Package ‘phylocurve’

June 14, 2015

Version 1.3.0

Date 2015-06-14

Title Phylogenetic Comparative Methods for Function-Valued and Other High-Dimensional Traits

Author Eric W. Goolsby

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

Description Tools for studying the evolution of function-valued traits (as well as other high-dimensional traits) 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`

Depends drc

Imports ape, geiger, phytools, abind, phylolm, GPfit, dtw

NeedsCompilation no

R topics documented:

phylocurve-package 2
compare.multivar.rate.mult 3
compare.rate.mult 4
get_aligned_function_data 5
get_tip_coefficients 6
GP.fit 7
K.mult 8
nonlinear.fit 9
pgls.mult 10
phylocurve 11
phylocurve.generalized 13
phylocurve.trim 14
polynomial.fit 15
print.lr.test 16
print.rate.mult 17
rate.mult 17
sim.curves 19
sim.mult 20

Index 22

`phylocurve-package` *Phylogenetic Comparative Methods for Function-Valued and Other High-Dimensional Traits*

Description

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

Details

Package:	phylocurve
Type:	Package
Version:	1.3.0
Date:	2015-06-14
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
rate.mult.fitted <- rate.mult(sim_data$tree,data)
```

`compare.multivar.rate.mult` *Compare multiple multivariate evolutionary rates*

Description

Compares hypotheses about evolutionary rates for multivariate traits combining methods from Adams (2013) and Adams (2014), using a covariance-based approach as described in Goolsby (2015).

Usage

```
compare.multivar.rate.mult(null_model, alt_model_list)
```

Arguments

`null_model`	A fitted rate.mult object in which all data is fit under the null hypothesis (e.g., all traits fit under the same evolutionary rate).
`alt_model_list`	A fitted rate.mult object in which all data is fit under the alternative hypothesis (e.g., traits are fit under different evolutionary rates).

Value

`null.logL`	Log-likelihood of the null model
`null.pars`	Number of parameters used to fit the null model
`alt.logL`	Log-likelihood of the alternative model
`alt.pars`	Number of parameters used to fit the alternative model
`df`	Degrees of freedom for likelihood ratio test
`chi_sq`	Chi-square value for likelihood ratio test
`p`	p-value for likelihood ratio test

Author(s)

Eric W. Goolsby

References

Adams D.C. 2013. Comparing evolutionary rates for different phenotypic traits on a phylogeny using likelihood. Systematic Biology. 62:181-192.

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

Golsby E.W. 2015. Covariance-based maximum likelihood estimation is equivalent to and more flexible than distance-based phylogenetic comparative methods for high-dimensional multivariate traits. In review.

See Also

`rate.mult`, `compare.rate.mult`, `compare.evol.rates`

Examples

```
# Simulate random data for 2 multivariate traits
# (5 dimensions each) with different evolutionary rates
s <- sim.mult(nspecies = 50,R = diag(c(rep(1,5),c(rep(2,5)))))

# Estimate the null model (a single rate parameter)
null <- rate.mult(tree = s$tree,Y = s$Y_means)

# Estimate components of the alternative hypothesis and store in a list
alt1 <- rate.mult(s$tree,Y = s$Y_means[,c(1,2:6)])
alt2 <- rate.mult(s$tree,Y = s$Y_means[,c(1,7:11)])
alt_list <- list(alt1,alt2)

# Compare the null to the alternative to hypothesis
comparison <- compare.multivar.rate.mult(null_model = null,alt_model_list = alt_list)
```

`compare.rate.mult` *Compare evolutionary rates*

Description

This function tests for differences in evolutionary rates (and optionally, evolutionary models) using the methods described in Adams (2014) and Goolsby (2015) in a covariance-based approach.

Usage

