

Package ‘phylocurve’

June 7, 2017

Version 2.0.8

Date 2017-06-06

Title Phylogenetic Comparative Methods for High-Dimensional Traits

Author Eric W. Goolsby

Maintainer Eric W. Goolsby <eric.goolsby.evolution@gmail.com>

Description Tools for studying the evolution of high-dimensional traits (morphometric, function-valued, etc.) including ancestral state reconstruction, estimating phylogenetic signal, and assessing correlated trait evolution. Visit <<http://www.phylocurve.org>> for more information.

License GPL (>= 2)

URL <http://www.phylocurve.org>

Imports Rcpp, ape, drc, dtw, geiger, geomorph (>= 3.0.0), GPfit, Matrix, mvnmle, phylolm, phytools, rgl, doParallel, foreach, parallel, grDevices, graphics, stats, utils

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-06-07 05:33:30 UTC

R topics documented:

phylocurve-package	2
compare.models	3
evo.model	5
fast.geomorph.compare.evol.rates	7
fast.geomorph.compare.multi.evol.rates	9
fast.geomorph.phylo.integration	11
fast.geomorph.physignal	12
fast.geomorph.procD.pgls	14
get.aligned.function.data	15

get.tip.coefficients	16
GP.fit	17
K.mult	18
multipic	19
nonlinear.fit	21
paint.edges	23
phylocurve	24
phylocurve.generalized	25
phylocurve.trim	27
polynomial.fit	28
prep_multipic	29
print.evo.model	31
sim.curves	31
sim.traits	32
ultraFastAnc	34

Index	36
--------------	-----------

phylocurve-package	<i>Phylogenetic Comparative Methods for Function-Valued and Other High-Dimensional Traits</i>
--------------------	---

Description

Tools for studying the evolution of high-dimensional traits (in particular, function-valued traits), including ancestral state reconstruction, estimating phylogenetic signal, and assessing correlated trait evolution.

Details

Package: phylocurve
 Type: Package
 Version: 2.0.8
 Date: 2017-06-06
 License: GPL (>= 2)

Author(s)

Eric W. Goolsby <eric.goolsby.evolution@gmail.com>

References

Goolsby, E.W. 2015. "Phylogenetic comparative methods for evaluating the evolutionary history of function-valued traits." *Systematic Biology*. In press.

Adams, D.C. 2014. A method for assessing phylogenetic least squares models for shape and other high-dimensional multivariate data. *Evolution*. 68:2675-2688.

Adams, D.C. 2014. A generalized K statistic for estimating phylogenetic signal from shape and other high-dimensional multivariate data. *Systematic Biology*. 63:685-697.

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.

Examples

```
require(phytools)

# simulate evolution of a function-valued trait (glm with logit link)
sim_data <- sim.curves()

# perform ancestral curve reconstruction
anc_recon <- phylocurve(y~x, tree = sim_data$tree, data = sim_data$data)

# get tip coefficients and aligned function data
tip_coefficients <- get.tip.coefficients(formula = y~x, tree = sim_data$tree, data = sim_data$data)
data <- get.aligned.function.data(tip_coefficients)

# estimate evolutionary rates
evo.model.fitted <- evo.model(sim_data$tree, data)
```

compare.models

Compare evolutionary hypotheses

Description

This function compares two evolutionary hypotheses via phylogenetic simulation under the null model (model1) and alternative model (model2) (Boettiger et al. 2012; Goolsby 2016). Typically the alternative model is nested within the null model, but is not a requirement of this testing procedure. By default, data are randomly simulated under model1 and model2, and model1 and model2 are fit to generate a distribution of likelihood ratios for data simulated under the null hypothesis and a distribution of likelihood ratios from data simulated under the alternative hypothesis. Comparison of the observed likelihood ratio to the null distribution provides an estimate of the P-value, whereas comparison of the alternative distribution to the critical value of the test statistic provides an estimate of statistical power.

Usage

```
compare.models(model1, model2, nsim = 1000,
plot = TRUE, estimate_power = TRUE, parallel = TRUE,
conf.int = TRUE)
```

Arguments

model1	The null model (an object of class <code>rate.mult</code>).
model2	The alternative model (an object of class <code>rate.mult</code>).
nsim	The number of iterations for phylogenetic simulation.
plot	Whether or not to plot the null and alternative distributions of likelihood ratios.
estimate_power	Whether to simulate the alternative distribution (default=TRUE).
parallel	Whether to use parallel processing to speed up computations (default=TRUE).
conf.int	Whether to estimate confidence intervals for tree transformation parameters in the alternative model (e.g., OU, EB, lambda, kappa, delta). If TRUE and the alternative evolutionary model is not 'BM', <code>estimate_power</code> is automatically set to TRUE.

Value

An object of class `compare.models`.

Author(s)

Eric W. Goolsby

References

Boettiger C., Coop G., Ralph P. 2012. Is your phylogeny informative? Measuring the power of comparative methods. *Evolution*. 66:2240-2251.

Goolsby E.W. 2015. Likelihood-Based Parameter Estimation for High-Dimensional Phylogenetic Comparative Models: Overcoming the Limitations of 'Distance-Based' Methods. In review.

See Also

[evo.model](#)

Examples

```

rand.data <- sim.traits()
X <- rowMeans(rand.data$trait_data)
null.model <- evo.model(tree = rand.data$tree,
  Y = rand.data$trait_data,method = "Pairwise ML")
alt.model <- evo.model(tree = rand.data$tree,
  Y = rand.data$trait_data,fixed.effects = X,method = "Pairwise ML")

compare.models(model1 = null.model,model2 = alt.model,
  nsim = 100,parallel = FALSE)

```

evo.model

*Fit an evolutionary model***Description**

This function fits the evolutionary model as specified (e.g., trait groups, BM/EB/OU model, fixed effects, etc.).

Usage

```
evo.model(tree, Y, fixed.effects = NA, species.groups, trait.groups,
model = "BM", diag.phylocov = FALSE, method = "Pairwise REML",
force.zero.phylocov = character(), species.id = "species",
max.combn = 10000, painted.edges, ret.level = 2, plot.LL.surface = FALSE,
par.init.iters = 50, fixed.par = numeric(), multirate = FALSE, subset = TRUE,
bounds)
```

Arguments

tree	An object of class 'phylo'
Y	An nspecies-by-ntraits matrix with row names corresponding to species names. Alternatively, a data frame with a column for species names (with name species.id, which by default is 'species').
fixed.effects	Optional. An matrix of fixed effects (predictor variables for phylogenetic regression), of the same format as Y.
species.groups	Optional. A named factor assigning species to regimes. Names correspond to species names.
trait.groups	Optional. A named factor assigning species to trait groups, for each a generalized evolutionary rate is fit (see Adams and Collyer 2015, Evolution). NOTE: if the argument is called, multirate is automatically set to TRUE.
model	Default = 'BM': The model for branch length transformations (default is 'BM'). Other options include 'OU', 'EB', 'lambda', 'kappa', and 'delta'.
diag.phylocov	Default = FALSE. Whether to assume trait independence (a diagonal evolutionary rate matrix). Not recommended unless specifically testing for trait independence or if trait independence is deliberately assumed a priori.
method	Default = 'Pairwise REML'. The method for calculating log-likelihood. Other options include 'Pairwise ML', 'Full ML', and 'Full REML'. In general, it is recommended to always use 'Pairwise REML', but when comparing fixed effects, use an 'ML' method should be used. NOTE: 'Full ML' and 'Full REML' are numerically unstable as the number of traits approaches the number of species, and is undefined if ntraits >= nspecies.
force.zero.phylocov	Optional. The names of traits to be assigned to group A, in which group A is assumed to be uncorrelated with the remaining traits in group B. This procedure is used to test for evolutionary covariance between two groups of multivariate traits, similar to phylogenetic partial least squares (Adams and Felice 2014).

