

Usage

```
pruningwise.distFromRoot(phy)
```

Arguments

phy	a phylogenetic tree of type phylo with branch lengths, already in "pruningwise" order.

Value

a vector of distances, with the original tip labels and internal node names if internal node names were available, or otherwise named by the node numbers in phy.

Author(s)

Lam Si Tung Ho

See Also

pruningwise.branching.times, cophenetic.

Examples

```
tre = reorder(rtree(50),"pruningwise")
pruningwise.distFromRoot(tre)
```

quartetCF *Quartet concordance factors across Arabidopsis thaliana*

Description

Concordance factors of quartets for 29 A. thaliana accessions and A. lyrata, obtained from chromosome 4 using MDL to delimit loci then BUCKy on each 4-taxon set.

Usage

```
data(quartetCF)
```

Format

Data frame with 7 variables and 27,405 rows. Each row corresponds to one 4-taxon set (choosing 4 taxa out of 30 makes 27,405 combinations). The first four variables, named 'taxon1' through 'taxon4', give the names of the 4 taxa in each 4-taxon set. Variables 5 through 7 are named CF12.34, CF13.24 and CF14.23, and give the estimated concordance factors of the 3 quartets on each set of 4 taxa: taxon 1 + taxon 2 versus taxon3 + taxon 4, etc. These concordance factors are the proportion of loci that have a given quartet tree. They were obtained from chromosome 4 using MDL to delimit loci then BUCKy to estimate quartet concordance factors (CFs).

References

Stenz, Noah W. M., Bret Larget, David A. Baum and Cécile Ané (2015). Exploring tree-like and non-tree-like patterns using genome sequences: An example using the inbreeding plant species *Arabidopsis thaliana* (L.) Heynh. Systematic Biology, **64**(5):809-823.

TICR pipeline: github.com/nstenz/TICR

---

rbinTrait *Binary trait simulation*

Description

Simulates a binary trait along a phylogeny, according to the model in Ives and Garland (2010).

Usage

```
rbinTrait(n=1, phy, beta, alpha, X = NULL, model = c("LogReg"))
```

Arguments

n	number of independent replicates.
phy	a phylogenetic tree of type phylo with brach lengths.
beta	a vector of coefficients for the logistic regression model.
alpha	the phylogenetic correlation parameter.
X	a design matrix with one row for each tip in the phylogenetic tree.
model	Currently, only phylogenetic logistic regression is implemented.

Value

If n=1, a numeric vector of 0-1 values with names from the tip labels in the tree. For more than 1 replicate, a matrix with the tip labels as row names, and one column per replicate.

Note

In the absence of a design matrix X, a single intercept is used. In this case beta should be a vector of length one and the model reduces to a 2-state Markov process on the tree with stationary mean $e^{\beta}/(1+e^{\beta})$. If a design matrix X is provided, the length of beta should be equal to the number of columns in X.

Author(s)

Lam Si Tung Ho and C. An?

References

Ives, A. R. and T. Garland, Jr. 2010. "Phylogenetic logistic regression for binary dependent variables". Systematic Biology **59**:9-26.

See Also

rTrait.

Examples

```
tre = rtree(50)
x = rTrait(n=1,phy=tre)
X = cbind(rep(1,50),x)
y = rbinTrait(n=1, phy=tre, beta=c(-1,0.5), alpha=1, X=X)
```

---

rTrait *Continuous trait simulation*

Description

Simulates a continuous trait along a tree from various phylogenetic models.

Usage

```
rTrait(n=1, phy, model=c("BM","OU","lambda","kappa","delta","EB","trend"),
       parameters = NULL, plot.tree=FALSE)
```

Arguments

n	number of independent replicates
phy	a phylogenetic tree of type phylo with branch lengths.
model	a phylogenetic model. Default is "BM", for Brownian motion. Alternatives are "OU", "lambda", "kappa", "delta", "EB" and "trend".
parameters	List of parameters for the model (see Note).
plot.tree	If TRUE, the tree with transformed branch lengths will be shown, except for the OU model.

