

Package ‘OUwie’

June 11, 2019

Version 1.53

Date 2019-06-10

Title Analysis of Evolutionary Rates in an OU Framework

Author Jeremy M. Beaulieu <jbeaulieu@nimbios.org>, Brian O'Meara <bomeara@utk.edu>

Maintainer Jeremy Beaulieu <jbeaulieu@nimbios.org>

Depends R (>= 3.2.0), ape, nloptr, lattice, grDevices, geiger

Imports numDeriv, corpcor, phytools, paleotree, phangorn, stats, expm,
corHMM, Rmpfr

Description Calculates and compares rate differences of continuous character evolution under Brownian motion and a new set of Ornstein-Uhlenbeck based Hansen models that allow the strength of selection and stochastic motion to vary across selective regimes. Beaulieu et al (2012) <doi:10.1111/j.1558-5646.2012.01619.x>.

License GPL (>= 2)

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2019-06-11 16:20:08 UTC

R topics documented:

Example	2
OUwie	2
OUwie.anc	6
OUwie.boot	8
OUwie.fixed	10
OUwie.joint	12
OUwie.sim	15
OUwie.slice	17

Index	21
--------------	-----------

Example	<i>An example dataset</i>
---------	---------------------------

Description

An example dataset containing a 64-tip birth-death tree with internal node labels denoting two selective regimes, and a trait file in the proper format: 1) Genus_species, 2) current selective regime, 3) continuous trait data.

Format

a tree of class “phylo” and a data frame with 3 columns and 64 rows

OUwie	<i>Generalized Hansen models</i>
-------	----------------------------------

Description

Fits generalized Ornstein-Uhlenbeck-based Hansen models of continuous characters evolving under discrete selective regimes.

Usage

```
OUwie(phy, data, model=c("BM1", "BMS", "OU1", "OUM", "OUMV", "OUMA", "OUMVA",
  "TrendyM", "TrendyMS"), simmap.tree=FALSE, root.age=NULL, scaleHeight=FALSE,
  root.station=TRUE, clade=NULL, mserr="none", starting.vals=NULL, diagn=FALSE,
  quiet=FALSE, warn=TRUE)
```

Arguments

phy	a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.
data	a data.frame containing species information (see Details).
model	models to fit to comparative data (see Details).
simmap.tree	a logical indicating whether the input tree is in SIMMAP format. The default is FALSE.
root.age	indicates the age of the tree. This is to be used in cases where the "tips" are not contemporary, such as in cases for fossil trees. Default is NULL meaning latest tip is modern day.
scaleHeight	a logical indicating whether the total tree height should be scaled to 1 (see Details). The default is FALSE.
root.station	a logical indicating whether the starting state, θ_0 , should be estimated (see Details).

<code>clade</code>	a list containing a pair of taxa whose MRCA is the clade of interest (see Details).
<code>mserr</code>	designates whether a fourth column in the data matrix contains measurement error for each species value ("known"). The measurement error is assumed to be the standard error of the species mean. The default is "none".
<code>starting.vals</code>	a vector of initial values for the optimization search. For OU models, two must be supplied, with the first being the initial alpha value and the second being the initial sigma squared. For BM models, just a single value is needed.
<code>diagn</code>	a logical indicating whether the full diagnostic analysis should be carried out. The default is FALSE.
<code>quiet</code>	a logical indicating whether progress should be written to the screen. The default is FALSE.
<code>warn</code>	a logical indicating whether a warning should be printed if the number of parameters exceeds <code>ntips/10</code> . The default is TRUE.