species.id	Default = 'species'. The name of the column in 'Y' corresponding to species names. Only applicable if 'Y' is a data.frame.
max.combn	Default = 10000. The maximum number of pairwise combinations allowed for pairwise log-likelihood calculations. If <code>ncol(combn(ncol(Y))) > max.combn</code> , a Monte Carlo approach with <code>max.combn</code> random samples is used to approximate the composite log-likelihood.
paint.edges	Optional. An nedge-by-nspecies.groups matrix corresponding to the proportion of each tree edge to assign to an evolutionary regime. This is fit automatically if not supplied, or can be fit using the paint.edges function. Edges MUST be in postorder.
ret.level	Default = 2. The level of information to return. <code>ret.level = 1</code> only returns the log-likelihood; <code>ret.level = 2</code> returns most information but not the evolutionary rate matrix. <code>ret.level = 3</code> returns most information including the evolutionary matrix.
plot.LL.surface	Default = FALSE. Whether to plot the log-likelihood surface when estimating tree transformation parameters (i.e. OU, EB, lambda, delta, or kappa).
par.init.iters	Default = 50. The number of values to evaluate along the feasible range of tree transformations parameters (not applicable if <code>model='BM'</code>).
fixed.par	Optional. If a tree transformation is used (i.e. OU, EB, lambda, delta, or kappa), this argument can be used to fix the tree transformation parameter to a fixed value.
multirate	Default = FALSE. Whether or not to use a single value for evolutionary rates instead of an unconstrained evolutionary rate matrix. For compatibility with compare.evol.rates and compare.multi.evol.rates .
subset	Default = TRUE. Whether or not traits are subsets of a single landmark configuration (implemented for compatibility with the compare.multi.evol.rates function).
bounds	Optional argument to set bounds for evolutionary model search space (does not apply to BM models).

Details

Coming soon.

Value

An object of class `evo.model`.

Author(s)

Eric W. Goolsby

References

- Goolsby E.W. 2016. Likelihood-Based Parameter Estimation for High-Dimensional Phylogenetic Comparative Models: Overcoming the Limitations of 'Distance-Based' Methods. In review.
- Adams D.C. 2014a. A generalized K statistic for estimating phylogenetic signal from shape and other high-dimensional multivariate data. *Syst. Biol.* 63:685-697.
- Adams D.C. 2014b. A method for assessing phylogenetic least squares models for shape and other high-dimensional multivariate data. *Evolution* 68:2675-2688.
- Adams D.C. 2014c. Quantifying and comparing phylogenetic evolutionary rates for shape and other high-dimensional phenotypic data. *Syst. Biol.* 63:166-177.
- Adams D.C., Felice R. 2014. Assessing phylogenetic morphological integration and trait covariation in morphometric data using evolutionary covariance matrices. *PLOS ONE* 9(4):e94335.
- Adams D.C., Collyer M. 2015. Permutation tests for phylogenetic comparative analyses of high-dimensional shape data: What you shuffle matters. *Evolution* 69:823-829.

See Also

[compare.models](#) [K.mult](#) [compare.evol.rates](#) [compare.multi.evol.rates](#) [physignal](#) [procD.pgls](#) [phylo.pls](#)

Examples

```
rand.data <- sim.traits()
X <- rowMeans(rand.data$trait_data)
null.model <- evo.model(tree = rand.data$tree,
  Y = rand.data$trait_data,method = "Pairwise ML")
alt.model <- evo.model(tree = rand.data$tree,
  Y = rand.data$trait_data,fixed.effects = X,method = "Pairwise ML")

compare.models(model1 = null.model,model2 = alt.model,
  nsim = 100,parallel = FALSE)

K.mult(model = alt.model,nsim = 100)
```

fast.geomorph.compare.evol.rates

Fast covariance-based implementations of distance-based methods

Description

The functions `fast.geomorph.compare.evol.rates`, `fast.geomorph.compare.multi.evol.rates`, `fast.geomorph.phylo.pls`, `fast.geomorph.procD.pgls`, and `fast.geomorph.physignal` are covariance-based implementations of the geomorph functions [compare.evol.rates](#), [compare.multi.evol.rates](#), [phylo.pls](#), [procD.pgls](#), and [physignal](#) using a fast linear-time algorithm. Code is directly modified from the original geomorph code for direct comparison between distance-based and covariance-based methods.

Usage

```
fast.geomorph.compare.evol.rates(phy, A, gp,  
method = "ML", ShowPlot = TRUE, iter = 1000,  
censored = FALSE, force.diag = FALSE)
```

Arguments

phy	An object of class phylo.
A	From geomorph: A matrix (n x [p x k]) or 3D array (p x k x n) containing GPA-aligned coordinates for a set of specimens
gp	From geomorph: A factor array designating group membership
method	Maximum likelihood "ML" or restricted maximum likelihood "REML"
ShowPlot	From geomorph: A logical value indicating whether or not the plot should be returned
iter	From geomorph: Number of iterations for significance testing
censored	Whether or not to use the censored approach for estimating regime-specific evolutionary rates (O'Meara et al. 2006). The censored approach is more efficient and has identical statistical power to the default method based on phylogenetic transformation of residuals.
force.diag	Whether or not to assume trait independence (a diagonal evolutionary rate matrix), as was originally described in Adams (2014c) and implemented in geomorph up to version 2.1.5. Should normally be set to FALSE (the default) unless trait independence is deliberately assumed.

Details

See [compare.evol.rates](#)

Value

See [compare.evol.rates](#)

Author(s)

Eric W. Goolsby

References

Goolsby E.W. 2016. Likelihood-Based Parameter Estimation for High-Dimensional Phylogenetic Comparative Models: Overcoming the Limitations of 'Distance-Based' Methods. In review.

Adams, D.C. 2014. Quantifying and comparing phylogenetic evolutionary rates for shape and other high-dimensional phenotypic data. Syst. Biol. 63:166-177.