```
compare.rate.mult(rate.mult.fitted, groups, fit_individual = FALSE)
```

Arguments

`rate.mult.fitted`	A fitted rate.mult object in which all data is fit under the null hypothesis (e.g., all trait dimensions fit under the same evolutionary rate and/or evolutionary model).
`groups`	A named factor by species specifying which group to assign individual species. For example, given a tree with 16 species, the command setNames(object = factor(c(rep("a",8),rep("b",8)),levels=c("a","b")),nm = tree$tip.label) assigns 8 species to group "a" and 8 species to group "b".
`fit_individual`	Whether or not to fit individual evolutionary models to each trait dimension (if TRUE), or if a single evolutionary model is fit to all trait dimensions (if FALSE, the default). Ignored if model="BM".

Value

`null.logL`	Log-likelihood of the null model
`null.pars`	Number of parameters used to fit the null model
`alt.logL`	Log-likelihood of the alternative model
`alt.pars`	Number of parameters used to fit the alternative model

`df`	Degrees of freedom for likelihood ratio test
`chi_sq`	Chi-square value for likelihood ratio test
`p`	p-value for likelihood ratio test
`null_model_list`	List of fitted rate.mult objects for the null model corresponding to each group
`alt_model_list`	List of fitted rate.mult objects for the alternative model corresponding to each group

Author(s)

Eric W. Goolsby

References

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

Golsby E.W. 2015. Covariance-based maximum likelihood estimation is equivalent to and more flexible than distance-based phylogenetic comparative methods for high-dimensional multivariate traits. In review.

See Also

`rate.mult`, `compare.multivar.rate.mult`, `compare.evol.rates`

Examples

```
# Simulate random data for a multivariate trait with 50 species and 10 trait dimensions
s <- sim.mult(nspecies = 50,R = diag(10))

# Estimate the null model (a single rate parameter for the entire tree)
null <- rate.mult(tree = s$tree,Y = s$Y_means)

# Set 2 groups of species to test for different evolutionary rates
groups <- setNames(factor(c(rep("a",25),rep("b",25)),levels=c("a","b")),s$tree$tip.label)

# Test the hypothesis of different evolutionary rates for each group
compare.rate.mult(rate.mult.fitted = null,groups = groups)
```

`get_aligned_function_data`
Estimate aligned data for function-valued traits

Description

Estiamtes 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
rate.mult.fitted <- rate.mult(sim_data$tree,data)
```

`get_tip_coefficients` *Estimate regression coefficients for tip species curves*

Description

Estiamtes 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`	ether 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
rate.mult.fitted <- rate.mult(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)

# Perform comparative analyses on aligned curves
# rate.mult.fitted <- rate.mult(tree = P$tree,Y = P$aligned_data)
# K.mult(rate.mult.fitted)
```

`K.mult` *Multivariate phylogenetic signal*

Description

Estimates multivariate phylogenetic signal (Adams 2014) using a covariance-based impelmentation (Goolsby 2015).

Usage

```
K.mult(rate.mult.fitted, iter = 1000)
```

Arguments

`rate.mult.fitted`	A fitted rate.mult object
`iter`	Number of phylogenetic permutations for significance testing

Value

`phy.signal`	Multivariate phylogenetic signal (Kmult)
`pvalue`	Significance of Kmult

Author(s)

Eric W. Goolsby

References

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.

Golsby E.W. 2015. Covariance-based maximum likelihood estimation is equivalent to and more flexible than distance-based phylogenetic comparative methods for high-dimensional multivariate traits. In review.

See Also

`physignal`

Examples

```
# Simulate random data for a multivariate trait with 50 species and 10 trait dimensions
s <- sim.mult(nspecies = 50,R = diag(10))

# Estimate evolutionary rates
rate.mult.fitted <- rate.mult(tree = s$tree,Y = s$Y_means)

# Estimate multivariate phylogenetic signal
K.mult(rate.mult.fitted = rate.mult.fitted)
```

`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

`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
`fct`	Function type to pass to `drm`. Use getMeanFunctions for a full list.
`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 drm 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
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)

# Perform comparative analyses on aligned curves
# rate.mult.fitted <- rate.mult(tree = P$tree,Y = P$aligned_data)
# K.mult(rate.mult.fitted)
```

`pgls.mult` *Multivariate phylogenetic generalized least squares*

Description

Performs phylogenetic generalized least squares using the method described in Adams (2014) using a covariance-based approach (Goolsby 2015)

Usage

```
pgls.mult(rate.mult.fitted, X)
```

Arguments

`rate.mult.fitted`	A fitted rate.mult object
`X`	A matrix of predictor variables X for PGLS. Note, an intercept is estimated automatically (do not add a column of ones to X). Rows should be named according to species names.

Value

An object of class "anova"

Author(s)

Eric W. Goolsby

References

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

Golsby E.W. 2015. Covariance-based maximum likelihood estimation is equivalent to and more flexible than distance-based phylogenetic comparative methods for high-dimensional multivariate traits. In review.