Details

This function fits various likelihood models for continuous characters evolving under discrete selective regimes. The function returns parameter estimates and their approximate standard errors. The R package `nloptr` provides a common interface to `NLOpt`, an open-source library for nonlinear optimization. The likelihood function is maximized using the bounded subplex optimization routine (`NLOPT_LN_SBPLX`). As input all `OUwie` requires is a tree and a trait `data.frame`. The tree must be of class "phylo" and must contain the ancestral selective regimes as internal node labels. Internal node labels can be applied manually or from some sort of ancestral state reconstruction procedure (`BayesTraits`, `ape`, `diversitree`, `SIMMAP`, etc.), which would then be brought into `OUwie`. This is essentially what is required by `ouch` and `Brownie` (though `Brownie` provides built-in ancestral state reconstruction capabilities). The trait `data.frame` must have column entries in the following order: [,1] species names, [,2] current selective regime, and [,3] the continuous trait of interest. Alternatively, if the user wants to incorporate measurement error (`mserr="known"`), then a fourth column, [,4] must be included that provides the standard error estimates for each species mean. However, a global measurement error for all taxa can be estimated from the data (`mserr="est"`); is not well tested, so use at your own risk. Also, a user can specify a particular clade as being in a different selective regime, by inputting a pair of species whose `mrca` is the root of the clade of interest [e.g., `clade=c("taxaA","taxaB")`]. `OUwie` will automatically assign internal node labels and update the data matrix according to this clade designation.

The initial implementation followed `ouch` in that the tree is automatically rescaled so that the branch lengths were in proportion to the total height of the tree. However, this makes the results inconsistent with other implementations such as `Brownie` or `geiger`. Therefore, we allow the user to choose whether the tree should be rescaled or not. Note that the when `scaleHeight=FALSE` the bounds will have to be adjusted to the appropriate scale.

Possible models are as follows: single-rate Brownian motion (`model=BM1`), Brownian motion with different rate parameters for each state on a tree (`model=BMS`), Ornstein-Uhlenbeck model with a single optimum for all species (`model=OU1`), Ornstein-Uhlenbeck model with different state means and a single α and σ^2 acting all selective regimes (`model=OUM`), and new Ornstein-Uhlenbeck models that assume different state means as well as either multiple σ^2 (`model=OUMV`), multiple α (`model=OUMA`), or multiple α and σ^2 per selective regime (`model=OUMVA`).

If `root.station` is TRUE (the default), θ_0 is dropped from the model. Under these conditions it is assumed that the starting value is distributed according to the stationary distribution of the OU

process. This would not fit a biological scenario involving moving away from an ancestral state, but it does fit a scenario of evolution at a steady state. Dropping θ_0 from the model can sometimes stabilize estimates of the primary optima, especially in situations where the estimates of θ in the full model are non-sensical. In regards to the accuracy of estimating θ_0 , it is important to note that in simulation, as α increases estimates of θ_0 converge to zero. Thus, when α is large (i.e. $\alpha > 2$) it is likely that any inference of an evolutionary trend will be an artifact and positively misleading.

Also note, when specifying the BMS model be mindful of the `root.station` flag. When `root.station=FALSE`, the non-censored model of O'Meara et al. 2006 is invoked (i.e., a single regime at the root is estimated), and when `root.station=TRUE` the group mean model of Thomas et al. 2006 (i.e., the number of means equals the number of regimes). The latter case appears to be a strange special case of OU, in that it behaves similarly to the OUMV model, but without selection. I would say that this is more consistent with the censored test of O'Meara et al. (2006), as opposed to having any real connection to OU. In any case, more work is clearly needed to understand the behavior of the group means model, and therefore, I recommend setting `root.station=FALSE` in the BMS case.

The Hessian matrix is used as a means to estimate the approximate standard errors of the model parameters and to assess whether they are the maximum likelihood estimates. The variance-covariance matrix of the estimated values of α and σ^2 are computed as the inverse of the Hessian matrix and the standard errors are the square roots of the diagonals of this matrix. The Hessian is a matrix of second-order derivatives and is approximated in the R package `numDeriv`. So, if changes in the value of a parameter results in sharp changes in the slope around the maximum of the log-likelihood function, the second-order derivative will be large, the standard error will be small, and the parameter estimate is considered stable. On the other hand, if the second-order derivative is nearly zero, then the change in the slope around the maximum is also nearly zero, indicating that the parameter value can be moved in any direction without greatly affecting the log-likelihood. In such situations, the standard error of the parameter will be large.