See Also

[compare.evol.rates](#)

Examples

```

### NOTE: this example is identical
### to the example code for the
### analogous geomorph function
### for direct comparisons with
### 'fast.geomorph' phylocurve functions

require(geomorph)
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land)    #GPA-alignment

gp.end<-factor(c(0,0,1,0,0,1,1,0,0)) #endangered species vs. rest
names(gp.end)<-plethspecies$phy$tip

#Calculate rates of shape
compare.evol.rates(phy=plethspecies$phy,Y.gpa$coords,gp=gp.end,iter=49)
fast.geomorph.compare.evol.rates(plethspecies$phy,Y.gpa$coords,gp=gp.end)

#Calculate rates of size
compare.evol.rates(phy=plethspecies$phy,Y.gpa$Csize,gp=gp.end,iter=49)
fast.geomorph.compare.evol.rates(plethspecies$phy,Y.gpa$Csize,gp=gp.end,iter=49)

```

```
fast.geomorph.compare.multi.evol.rates
```

Fast covariance-based implementations of distance-based methods

Description

The functions `fast.geomorph.compare.evol.rates`, `fast.geomorph.compare.multi.evol.rates`, `fast.geomorph.phylo.pls`, `fast.geomorph.procD.pgls`, and `fast.geomorph.physignal` are covariance-based implementations of the geomorph functions [compare.evol.rates](#), [compare.multi.evol.rates](#), [phylo.pls](#), [procD.pgls](#), and [physignal](#) using a fast linear-time algorithm. Code is directly modified from the original geomorph code for direct comparison between distance-based and covariance-based methods.

Usage

```
fast.geomorph.compare.multi.evol.rates(A, gp,
  phy, Subset = TRUE, method = "ML", ShowPlot = TRUE, iter = 1000)
```

Arguments

A	From geomorph: A matrix (n x [p x k]) or 3D array (p x k x n) containing GPA-aligned coordinates for a set of specimens
gp	From geomorph: A factor array designating group membership
phy	From geomorph: A phylogenetic tree of class phylo
Subset	From geomorph: A logical value indicating whether or not the traits are subsets from a single landmark configuration (default is TRUE)

method	Maximum likelihood "ML" or restricted maximum likelihood "REML"
ShowPlot	From geomorph: A logical value indicating whether or not the plot should be returned
iter	From geomorph: Number of iterations for significance testing

Details

See [compare.multi.evol.rates](#)

Value

See [compare.multi.evol.rates](#)

Author(s)

Eric W. Goolsby

References

Goolsby E.W. 2016. Likelihood-Based Parameter Estimation for High-Dimensional Phylogenetic Comparative Models: Overcoming the Limitations of 'Distance-Based' Methods. In review.

Adams, D.C. 2014. Quantifying and comparing phylogenetic evolutionary rates for shape and other high-dimensional phenotypic data. *Syst. Biol.* 63:166-177.

Denton, J.S.S., and D.C. Adams. 2015. A new phylogenetic test for comparing multiple high-dimensional evolutionary rates suggests interplay of evolutionary rates and modularity in lanternfishes (Myctophiformes; Myctophidae). *Evolution*. 69: doi:10.1111/evo.12743

See Also

[compare.multi.evol.rates](#)

Examples

```
### NOTE: this example is identical
### to the example code for the
### analogous geomorph function
### for direct comparisons with
### 'fast.geomorph' phylocurve functions
require(geomorph)
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land) #GPA-alignment
land.gp<-c("A","A","A","A","A","B","B","B","B","B","B") #mandible and cranium subsets

compare.multi.evol.rates(Y.gpa$coords,land.gp,phy=plethspecies$phy,iter=99)
fast.geomorph.compare.multi.evol.rates(Y.gpa$coords,land.gp,plethspecies$phy)
```

`fast.geomorph.phylo.integration`*Fast covariance-based implementations of distance-based methods*

Description

The functions `fast.geomorph.compare.evol.rates`, `fast.geomorph.compare.multi.evol.rates`, `fast.geomorph.phylo.integration`, `fast.geomorph.procD.pgls`, and `fast.geomorph.physignal` are covariance-based implementations of the geomorph functions [compare.evol.rates](#), [compare.multi.evol.rates](#), [phylo.integration](#), [procD.pgls](#), and [physignal](#) using a fast linear-time algorithm. Code is directly modified from the original geomorph code for direct comparison between distance-based and covariance-based methods.

Usage

```
fast.geomorph.phylo.integration(A1, A2, phy, iter = 1000,  
label = NULL, verbose = FALSE, ShowPlot = TRUE)
```

Arguments

A1	From geomorph: A 2D array (n x [p1 x k1]) or 3D array (p1 x k1 x n) containing landmark coordinates for the first block
A2	From geomorph: A 2D array (n x [p2 x k2]) or 3D array (p2 x k2 x n) containing landmark coordinates for the second block
phy	From geomorph: A phylogenetic tree of class phylo
iter	From geomorph: Number of iterations for significance testing
label	From geomorph: An optional vector indicating labels for each specimen that are to be displayed
verbose	From geomorph: A logical value indicating whether the output is basic or verbose
ShowPlot	From geomorph: A logical value indicating whether or not the plot should be returned

Details

See `phylo.integration`

Value

See `phylo.integration`

Author(s)

Eric W. Goolsby

References

Goolsby E.W. 2016. Likelihood-Based Parameter Estimation for High-Dimensional Phylogenetic Comparative Models: Overcoming the Limitations of 'Distance-Based' Methods. In review.

Adams, D.C. and R. Felice. 2014. Assessing phylogenetic morphological integration and trait covariation in morphometric data using evolutionary covariance matrices. PLOS ONE. 9(4):e94335.

See Also

[phylo.integration](#)

Examples

```
### NOTE: this example is identical
### to the example code for the
### analogous geomorph function
### for direct comparisons with
### 'fast.geomorph' phylocurve functions
require(geomorph)
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land)    #GPA-alignment

# NOTE: phylo.integration is phylo.integration in newer versions of geomorph
# phylo.integration(Y.gpa$coords[1:5,,],Y.gpa$coords[6:11,,],plethspecies$phy,iter=99)
# phylo.integration(Y.gpa$coords[1:5,,],Y.gpa$coords[6:11,,],plethspecies$phy,iter=99)
fast.geomorph.phylo.integration(Y.gpa$coords[1:5,,],Y.gpa$coords[6:11,,],plethspecies$phy)
```

fast.geomorph.physignal

Fast covariance-based implementations of distance-based methods

Description

The functions `fast.geomorph.compare.evol.rates`, `fast.geomorph.compare.multi.evol.rates`, `fast.geomorph.phylo.pls`, `fast.geomorph.procD.pgls`, and `fast.geomorph.physignal` are covariance-based implementations of the geomorph functions [compare.evol.rates](#), [compare.multi.evol.rates](#), [phylo.pls](#), [procD.pgls](#), and [physignal](#) using a fast linear-time algorithm. Code is directly modified from the original geomorph code for direct comparison between distance-based and covariance-based methods.

Usage

```
fast.geomorph.physignal(phy, A, iter = 1000,
  ShowPlot = TRUE, method = c("Kmult", "SSC"))
```

Arguments

phy	From geomorph: A phylogenetic tree of class phylo
A	From geomorph: A matrix (n x [p x k]) or 3D array (p x k x n) containing GPA-aligned coordinates for a set of specimens
iter	From geomorph: Number of iterations for significance testing
ShowPlot	From geomorph: A logical value indicating whether or not the plot should be returned
method	From geomorph: Method for estimating phylogenetic signal (Kmult or SSC)

Details

See [physignal](#)

Value

See [physignal](#)

Author(s)

Eric W. Goolsby

References

- Goolsby E.W. 2016. Likelihood-Based Parameter Estimation for High-Dimensional Phylogenetic Comparative Models: Overcoming the Limitations of 'Distance-Based' Methods. In review.
- Blomberg SP, Garland T, Ives AR. 2003. Testing for phylogenetic signal in comparative data: behavioral traits are more labile. *Evolution*, 57:717-745.
- Klingenberg, C. P., and N. A. Gidaszewski. 2010. Testing and quantifying phylogenetic signals and homoplasy in morphometric data. *Syst. Biol.* 59:245-261.
- Adams, D.C. 2014. A generalized K statistic for estimating phylogenetic signal from shape and other high-dimensional multivariate data. *Systematic Biology*. 63:685-697.