See Also

`procD.pgls`

Examples

```
# Simulate random data for a multivariate trait with 50 species and 10 trait dimensions
s <- sim.mult(nspecies = 50,R = diag(10))

# Simulate a univariate trait
require(phytools)
X <- as.matrix(fastBM(s$tree))

# Estimate evolutionary rates
rate.mult.fitted <- rate.mult(tree = s$tree,Y = s$Y_means)

# Perform multivariate phylogenetic regression
pgls.mult(rate.mult.fitted = rate.mult.fitted,X = X)
```

`phylocurve` *Ancestral curve reconstruction for logistic regression (glm with logit link)*

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~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 paramter.
`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 are (fitted_x[,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 are (fitted_x[,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
rate.mult.fitted <- rate.mult(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 marix 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$anc_Y)
#
# P.trimmed <- phylocurve.trim(phylocurve.generalized = P,min_X = 4,max_X = 8)
#
# Perform comparative analyses on trimmed curves
# rate.mult.fitted <- rate.mult(tree = P.trimmed$tree,Y = P.trimmed$aligned_data)
# K.mult(rate.mult.fitted)
```

`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$anc_Y)
#
# P.trimmed <- phylocurve.trim(phylocurve.generalized = P,min_X = 4,max_X = 8)
#
# Perform comparative analyses on trimmed curves
# rate.mult.fitted <- rate.mult(tree = P.trimmed$tree,Y = P.trimmed$aligned_data)
# K.mult(rate.mult.fitted)
```

`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)

# Perform comparative analyses on aligned curves
# rate.mult.fitted <- rate.mult(tree = P$tree,Y = P$aligned_data)
# K.mult(rate.mult.fitted)
```

`print.lr.test` *Print lr.test*

Description

Generic S3 method for lr.test

Usage

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

Arguments

`x`	An object of class lr.test
`...`	Further arguments passed to or from other methods.

`print.rate.mult` *Print rate.mult*

Description

Generic S3 method for rate.mult

Usage

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

Arguments

`x`	An object of class rate.mult
`...`	Further arguments passed to or from other methods.

`rate.mult` *Estimate multivariate evolutionary rates and models*

Description

This function estimates multivariate evolutionary rates (Adams 2014) and alternative evolutionary models using a covariance-based approach (Goolsby 2015). The function must be run prior to running other multivariate functions, including K.mult, pgls.mult, compare.rate.mult, and compare.multivar.rate.mult.

Usage

```
rate.mult(tree = tree, Y = Y, type = c("mult", "diag", "all"),
method = c("REML", "ML"), error = c("none", "estimate", "supply"),
error_n = 20, error_supply, model = "BM", fixed_sigma2, fixed_model_pars)
```

Arguments

`tree`	An object of class phylo
`Y`	A data frame in which the first column is labeled "species", and is filled with species names. The remaining columns contain values for the multivariate trait Y. Multiple observations per species (using multiple rows) and missing data are allowed.
`type`	The form of evolutionary rate estimation to use. If type="mult", a single evolutionary rate is estimated for all trait dimensions. If type="diag", a separate evolutionary rate is estimated for all trait dimensions. NOTE: the option type="all" is also available, in which an entire covariance matrix R is estimated for traits (not compatbile with other functions!), in which case, only BM evolution with no missing data and no within-species error is allowed.
`method`	Whether to use restricted maximum likelihood (REML, the default) or maximum likelihood.
`error`	Whether to incorporate within-species measurement error. Can be set to "none" (the default) if only a single observation is available per species. If raw data with multiple observations per species is supplied, select "estimate" to estimate pooled within-species variance for species with less than error_n observations (for species with >error_n observations, variance is used). Alternatively, a matrix of the square of standard errors (variance divided by the number of within-species observations) for each species and trait dimension can be supplied if error is set to "supply".
`error_n`	Only relevant if error="estimate". For trait dimensions with less than error_n within-species observations, pooled within-species variance is used. If the number of within-species observations for a trait dimension is greater than or equal to error_n, variance is estimated directly.
`error_supply`	Only relevent if error="supply". A matrix of species and trait dimension squared standard errors (or variance divided by the number of within-species observations). Row names should be species names.
`model`	Evolutionary model specification – default is "BM". Other options include "OUrandomRoot", "OUfixedRoot", "EB", "lambda", "kappa", and "delta". More than one model may be specified (e.g., c("lambda","EB")), in which case the branch length transformations are applied in order of the specified models. NOTE: if incorporating alternative models, DO NOTE include "BM".
`fixed_sigma2`	If the evolutionary rate (single value if type="mult") or rates (if type="diag") is desired to be fixed, entire it here.
`fixed_model_pars`	If the evolutionary model parameter or parameters are desired to be fixed, entire them here in a vector named according to the appropraite model (should be in order of the models specified by the model argument).

Value

`sigma2`	Estimated evolutionary rate
`pars`	Estimated evolutionary model parameters (NULL If model="BM")
`logL`	Log-likelihood of the estimated parameters
`method`	REML or ML
`model`	Evoluionary model
`rate.mult.args`	List of arguments to be called with do.call(rate.mult,rate.mult$rate.mult.args)

Author(s)

Eric W. Goolsby

References

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

Golsby E.W. 2015. Covariance-based maximum likelihood estimation is equivalent to and more flexible than distance-based phylogenetic comparative methods for high-dimensional multivariate traits. In review.

See Also

`compare.multivar.rate.mult`, `compare.rate.mult`, `compare.evol.rates`

Examples