Details

Possible phylogenetic models are the Brownian motion model (BM), the Ornstein-Uhlenbeck model (OU), Pagel's λ model (lambda), Pagel's κ model (kappa), Pagel's δ model (delta), the early burst model (EB), and the Brownian motion model with a trend (trend).

Value

If n=1, a numeric vector with names from the tip labels in the tree. For more than 1 replicate, a matrix with the tip labels as row names, and one column per replicate.

Note

The default choice for the parameters are as follows: ancestral.state=0, sigma2=1, optimal.value=0 for the OU model, alpha=0 for the selection strength in the OU model, lambda=1, kappa=1, delta=1, rate=0 for the EB model, trend=0. These default choices correspond to the BM model.

Author(s)

Lam Si Tung Ho and C. Ane

See Also

rTraitCont.

Examples

```
tre = rtree(50)
y = rTrait(n=1, phy=tre, model="OU",
           parameters=list(optimal.value=2,sigma2=1,alpha=0.1))
```

---

stepwise.test.tree *Fits a population tree to data from quartet concordance factors*

Description

From a set of quartet concordance factors obtained from genetic data (proportion of loci that truly have a given quartet) and from a guide tree, this functions uses a stepwise search to find the best resolution of that guide tree. Any unresolved edge corresponds to ancestral panmixia, on which the coalescent process is assumed.

Usage

```
stepwise.test.tree(cf, guidetree, search="both", method="PLL", kbest=5,
                   maxiter=100, startT="panmixia", shape.correction=TRUE)
```

Arguments

cf	data frame containing one row for each 4-taxon set and containing taxon names in columns 1-4, and concordance factors in columns 5-7.
guidetree	tree of class phylo on the same taxon set as those in cf, with branch lengths in coalescent units.
search	one of "both" (stepwise search both forwards and backwards at each step), or "heuristic" (heuristic shallow search: not recommended).

`method` Only "PLL" is implemented. The scoring criterion to rank population trees is the pseudo log-likelihood (ignored if search="heuristic").

`kbest` Number of candidate population trees to consider at each step for the forward and for the backward phase (separately). Use a lower value for faster but less thorough search.

`maxiter` Maximum number of iterations. One iteration consists of considering multiple candidate population trees, using both a forward step and a backward step.

`startT` starting population tree. One of "panmixia", "fulltree", or a numeric vector of edge numbers to keep resolved. The other edges are collapsed for panmixia.

`shape.correction`
boolean. If true, the shapes of all Dirichlet distributions used to test the adequacy of a population tree are corrected to be greater or equal to 1. This correction avoids Dirichlet densities going near 0 or 1. It is applied both when the α parameter is estimated and when the outlier p-values are calculated.

Value

`Nedge` Number of edges kept resolved in the guide tree. Other edges are collapsed to model ancestral panmixia.

`edges` Indices of edges kept resolved in the guide tree.

`notincluded` Indices of edges collapsed in the guide tree, to model ancestral panmixia.

`alpha` estimated α parameter.

`negPseudoLoglik`
Negative pseudo log-likelihood of the final estimated population tree.

`X2` Chi-square statistic, from comparing the counts of outlier p-values (in `outlier.table`) to the expected counts.

`chisq.pval` p-value from the chi-square test, obtained from the comparing the `X2` value to a chi-square distribution with 3 df.

`chisq.conclusion`
character string. If the chi-square test is significant, this statement says if there is an excess (or deficit) of outlier 4-taxon sets.

`outlier.table` Table with 2 rows (observed and expected counts) and 4 columns: number of 4-taxon sets with p-values $p \leq 0.01$, $0.01 < p \leq 0.05$, $0.05 < p \leq 0.10$ or $p > 0.10$.

`outlier.pvalues`
Vector of outlier p-values, with as many entries as there are rows in `cf`, one for each set of 4 taxa.

`cf.exp` Matrix of concordance factors expected from the estimated population tree, with as many rows as in `cf` (one row for each 4-taxon set) and 3 columns (one for each of the 3 possible quartet trees).

Author(s)

Cécile Ané

References

Stenz, Noah W. M., Bret Larget, David A. Baum and Cécile Ané (2015). Exploring tree-like and non-tree-like patterns using genome sequences: An example using the inbreeding plant species *Arabidopsis thaliana* (L.) Heynh. Systematic Biology, **64**(5):809-823.