See Also

[physignal](#)

Examples

```
### NOTE: this example is identical
### to the example code for the
### analogous geomorph function
### for direct comparisons with
### 'fast.geomorph' phylocurve functions
require(geomorph)
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land) #GPA-alignment

#Test for phylogenetic signal in shape
```

```
physignal(phy=plethspecies$phy,Y.gpa$coords,iter=99)
fast.geomorph.physignal(plethspecies$phy,Y.gpa$coords,method="Kmult")

#Test for phylogenetic signal in size
physignal(phy=plethspecies$phy,Y.gpa$Csize,iter=99)
fast.geomorph.physignal(plethspecies$phy,Y.gpa$Csize,method="Kmult")
```

```
fast.geomorph.procD.pgls
```

Fast covariance-based implementations of distance-based methods

Description

The functions `fast.geomorph.compare.evol.rates`, `fast.geomorph.compare.multi.evol.rates`, `fast.geomorph.phylo.pls`, `fast.geomorph.procD.pgls`, and `fast.geomorph.physignal` are covariance-based implementations of the geomorph functions [compare.evol.rates](#), [compare.multi.evol.rates](#), [phylo.pls](#), [procD.pgls](#), and [physignal](#) using a fast linear-time algorithm. Code is directly modified from the original geomorph code for direct comparison between distance-based and covariance-based methods.

Usage

```
fast.geomorph.procD.pgls(f1, phy, iter = 1000,
int.first = FALSE, verbose = FALSE)
```

Arguments

<code>f1</code>	From geomorph: A formula for the linear model (e.g., $y \sim x1 + x2$)
<code>phy</code>	From geomorph: A phylogenetic tree of class <code>phylo</code>
<code>iter</code>	From geomorph: Number of iterations for significance testing
<code>int.first</code>	From geomorph: A logical value to indicate if interactions of first main effects should precede subsequent main effects
<code>verbose</code>	From geomorph: A logical value specifying whether additional output should be displayed

Details

See [procD.pgls](#)

Value

See [procD.pgls](#)

Author(s)

Eric W. Goolsby

References

- Goolsby E.W. 2016. Likelihood-Based Parameter Estimation for High-Dimensional Phylogenetic Comparative Models: Overcoming the Limitations of 'Distance-Based' Methods. In review.
- Adams, D.C. 2014. A method for assessing phylogenetic least squares models for shape and other high-dimensional multivariate data. *Evolution*. 68:2675-2688.

See Also

[procD.pgls](#)

Examples

```
### NOTE: this example is identical
### to the example code for the
### analogous geomorph function
### for direct comparisons with
### 'fast.geomorph' phylocurve functions
require(geomorph)

### Example of D-PGLS for high-dimensional data
data(plethspecies)
Y.gpa<-gpagen(plethspecies$land) #GPA-alignment
gdf <- geomorph.data.frame(Y.gpa, phy = plethspecies$phy)
procD.pgls(coords ~ Csize,data = gdf,plethspecies$phy,iter=1000)
fast.geomorph.procD.pgls(Y.gpa$coords ~ Y.gpa$Csize,plethspecies$phy)
```

```
get.aligned.function.data
```

Estimate aligned data for function-valued traits

Description

Estimates function data for tip species curves from tip coefficients.

Usage

```
get.aligned.function.data(tip.coefficients, ylength, ymin = 0.01, ymax = 0.99)
```

Arguments

tip.coefficients	Matrix of estimated regression coefficients of tip curves. Row names should correspond to species names. The first column should contain the logit glm intercept; the second column contains the logit glm slope.
ylength	How many landmarks (points on the curve) to evaluate.
ymin	Because 0 and 1 are undefined for inverse logit functions, the minimum and maximum values are defaulted to .01 and .99. Can be adjusted as needed.
ymax	Because 0 and 1 are undefined for inverse logit functions, the minimum and maximum values are defaulted to .01 and .99. Can be adjusted as needed.

Value

A data frame of aligned X-coordinates for function-valued traits for a given constant Y.

Author(s)

Eric W. Goolsby

References

Goolsby, E.W. 2015. "Phylogenetic comparative methods for evaluating the evolutionary history of function-valued traits." *Systematic Biology*. In press.

Examples

```
# simulate evolution of a function-valued trait (glm with logit link)
sim_data <- sim.curves()

# get tip coefficients and aligned function data
tip.coefficients <- get.tip.coefficients(formula = y~x, tree = sim_data$tree, data = sim_data$data)
data <- get.aligned.function.data(tip.coefficients)

# estimate evolutionary rates
evo.model.fitted <- evo.model(sim_data$tree, data)
```

get.tip.coefficients *Estimate regression coefficients for tip species curves*

Description

Estimates regression coefficients for tip species curves from raw data. May be useful if wanting to perform methods without first performing ancestral curve reconstruction via the `phylocurve()` function.

Usage

```
get.tip.coefficients(formula, tree, data, ymin = 0.01, ymax = 0.99,
  ylength = 30, species.identifier = "species", verbose = FALSE)
```

Arguments

formula	Formula for function-valued trait (currently only supports models of the form Y~X)
tree	A phylogenetic tree of class "phylo"
data	A data frame with data for tip curve estimation, where each row contains a single data point. A column named "species" has the species names corresponding to each data point, a predictor (X) variable and the response (Y) variable, which must be scaled between 0 and 1.

ymin	Because 0 and 1 are undefined for inverse logit functions, the minimum and maximum values are defaulted to .01 and .99. Can be adjusted as needed.
ymax	Because 0 and 1 are undefined for inverse logit functions, the minimum and maximum values are defaulted to .01 and .99. Can be adjusted as needed.
ylength	How many landmarks (points on the curve) to evaluate.
species.identifier	Default is "species". Can be changed if the column in data has a different species identifier name.
verbose	either to print progress during tip curve coefficient estimation.

Value

Estimated regression coefficients of tip curves.

Examples

```
# simulate evolution of a function-valued trait (glm with logit link)
sim_data <- sim.curves()

# get tip coefficients and aligned function data
tip.coefficients <- get.tip.coefficients(formula = y~x, tree = sim_data$tree, data = sim_data$data)
data <- get.aligned.function.data(tip.coefficients)

# estimate evolutionary rates
evo.model.fitted <- evo.model(sim_data$tree, data)
```

GP.fit

Fit Gaussian process curves to species data

Description

Transforms raw data into regression curves using the `GP_fit` function in the `GPfit` package. The response data can be constrained between minima and maxima (for example, the default sets any negative predicted y value to 0).