```
# Simulate random data for a multivariate trait with 50 species and 10 trait dimensions
s <- sim.mult(nspecies = 50,R = diag(10))

# Estimate the null model (a single rate parameter for the entire tree)
null <- rate.mult(tree = s$tree,Y = s$Y_means)

# Set 2 groups of species to test for different evolutionary rates
groups <- setNames(factor(c(rep("a",25),rep("b",25)),levels=c("a","b")),s$tree$tip.label)

# Test the hypothesis of different evolutionary rates for each group
compare.rate.mult(rate.mult.fitted = null,groups = groups)
```

`sim.curves` *Simulate function-valued curve evolution*

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, startree = FALSE, lambda = 1, seed)
```

Arguments

`nspecies`	Number of species
`x_length`	Number of points on the curve to simulate
`startree`	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
rate.mult.fitted <- rate.mult(sim_data$tree,data)
```

`sim.mult` *Simulate multivariate trait evolution*

Description

Simulates multivariate trait evolution

Usage

```
sim.mult(nspecies, R, error, nreps = 1, nmissing = 0,
model, parameters, anc, tree, seed, nsims = 1)
```

Arguments

`nspecies`	Number of species to simulate
`R`	Covariance matrix for traits to simulate
`error`	Vector of variances to simulate within-species variation for each trait
`nreps`	Number of replicates to simulate for each trait per species
`nmissing`	Number of data points to randomly remove
`model`	Model of evolution (if not "BM", can be "OUrandomRoot", "OUfixedRoot", "EB", "lambda", "kappa", or "delta")
`parameters`	List of evolutionary model parameters. For either OU model, use alpha. For EB, use rate. For lambda, kappa, or delta, use lambda, kappa, or delta.
`anc`	Vector of ancestral values for each trait (default is 0).
`tree`	Can provide phylogenetic tree for simulations here; otherwise it is randomly generated based on nspecies
`seed`	Whether or not to set a seed for reproducible results
`nsims`	Number of simulations (default is 1)

Value

`Y_means`	Means of simulated data
`Y_raw`	Raw simulated data (only different from means if nreps>1)
`Y_array`	Means of simulated data in array format
`Y_original`	Original simulated species means (only different if nerps>1)
`tree`	An object of class phylo
`tree_sim`	The tree on which data was simulated (only different if model!="BM")

Author(s)

Eric W. Goolsby

Examples

```
# Simulate random data for a multivariate trait with 50 species and 10 trait dimensions
s <- sim.mult(nspecies = 50,R = diag(10))

# Estimate evolutionary rates
rate.mult.fitted <- rate.mult(tree = s$tree,Y = s$Y_means)

# Estimate multivariate phylogenetic signal
K.mult(rate.mult.fitted = rate.mult.fitted)
```

Index

∗Topic **PGLS**
    phylocurve-package, 2
∗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 **infnite-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

compare.evol.rates, *4*, *5*, *19*
compare.multivar.rate.mult, 3, *5*, *19*
compare.rate.mult, *4*, 4, *19*

drm, *10*

get_aligned_function_data, 5
get_tip_coefficients, 6
getMeanFunctions, *10*
GP.fit, 7
GP_fit, *8*

K.mult, 8

nonlinear.fit, 9

pgls.mult, 10
phylocurve, 11
phylocurve-package, 2
phylocurve.generalized, 13
phylocurve.trim, 14
physignal, *9*
polynomial.fit, 15
print.lr.test, 16
print.rate.mult, 17
procD.pgls, *11*

rate.mult, *4*, *5*, 17

sim.curves, 19
sim.mult, 20