See Also

test.one.species.tree.

Examples

```
data(quartetCF)
data(guidetree)
resF <- stepwise.test.tree(quartetCF,guidetree,startT="fulltree") # takes ~ 1 min
resF[1:9]
```

---

test.one.species.tree *Tests the fit of a population tree to quartet concordance factor data*

Description

From a set of quartet concordance factors obtained from genetic data (proportion of loci that truly have a given quartet), this function tests the adequacy of the coalescent process on a given population tree, where branch lengths indicate coalescent units.

Usage

```
test.one.species.tree(cf, guidetree, prep, edge.keep,
                      plot=TRUE, shape.correction = TRUE)
```

Arguments

cf	data frame containing one row for each 4-taxon set, with taxon names in columns 1-4 and concordance factors in columns 5-7.
guidetree	tree of class phylo on the same taxon set as those in cf, with branch lengths in coalescent units.
prep	result of test.tree.preparation(cf,guidetree). If the test is repeated multiple times on various resolutions of the guide tree (see edge.keep), it saves time to do this only once.
edge.keep	Indices of edges to keep in the guide tree. All other edges are collapsed to reflect ancestral panmixia. In the tested population tree, the collapsed edges have length set to 0.
plot	boolean. If TRUE, a number of plots are output.

`shape.correction`
: boolean. If TRUE, the shapes of all Dirichlet distributions are corrected to be greater or equal to 1. This correction avoids Dirichlet densities going near 0 or 1. It applies when the α parameter is estimated and when the outlier p-values are calculated.

Value

`alpha`
: estimated α parameter.

`negPseudoLoglik`
: Negative pseudo log-likelihood of the population tree.

`X2`
: Chi-square statistic, from comparing the counts of outlier p-values (in `outlier.table`) to the expected counts.

`chisq.pval`
: p-value from the chi-square test, obtained from the comparing the `X2` value to a chi-square distribution with 3 df.

`chisq.conclusion`
: character string. If the chi-square test is significant, this statement says if there is an excess (or deficit) of outlier 4-taxon sets.

`outlier.table`
: Table with 2 rows (observed and expected counts) and 4 columns: number of 4-taxon sets with p-values $p \leq 0.01$, $0.01 < p \leq 0.05$, $0.05 < p \leq 0.10$ or $p > 0.10$.

`outlier.pvalues`
: Vector of outlier p-values, with as many entries as there are rows in `cf`, one for each set of 4 taxa.

`cf.exp`
: Matrix of concordance factors expected from the estimated population tree, with as many rows as in `cf` (one row for each 4-taxon set) and 3 columns (one for each of the 3 possible quartet trees).

Author(s)

Cécile Ané

References

Stenz, Noah W. M., Bret Larget, David A. Baum and Cécile Ané (2015). Exploring tree-like and non-tree-like patterns using genome sequences: An example using the inbreeding plant species *Arabidopsis thaliana* (L.) Heynh. Systematic Biology, **64**(5):809-823.

See Also

`stepwise.test.tree`, `test.tree.preparation`.

Examples

```
data(quartetCF)
data(guidetree)
prelim <- test.tree.preparation(quartetCF,guidetree) # takes 5-10 seconds
```

```
# test of panmixia: all edges collapsed, none resolved.
panmixia <- test.one.species.tree(quartetCF,guidetree,prelim,edge.keep=NULL)
panmixia[1:6]

# test of full tree: all internal edges resolved, none collapsed.
Ntaxa = length(guidetree$tip.label)
# indices of internal edges:
internal.edges = which(guidetree$edge[,2] > Ntaxa)
fulltree <- test.one.species.tree(quartetCF,guidetree,prelim,edge.keep=internal.edges)
fulltree[1:6]

# test of a partial tree, some edges (but not all) collapsed
edges2keep <- c(1,2,4,6,7,8,11,14,20,21,23,24,31,34,35,36,38,39,44,47,53)
partialTree <- test.one.species.tree(quartetCF,guidetree,prelim,edge.keep=edges2keep)
partialTree[1:5]
partialTree$outlier.table
# identify taxa most responsible for the extra outlier quartets
outlier.4taxa <- which(partialTree$outlier.pvalues < 0.01)
length(outlier.4taxa) # 483 4-taxon sets with outlier p-value below 0.01
q01 = as.matrix(quartetCF[outlier.4taxa,1:4])
sort(table(as.vector(q01)),decreasing=TRUE)
# So: Cnt_1 and Vind_1 both appear in 239 of these 483 outlier 4-taxon sets.
sum(apply(q01,1,function(x){"Cnt_1" %in% x | "Vind_1" %in% x}))
# 266 outlier 4-taxon sets have either Cnt_1 or Vind_1
sum(apply(q01,1,function(x){"Cnt_1" %in% x & "Vind_1" %in% x}))
# 212 outlier 4-taxon sets have both Cnt_1 and Vind_1
```