Usage

```
GP.fit(data, x_variable, y_variable, min_x = -Inf, max_x = Inf,
       min_y = 0, max_y = Inf, eval_length = 30, ...)
```

Arguments

data	Data frame with function-valued species data along with a column named "species"
x_variable	The name of the column in data corresponding to the predictor (x) variable
y_variable	The name of the column in data corresponding to the response (y) variable
min_x	Lower range to remove from raw x data

max_x	Upper range to remove from raw x data
min_y	Lower range to remove from predicted response (for example, for response data in which only positive values are allowed, the min_y can be set to 0 (the default)).
max_y	Upper range to remove from predicted response.
eval_length	How many points to evaluate along the range of x.
...	Optional further arguments to pass to the GP_fit function.

Value

X	X (formatted for phylocurve.generalized)
Y	Y (formatted for phylocurve.generalized)

Examples

```
x_length <- 50
nspecies <- 25
sim_data <- sim.curves(nspecies = nspecies,x_length = x_length)

# Fit nonlinear curves to raw data
# NOT RUN (takes several minutes)
#curve_data <- GP.fit(data = sim_data$data,x_variable = "x",y_variable = "y",min_y = 0)

# Align curve data using dynamic time warping
# P <- phylocurve.generalized(tree = sim_data$tree,X = curve_data$X,Y = curve_data$Y)
```

K.mult

*Test phylogenetic signal (Kmult) using phylogenetic simulation***Description**

An implementation of Kmult (Blomberg et al. 2003; Adams 2014) in which Kmult is compared to the null distribution generated by simulation on a star phylogeny (rather than phylogenetic permutation). Additionally, the expectation of the ratio of raw to phylogenetic mean squared error (the denominator of the formula for K) is estimated via phylogenetic simulation under Brownian motion.

Usage

```
K.mult(model, nsim = 1000, plot = TRUE)
```

Arguments

model	An object of class <code>evo.model</code> .
nsim	The number of iterations for phylogenetic simulation.
plot	Whether or not to plot the null and alternative distribution of Kmult.

Value

An object of class `compare.model`.

Author(s)

Eric W. Goolsby

References

Golsby E.W. 2015. Likelihood-based parameter estimation for high-dimensional phylogenetic comparative models: overcoming the limitations of 'distance-based' methods. In review.

Adams D.C. 2014. A generalized K statistic for estimating phylogenetic signal from shape and other high-dimensional multivariate data. *Syst. Biol.* 63:685-697.

See Also

[evo.model](#) [compare.models](#) [physignal](#)

Examples

```
rand.data <- sim.traits()
null.model <- evo.model(tree = rand.data$tree,
  Y = rand.data$trait_data,method = "Pairwise ML")

K.mult(model = null.model,nsim = 100)
```

`multipic`

Efficient PIC algorithm for multiple traits

Description

Following the function `prep_multipic`, this function efficiently performs the PIC algorithm for multiple traits simultaneously, allowing for rapid repeated calls using different datasets (e.g., for bootstrap simulations). NOTE: this function is intended for use via the `do.call` function (see example).

Usage

```
multipic(ntip, nnode, edge1, edge2, edge_len, phe, contr,
  var_contr, scaled, pic_len, pic_recon)
```

Arguments

<code>ntip</code>	This quantity is computed by the function <code>prep_multipic</code>
<code>nnode</code>	This quantity is computed by the function <code>prep_multipic</code>
<code>edge1</code>	This quantity is computed by the function <code>prep_multipic</code>
<code>edge2</code>	This quantity is computed by the function <code>prep_multipic</code>

edge_len	This quantity is computed by the function prep_multipic
phe	This quantity is computed by the function prep_multipic
contr	This quantity is computed by the function prep_multipic
var_contr	This quantity is computed by the function prep_multipic
scaled	This quantity is computed by the function prep_multipic
pic_len	This quantity is computed by the function prep_multipic
pic_recon	This quantity is computed by the function prep_multipic

Value

contrasts	A matrix of PICs, one column for each trait (rows correspond to node numbers on a postorder tree)
sum_invV	Sum of the inverse of the phylogenetic covariance matrix: <code>sum(solve(vcv(tree)))</code>
log_detV	The natural log of the determinant of the phylogenetic covariance matrix
root	Vector of estimated root values for each trait
pic_len	Vector of PIC-rescaled branch lengths
pic_recon	Matrix of PIC-reconstructed ancestral values. NOTE: this is not equivalent to the ML ancestral estimates except for the root value!

References

- Felsenstein, J. (1985) Phylogenies and the comparative method. *American Naturalist*, 125, 1-15.
- Paradis, E., Claude, J. and Strimmer, K. (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20, 289-290.
- Paradis, E. (2012) *Analysis of Phylogenetics and Evolution with R (Second Edition)*. New York: Springer.

See Also

[pic](#)

Examples

```
require(phylocurve)
require(ape)

nspecies <- 1000
ntraits <- 500
tree <- rtree(n = nspecies) # Simulate a random 1000-species phylogeny
Y <- matrix(rnorm(nspecies*ntraits),ncol=ntraits) # Generate random data
rownames(Y) <- tree$tip.label

# Call prep_multipic
prep.Y <- prep_multipic(x = Y,phy = tree)

# Calculate phylogenetically independent contrasts using pic (ape package)
```

```

Y.pics.ape <- apply(X = Y,MARGIN = 2,FUN = pic,phy = tree)

# Calculate PICs using multipic
# returns a list with contrasts, sum_invV, log_detV, root
# 'contrast' is a matrix of PICs
# 'sum_invV' is equal to sum(solve(vcv(tree)))
# 'log_detV' is equal to log(det(vcv(tree)))
# 'root' is the maximum likelihood phenotypic value at the root
Y.pics.phylocurve <- do.call(multipic,prep.Y)

# Verify that results are identical
range(Y.pics.ape - Y.pics.phylocurve$contrasts)

# Generate 50 random datasets (NOT RUN)
#niter <- 50
#randomY <- vector(mode = "list",length = niter)
#for(i in 1:niter)
#{
# randomY[[i]] <- matrix(rnorm(nspecies*ntraits),ncol=ntraits)
# rownames(randomY[[i]]) <- tree$tip.label
#}

#####
# Compare time to calculate PICs on 50 random datasets using pic vs multipic
#
##### pic function (NOT RUN)
#system.time(for(i in 1:niter) apply(X = randomY[[i]],MARGIN = 2,FUN = pic,phy = tree))
##### user system elapsed
##### 18.35 0.23 18.61

##### multipic function (NOT RUN)
#system.time(for(i in 1:niter)
#{
# prep.Y$phe[1:nspecies,] <- randomY[[i]] # update prep.Y$phe with new data
# do.call(multipic,prep.Y)
#})
##### user system elapsed
##### 1.38 0.14 1.52
#
#####

```

nonlinear.fit

Fit nonlinear curves to species data

Description

Transforms raw data into regression curves using the `drm` function in the `drc` package. The response data can be constrained between minima and maxima (for example, the default sets any negative predicted y value to 0).