`test.tree.preparation` *data structure preparation for testing a population tree*

Description

Takes a guide tree and quartet concordance factor data, and makes preliminary calculations to speed up the test of adequacy of a population tree with `test.one.species.tree`.

Usage

```
test.tree.preparation(cf, guidetree)
```

Arguments

`cf`	data frame containing one row for each 4-taxon set, with taxon names in columns 1-4.
`guidetree`	tree of class phylo on the same taxon set as those in `cf`.

Value

`quartet2edge` matrix of booleans with as many rows as in `cf` (one row for each 4-taxon set) and with as many columns as edges in `guidetree`. For a given set of 4 taxa and for edge `i`, the corresponding entry is true if the guide tree pruned to the 4 taxa includes edge `i` as part of its internal edge.

`dominant` Vector with as many entries as rows in `cf`, one for each 4-taxon set. Entries are 1, 2 or 3 depending on the quartet displayed in the guide tree: 1 for the quartet 12|34 (i.e. taxon1 + taxon2 versus taxon3 + taxon4), 2 for the quartet 13|24 and 3 for the quartet 14|23.

See Also

`test.one.species.tree`.

`three.point.compute` *Computations with a (generalized) three-point structured tree*

Description

Computes $P'V^{-1}Q$ and the $\log(\det V)$ of a (generalized) three-point structured matrix.

Usage

```
three.point.compute(phy, P, Q = NULL, diagWeight = NULL,
          check.pruningwise = TRUE, check.names = TRUE, check.precision = TRUE)
```

Arguments

`phy` a rooted phylogenetic tree of type phylo with branch lengths, to represent the 3-point structured matrix V_0. Note that the matrix of interest is $V = DV_0D$.

`P, Q` two matrices.

`diagWeight` a vector containing the diagonal elements of the diagonal matrix D if V has a generalized 3-point structure: $V = DV_0D$

`check.pruningwise`
If FALSE, the tree is assumed to be in pruningwise order.

`check.names` if FALSE, the row names of `P`, `Q`, and the names of `diagWeight` are assumed to be the same as the labels of the tips in the tree.

`check.precision`
if FALSE, diagWeight will be allowed to be below Machine epsilon. Recommended to remain TRUE.

Value

vec11	$1'V^{-1}1$.
P1	$P'V^{-1}1$.
PP	$P'V^{-1}P$.
Q1	$Q'V^{-1}1$.
QQ	$Q'V^{-1}Q$.
PQ	$P'V^{-1}Q$.
logd	$\log(\det V)$.

Note

The matrix V is assumed to be $V = DV_0D$ where D is the diagonal matrix with non-zero diagonal elements in diagWeight, and where V_0 is the 3-point structured covariance matrix determined by phy and its branch lengths. Note that D do not correspond to measurement error terms.

The number of rows in P and Q and the length of diagWeight need to be the same as the number of tips in the tree. When Q = NULL, the function only returns $1'V^{-1}1$, $P'V^{-1}1$ and $P'V^{-1}P$.

Author(s)

Lam Si Tung Ho, Robert Lachlan

References

Ho, L. S. T. and An?, C. (2014). "A linear-time algorithm for Gaussian and non-Gaussian trait evolution models". Systematic Biology **63**(3):397-408.

See Also

transf.branch.lengths.

Examples

```
tre1 = rtree(500)
tre2 = transf.branch.lengths(phy=tre1, model="OUrandomRoot",
                             parameters = list(alpha = 0.5))
Q = rTrait(n=2,tre1)
y = rTrait(n=1,tre1)
P = cbind(1,y)
three.point.compute(tre2$tree,P,Q,tre2$diagWeight)
```

`transf.branch.lengths` *Creates a tree with branch lengths to represent the 3-point structure of a covariance matrix*

Description

Creates a phylogenetic tree with branch lengths and a diagonal matrix to represent a (generalized) 3-point structured covariance matrix from a trait evolution model on a known tree.

Usage

```
transf.branch.lengths(phy, model = c("BM", "OUrandomRoot",
       "OUfixedRoot", "lambda", "kappa", "delta", "EB", "trend"),
       parameters = NULL, check.pruningwise = TRUE,
       check.ultrametric=TRUE, D=NULL, check.names = TRUE)
```

Arguments

`phy`	a phylogenetic tree of type phylo with branch lengths.
`model`	a phylogenetic model. Default is "BM", for Brownian motion. Alternatives are "OU", "lambda", "kappa", "delta", "EB" and "trend".
`parameters`	List of parameters for the model (see Note).
`check.pruningwise`	if FALSE, the tree is assumed to be in pruningwise order.
`check.ultrametric`	if FALSE, the tree is assumed to be ultrametric and `D` needs to be provided. This is used for the OU transformations only.
`D`	vector of ajustments for the external edge lengths, to make the tree ultrametric. Used for the OU transformations only.
`check.names`	`D` needs to have names that match tip labels unless check.names=FALSE, in which case the elements in `D` are assumed to come in the same order as tip labels in the tree.

Details

Possible phylogenetic models are the Brownian motion model (BM), the Ornstein-Uhlenbeck model (OU), Pagel's lambda model (lambda), Pagel's kappa model (kappa), Pagel's delta model (delta), the early burst model (EB), and the Brownian motion with a trend (trend). Edge lengths are unchanged under BM and the trend model. Under the kappa model, each branch length ℓ is transformed to ℓ^κ. If the time from the root to a node is t in phy, it is transformed to $T * (t/T)^\delta$ under the delta model, where T is the maximum root-to-tip distance. The transformed tree has the same T. Under EB, t is transformed to $(e^{\mathrm{rate}*t} - 1)/\mathrm{rate}$, which is very close to t (i.e. to the BM model) when `rate` is close to 0. Under the lambda model, the time t from the root to a node is transformed to λt for an internal node and is unchanged for a tip. Under "OUrandomRoot", t is transformed to $\exp(-2\alpha(T-t))$, where T is now the mean root-to-tip distance. Under "OUfixedRroot", t is transformed to $\exp(-2\alpha(T-t))(1-\exp(-2\alpha t))$. Under the OU models, `diagWeight` contains $\exp(\alpha D_i)$ for tip i, where D_i is the extra length added to tip i to make the tree ultrametric.

Value

`tree`	a rooted tree with a root edge and transformed branch lengths.
`diagWeight`	a vector containing the diagonal elements of the diagonal matrix for the generalized 3-point structure.

Note

The default choice for the parameters are as follows: `alpha=0` for the selection strength in the OU model, `lambda=1`, `kappa=1`, `delta=1`, `rate=0` for the EB model, `sigma2_error=0` for the variance of measurement errors. These default choices correspond to the BM model.

Edges in the output tree are in pruningwise order.

If `model="BM"` or `model="trend"`, the output tree is the same as the input tree except that the output tree is in pruningwise order.

Author(s)

Lam Si Tung Ho

References

Ho, L. S. T. and Ane, C. *A linear-time algorithm for Gaussian and non-Gaussian trait evolution models*. Systematic Biology **63**(3):397-408.

See Also

`three.point.compute`.

Examples

```
set.seed(123456)
tre1 = rcoal(10)
tre2 = transf.branch.lengths(phy=tre1, model="OUrandomRoot",
                             parameters = list(alpha=1))
par(mfrow = c(2,1))
plot(tre1)
plot(tre2$tree,root.edge=TRUE)
```

Index