Usage

```
nonlinear.fit(data, x_variable, y_variable, fct = LL2.3(),
  min_x = -Inf, max_x = Inf, min_y = 0, max_y = Inf, eval_length = 30, ...)
```

Arguments

<code>data</code>	Data frame with function-valued species data along with a column named "species"
<code>x_variable</code>	The name of the column in data corresponding to the predictor (x) variable
<code>y_variable</code>	The name of the column in data corresponding to the response (y) variable
<code>fct</code>	Function type to pass to <code>drm</code> . Use getMeanFunctions for a full list.
<code>min_x</code>	Lower range to remove from raw x data
<code>max_x</code>	Upper range to remove from raw x data
<code>min_y</code>	Lower range to remove from predicted response (for example, for response data in which only positive values are allowed, the <code>min_y</code> can be set to 0 (the default)).
<code>max_y</code>	Upper range to remove from predicted response.
<code>eval_length</code>	How many points to evaluate along the range of x.
<code>...</code>	Optional further arguments to pass to the drm function.

Value

X	X (formatted for <code>phylocurve.generalized</code>)
Y	Y (formatted for <code>phylocurve.generalized</code>)

Examples

```
x_length <- 50
nspecies <- 25
sim_data <- sim.curves(nspecies = nspecies, x_length = x_length)

# Fit nonlinear curves to raw data
curve_data <- nonlinear.fit(data = sim_data$data, x_variable = "x", y_variable = "y", min_y = 0)

# Align curve data using dynamic time warping
# NOT RUN (takes several minutes)
# P <- phylocurve.generalized(tree = sim_data$tree, X = curve_data$X, Y = curve_data$Y)
```

paint.edges	<i>Paint tree edges based on species regimes</i>
-------------	--

Description

This function estimates the proportion of edges to be assigned to regimes based on ancestral regime reconstruction (Yang et al. 1995; Pupko et al. 2000; Paradis et al. 2004; Revell 2012).

Usage

```
paint.edges(tree, species.groups, average.nodes = TRUE, root.edge = TRUE)
```

Arguments

tree	An object of class phylo.
species.groups	A named factor assigning species to regimes.
average.nodes	Whether or not to average reconstructed probabilities of regime states from parent and daughter nodes for tree edges.
root.edge	Whether or not to return a root edge.

Value

A matrix of dimension nedge-by-nregimes. NOTE: edges are in postorder.

Author(s)

Eric W. Goolsby

References

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

Pupko T., Pe'er I., Shamir R., Graur D. 2000. A fast algorithm for joint reconstruction of ancestral amino acid sequences. *Molecular Biology and Evolution*. 17:890-896.

Yang Z., Kumar S., Nei, M. 1995. A new method of inference of ancestral nucleotide and amino acid sequences. *Genetics*. 141:1641-1650.

Examples

```
require(ape)
tree <- rtree(n=10)
groups <- setNames(factor(c(rep("groupA", 5), rep("groupB", 5))), tree$tip.label)
painted.edges <- paint.edges(tree = tree, species.groups = groups)
```

phylocurve	<i>Ancestral curve reconstruction for logistic regression (glm with logit link)</i>
------------	---

Description

This function performs ancestral reconstruction of function-valued traits (assuming Brownian motion).

Usage

```
phylocurve(formula, tree, data, ymin = 0.01, ymax = 0.99, ylength = 30,
tip.coefficients, species.identifier = "species", verbose = FALSE)
```

Arguments

formula	Formula for function-valued trait (currently only supports models of the form $Y \sim X$)
tree	An phylogenetic tree of class "phylo"
data	A data frame with data for tip curve estimation, where each row contains a single data point. A column named "species" has the species names corresponding to each data point, a predictor (X) variable and the response (Y) variable, which must be scaled between 0 and 1.
ymin	Because 0 and 1 are undefined for inverse logit functions, the minimum and maximum values are defaulted to .01 and .99. Can be adjusted as needed.
ymax	Because 0 and 1 are undefined for inverse logit functions, the minimum and maximum values are defaulted to .01 and .99. Can be adjusted as needed.
ylength	How many landmarks (points on the curve) to evaluate.
tip.coefficients	A matrix of regression coefficients can be provided instead of raw data. Row names should be species names, the first column should be the glm logit intercept parameter, and the second column should be the glm logit slope parameter.
species.identifier	Default is "species". Can be changed if the column in data has a different species identifier name.
verbose	Whether to print progress during ancestral curve reconstruction.

Details

This function uses a PGLS-based method described in Goolsby (2015) to perform ancestral curve reconstruction. This function uses a fast tree transversal method via the phylolm package (Ho and Ane, 2014)

Value

node_coefficients	Estimated regression coefficients of internal nodes
fitted_x	Reconstructed x-values for each internal node. Curve coordinates for internal node <i>i</i> are (fitted_x[<i>i</i>],y_vals)
lower_CI_x	Lower 95 percent confidence interval for fitted x-values for each internal node
upper_CI_x	Upper 95 percent confidence interval for fitted x-values for each internal node
y_vals	Vector of y-values used as inputs for inverse function evaluation. Curve coordinates for internal node <i>i</i> are (fitted_x[<i>i</i>],y_vals)
tip_coefficients	Estimated regression coefficients of tip curves.

Author(s)

Eric W. Goolsby

References

- Goolsby, E.W. 2015. "Phylogenetic comparative methods for evaluating the evolutionary history of function-valued traits." *Systematic Biology*. In press.
- 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.

Examples

```
# simulate evolution of a function-valued trait (glm with logit link)
sim_data <- sim.curves()

# perform ancestral curve reconstruction
anc_recon <- phylocurve(y~x, tree = sim_data$tree, data = sim_data$data)

# get tip coefficients and aligned function data
tip_coefficients <- get.tip.coefficients(formula = y~x, tree = sim_data$tree, data = sim_data$data)
data <- get.aligned.function.data(tip_coefficients)

# estimate evolutionary rates
evo.model.fitted <- evo.model(sim_data$tree, data)
```

phylocurve.generalized

Align curves of any shape

Description

Aligns curves by x,y coordinates so that species curves can be analyzed as a single high-dimensional trait.

Usage

```
phylocurve.generalized(tree, X, Y)
```

Arguments

tree	An object of class phylo
X	A numeric vector of length M, used for each species
Y	An N x M matrix where each row corresponds to a species and each column corresponds to index M _i of X

Value

aligned_data	x and y coordinates arranged in a single row vector for each species
aligned_coordinates	x and y coordinates for each species (one data point per row)
aligned_X	Aligned X coordinates in N x M matrix
aligned_Y	Aligned Y coordinates in N x M matrix
nr	Number of aligned landmarks
anc_X	Vector of root X values (assuming Brownian motion)
anc_Y	Vector of root Y values (assuming Brownian motion)
tree	Phylogenetic tree supplied into the function

Author(s)

Eric W. Goolsby

References

Goolsby, E.W. 2015. "Phylogenetic comparative methods for evaluating the evolutionary history of function-valued traits." *Systematic Biology*. In press.

Examples

```
# NOT RUN (takes several minutes)
# x_length <- 50
# nspecies <- 25
#sim_data <- sim.curves(nspecies = nspecies,x_length = x_length)
#
# Perform ancestral curve reconstruction without any assumptions about function shape
# P <- phylocurve.generalized(tree = sim_data$tree,X = sim_data$data$x[1:x_length],
# Y = t(matrix(sim_data$data$y,nrow = x_length,
# dimnames = list(1:x_length,sim_data$tree$tip.label))))
#
# Data might need trimming for values where there is no variation
# plot(P$anc_X,P$anc_Y)
#
# P.trimmed <- phylocurve.trim(phylocurve.generalized = P,min_X = 4,max_X = 8)
```

phylocurve.trim *Trim aligned curves*

Description

Trims curves to specified range of X and Y coordinates

Usage

```
phylocurve.trim(phylocurve.generalized, min_Y = -Inf,
  max_Y = Inf, min_X = -Inf, max_X = Inf)
```

Arguments

phylocurve.generalized	The value returned by the phylocurve.generalized function
min_Y	The minimum Y value for trimming curves
max_Y	The maximum Y value for trimming curves
min_X	The minimum X value for trimming curves
max_X	The maximum X value for trimming curves

Value

aligned_data	x and y coordinates arranged in a single row vector for each species
aligned_coordinates	x and y coordinates for each species (one data point per row)
aligned_X	Aligned X coordinates in N x M matrix
aligned_Y	Aligned Y coordinates in N x M matrix
nr	Number of aligned landmarks
anc_X	Vector of root X values (assuming Brownian motion)
anc_Y	Vector of root Y values (assuming Brownian motion)
tree	Phylogenetic tree supplied into the function

Author(s)

Eric W. Goolsby

References

Goolsby, E.W. 2015. "Phylogenetic comparative methods for evaluating the evolutionary history of function-valued traits." *Systematic Biology*. In press.

Examples

```
# NOT RUN (takes several minutes)
# x_length <- 50
# nspecies <- 25
#sim_data <- sim.curves(nspecies = nspecies,x_length = x_length)
#
# Perform ancestral curve reconstruction without any assumptions about function shape
# P <- phylocurve.generalized(tree = sim_data$tree,
# X = sim_data$data$x[1:x_length],Y = t(matrix(sim_data$data$y,
# nrow = x_length,dimnames = list(1:x_length,sim_data$tree$tip.label))))
#
# Data might need trimming for values where there is no variation
# plot(P$anc_X,P$anc_Y)
#
# P.trimmed <- phylocurve.trim(phylocurve.generalized = P,min_X = 4,max_X = 8)
```

polynomial.fit

Fit polynomial curves to species data using stepwise regression

Description

Transforms raw data into regression curves using stepwise (AIC or BIC) polynomial regression. The maximum number of parameters (nterms), response data can be constrained between minima and maxima (for example, the default sets any negative predicted y value to 0).

Usage

```
polynomial.fit(data, x_variable, y_variable, method = "BIC",
  nterms = 2, min_x = -Inf, max_x = Inf, min_y = 0, max_y = Inf, eval_length = 30)
```

Arguments

data	Data frame with function-valued species data along with a column named "species"
x_variable	The name of the column in data corresponding to the predictor (x) variable
y_variable	The name of the column in data corresponding to the response (y) variable
method	Metric for stepwise polynomial regression. Either "AIC" or "BIC" (the default).
nterms	Maximum number of terms to allow in polynomial regression.
min_x	Lower range to remove from raw x data
max_x	Upper range to remove from raw x data
min_y	Lower range to remove from predicted response (for example, for response data in which only positive values are allowed, the min_y can be set to 0 (the default)).
max_y	Upper range to remove from predicted response.
eval_length	How many points to evaluate along the range of x.

Value

X X (formatted for phylocurve.generalized)
 Y Y (formatted for phylocurve.generalized)

Examples

```
x_length <- 50
nspecies <- 25
sim_data <- sim.curves(nspecies = nspecies,x_length = x_length)

# Fit polynomial curves to raw data
poly_data <- polynomial.fit(data = sim_data$data,x_variable = "x",y_variable = "y",min_y = 0)

# Align curve data using dynamic time warping
# NOT RUN (takes several minutes)
# P <- phylocurve.generalized(tree = sim_data$tree,X = poly_data$X,Y = poly_data$Y)
```

prep_multipic	<i>Prep multipic</i>
---------------	----------------------

Description

Prepares data and phylogeny for fast repeated phylogenetically independent contrast calculations in the function `multipic`.

Usage

```
prep_multipic(x, phy, scaled = TRUE, var.contrasts = FALSE,
  rescaled.tree = FALSE, pic_recon = FALSE)
```

Arguments

x	A matrix with rownames corresponding to species in the phylogeny (one observation per species and no missing data), and columns corresponding to individual traits.
phy	An object of class "phylo"
scaled	logical, indicates whether the contrasts should be scaled with their expected variances (default to TRUE).
var.contrasts	logical, indicates whether the expected variances of the contrasts should be returned (default to FALSE).
rescaled.tree	logical, if TRUE the rescaled tree is returned together with the main results.
pic_recon	logical, whether to return PIC ancestral reconstruction values (NOTE: this is not equivalent to the ML ancestral estimates except for the root value!)

Value

A list consisting of quantities used by the function `multipic`

References

- Felsenstein, J. (1985) Phylogenies and the comparative method. *American Naturalist*, 125, 1-15.
- Paradis, E., Claude, J. and Strimmer, K. (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, 20, 289-290.
- Paradis, E. (2012) *Analysis of Phylogenetics and Evolution with R* (Second Edition). New York: Springer.

See Also

[pic](#)

Examples

```
require(phylocurve)
require(ape)

nspecies <- 1000
ntraits <- 500
tree <- rtree(n = nspecies) # Simulate a random 1000-species phylogeny
Y <- matrix(rnorm(nspecies*ntraits),ncol=ntraits) # Generate random data
rownames(Y) <- tree$tip.label

# Call prep_multipic
prep.Y <- prep_multipic(x = Y,phy = tree)

# Calculate phylogenetically independent contrasts using pic (ape package)
Y.pics.ape <- apply(X = Y,MARGIN = 2,FUN = pic,phy = tree)

# Calculate PICs using multipic
# returns a list with contrasts, sum_invV, log_detV, root
# 'contrast' is a matrix of PICs
# 'sum_invV' is equal to sum(solve(vcv(tree)))
# 'log_detV' is equal to log(det(vcv(tree)))
# 'root' is the maximum likelihood phenotypic value at the root
Y.pics.phylocurve <- do.call(multipic,prep.Y)

# Verify that results are identical
range(Y.pics.ape - Y.pics.phylocurve$contrasts)

# Generate 50 random datasets (NOT RUN)
#niter <- 50
#randomY <- vector(mode = "list",length = niter)
#for(i in 1:niter)
#{
# randomY[[i]] <- matrix(rnorm(nspecies*ntraits),ncol=ntraits)
# rownames(randomY[[i]]) <- tree$tip.label
#}

#####
# Compare time to calculate PICs on 50 random datasets using pic vs multipic
```

```

#
##### pic function (NOT RUN)
#system.time(for(i in 1:niter) apply(X = randomY[[i]],MARGIN = 2,FUN = pic,phy = tree))
##### user system elapsed
##### 18.35 0.23 18.61

##### multipic function (NOT RUN)
#system.time(for(i in 1:niter)
#{
# prep.Y$phe[1:nspecies,] <- randomY[[i]] # update prep.Y$phe with new data
# do.call(multipic,prep.Y)
#})
##### user system elapsed
##### 1.38 0.14 1.52
#
#####

```

print.evo.model	<i>Print evo.model</i>
-----------------	------------------------

Description

Generic S3 method for evo.model

Usage

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

Arguments

x	An object of class evo.model
...	Further arguments passed to or from other methods.

sim.curves	<i>Simulate function-valued curve evolution</i>
------------	---

Description

This function simulates evolution of a function-valued trait (glm with logit link) under Brownian motion.

Usage

```
sim.curves(nspecies = 30, x_length = 20, starttree = FALSE, lambda = 1, seed)
```

Arguments

nspecies	Number of species
x_length	Number of points on the curve to simulate
starttree	Simulate using a star tree
lambda	Pagel's lambda tree transformation parameter (lambda = 0 for star tree; lambda = 1 for original tree)
seed	Optional seed to set for reproducibility

Value

data	Data frame with simulated curves formatted appropriately for phylocurve()
tree	Phylogenetic tree of class "phylo"
true_coefs	True tip and ancestral coefficients for glm logit link

References

Pagel, M. 1999. "Inferring the historical patterns of biological evolution". Nature 401:877-884.

Examples

```
# simulate evolution of a function-valued trait (glm with logit link)
sim_data <- sim.curves()

# perform ancestral curve reconstruction
anc_recon <- phylocurve(y~x, tree = sim_data$tree, data = sim_data$data)

# get tip coefficients and aligned function data
tip.coefficients <- get.tip.coefficients(formula = y~x, tree = sim_data$tree, data = sim_data$data)
data <- get.aligned.function.data(tip.coefficients)

# estimate evolutionary rates
evo.model.fitted <- evo.model(sim_data$tree, data)
```

sim.traits

Simulate multivariate trait evolution

Description

This function wraps the [sim.char](#) in [geiger](#) function to simulate evolutionary models under a given evolutionary rate matrix, with the option of simulating multiple within-species observations and missing data.

Usage

```
sim.traits(ntaxa = 15, ntraits = 4, nreps = 1,
           nmissing = 0, tree, v, anc, intraspecific,
           model = "BM", parameters, nsim = 1, return.type = "matrix")
```


Arguments

ntaxa	Number of species to randomly generate on a pure-birth phylogeny (used if tree is not provided).
ntraits	Number of traits to randomly simulate according to a random evolutionary rate matrix (used if v is not provided).
nreps	Number of within-species observations to simulate (default is 1).
nmissing	Number of observations to randomly remove from the simulated data (default is 0).
tree	Optional: an object of class phylo (if none provided, a random pure-birth phylogeny of size ntaxa is generated).
v	The evolutionary covariance matrix. If none provided, a random covariance matrix of size ntraits-by-ntraits is generated.
anc	An optional vector of ancestral values.
intraspecific	An optional vector of within-species variance.
model	The evolutionary model to simulate trait evolution (BM, OUrandomRoot, OUfixedRoot, EB, lambda, kappa, or delta).
parameters	The evolutionary model parameter in a named list. Example: <code>list(lambda = 0.5)</code>
nsim	Number of datasets to simulate.
return.type	"data.frame" or "matrix"

Value

trait_data	Simulated trait data
tree	The phylogenetic tree
sim_tree	The phylogenetic tree with specified branch length transformations

Author(s)

Eric W. Goolsby

References

Harmon Luke J, Jason T Weir, Chad D Brock, Richard E Glor, and Wendell Challenger. 2008. GEIGER: investigating evolutionary radiations. *Bioinformatics* 24:129-131.

See Also

[sim.char](#)

ultraFastAnc

*Ultra-fast maximum likelihood ancestral state reconstruction***Description**

This function performs ancestral state reconstruction using a fast algorithm based on phylogenetically independent contrasts (Felsenstein 1985). Contrary to [fastAnc](#), which reroots the tree at each internal node and recalculates the root state for each tree, this algorithm only requires a single tree traversal (Goolsby, In Review). This function is several hundred to several thousand times faster than the rerooting method used in [fastAnc](#), which is the next-fastest maximum likelihood ancestral state reconstruction method currently described (Revell 2012). As in the [fastAnc](#) function, the variance and 95% intervals of estimates are optionally returned.

NOTE: trees with polytomies are supported but (currently) slow down the algorithm considerably.

Usage

```
ultraFastAnc(phy, x, vars = FALSE, CI = FALSE)
```

Arguments

phy	A tree of class 'phylo'
x	A named vector of trait values
vars	Whether to return the variances of the (restricted) maximum likelihood estimates
CI	Whether to return 95% confidence intervals of the (restricted) maximum likelihood estimates

Value

A named vector of maximum likelihood ancestral states (with names corresponding to node number). If vars or CI is set to TRUE, a list is returned with these values included.

Author(s)

Eric W. Goolsby

References

Goolsby E.W. (In Review). "Ultra-fast ancestral state reconstruction of continuous characters: a rerooting-free maximum likelihood approach."

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

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

See Also

[fastAnc](#), [ace](#), [pic](#)

Examples

```
require(ape)
tree <- rtree(1e4) # random tree with 10,000 taxa
x <- setNames(rnorm(1e4),tree$tip.label) # random trait data
recon <- ultraFastAnc(phy=tree,x=x,CI=TRUE)
```

Index

- *Topic **PGLS**
 - phylocurve-package, 2
 - *Topic **\textasciitildekw1**
 - ultraFastAnc, 34
 - *Topic **\textasciitildekw2**
 - ultraFastAnc, 34
 - *Topic **ancestral state reconstruction**
 - phylocurve-package, 2
 - *Topic **dose-response curves**
 - phylocurve-package, 2
 - *Topic **evolution of tolerance**
 - phylocurve-package, 2
 - *Topic **function-valued traits**
 - phylocurve-package, 2
 - *Topic **infinite-dimensional traits**
 - phylocurve-package, 2
 - *Topic **phenotypic plasticity**
 - phylocurve-package, 2
 - *Topic **phylogenetic comparative method**
 - phylocurve-package, 2
 - *Topic **phylogenetic signal**
 - phylocurve-package, 2
 - *Topic **reaction norms**
 - phylocurve-package, 2
- ace, 34
- compare.evol.rates, 6–9, 11, 12, 14
- compare.models, 3, 7, 19
- compare.multi.evol.rates, 6, 7, 9–12, 14
- drm, 22
- evo.model, 4, 5, 19
- fast.geomorph.compare.evol.rates, 7
- fast.geomorph.compare.multi.evol.rates, 9
- fast.geomorph.phylo.integration, 11
- fast.geomorph.physignal, 12
- fast.geomorph.procD.pgl, 14
- fastAnc, 34
- get.aligned.function.data, 15
- get.tip.coefficients, 16
- getMeanFunctions, 22
- GP.fit, 17
- GP_fit, 18
- K.mult, 7, 18
- multipic, 19
- nonlinear.fit, 21
- paint.edges, 6, 23
- phylo.integration, 11, 12
- phylo.pls, 7, 9, 12, 14
- phylocurve, 24
- phylocurve-package, 2
- phylocurve.generalized, 25
- phylocurve.trim, 27
- physignal, 7, 9, 11–14, 19
- pic, 20, 30, 34
- polynomial.fit, 28
- prep_multipic, 29
- print.evo.model, 31
- procD.pgl, 7, 9, 11, 12, 14, 15
- sim.char, 32, 33
- sim.curves, 31
- sim.traits, 32
- ultraFastAnc, 34