For models that allow α and σ^2 to vary (i.e., OUMV, OUMA, and OUMVA), the complexity of the model can often times be greater than the information that is contained within the data. As a result one or many parameters are poorly estimated, which can cause the function to return a log-likelihood that is suboptimal. This has great potential for poor model choice and incorrect biological interpretations. An eigendecomposition of the Hessian can provide an indication of whether the search returned the maximum likelihood estimates. If all the eigenvalues of the Hessian are positive, then the Hessian is positive definite, and all parameter estimates are considered reliable. However, if there are both positive and negative eigenvalues, then the objective function is at a saddlepoint and one or several parameters cannot be estimated adequately. One solution is to just fit a simpler model. Another is to actually identify the offending parameters. This can be done through the examination of the eigenvectors. The row order corresponds to the entries in `index.matrix`, the columns correspond to the order of values in `eigval`, and the larger the value of the row entry the greater the association between the corresponding parameter and the eigenvalue. Thus, the largest values in the columns associated with negative eigenvalues are the parameters that are causing the objective function to be at a saddlepoint.

Value

OUwie returns an object of class `OUwie`. This is a list with elements:

<code>\$loglik</code>	the maximum log-likelihood.
<code>\$AIC</code>	Akaike information criterion.
<code>\$AICc</code>	Akaike information criterion corrected for sample-size.

\$BIC	Bayesian information criterion.
\$model	The model being fit
\$param.count	The number of parameters counted in the model.
\$solution	a matrix containing the maximum likelihood estimates of α and σ^2 .
\$theta	a matrix containing the maximum likelihood estimates of θ and its standard error.
\$solution.se	a matrix containing the approximate standard errors of α and σ^2 . The standard error is calculated as the diagonal of the inverse of the Hessian matrix.
\$tot.state	A vector of names for the different regimes
\$index.mat	The indices of the parameters being estimated are returned. The numbers correspond to the row in the eigvect and can useful for identifying the parameters that are causing the objective function to be at a saddlepoint (see Details)
\$simmap.tree	A logical indicating whether the input phylogeny is a SIMMAP formatted tree.
\$root.age	The user-supplied age at the root of the tree.
\$opts	Internal settings of the likelihood search
\$data	User-supplied dataset
\$phy	User-supplied tree
\$root.station	A logical indicating whether the starting state, θ_0 , was estimated
\$starting.vals	A vector of user-supplied initial search parameters.
\$lb	The lower bound set
\$ub	The upper bound set
\$iterations	Number of iterations of the likelihood search that were executed
\$mserr.est	The estimated measurement error if mserr="est". Otherwise, the value is NULL.
\$res	A vector of residuals from the model fit. The residuals are ordered in the same way as the tips in the tree.
\$eigval	The eigenvalues from the decomposition of the Hessian of the likelihood function. If any eigval<0 then one or more parameters were not optimized during the likelihood search (see Details)
\$eigvect	The eigenvectors from the decomposition of the Hessian of the likelihood function is returned (see Details)

Author(s)

Jeremy M. Beaulieu and Brian C. O'Meara

References

- Beaulieu J.M., Jhwueng D.C., Boettiger C., and O'Meara B.C. 2012. Modeling stabilizing selection: Expanding the Ornstein-Uhlenbeck model of adaptive evolution. *Evolution* 66:2369-2383.
- O'Meara B.C., Ane C., Sanderson P.C., Wainwright P.C. 2006. Testing for different rates of continuous trait evolution using likelihood. *Evolution* 60:922-933.

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

Thomas G.H., Freckleton R.P., and Szekely T. 2006. Comparative analysis of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society, B.* 273:1619-1624.

Examples

```
data(tworegime)

#Plot the tree and the internal nodes to highlight the selective regimes:
select.reg<-character(length(tree$node.label))
select.reg[tree$node.label == 1] <- "black"
select.reg[tree$node.label == 2] <- "red"
plot(tree)
nodelabels(pch=21, bg=select.reg)

#Not run
#To see the first 5 lines of the data matrix to see what how to
#structure the data:
#trait[1:5,]

#Now fit an OU model that allows different sigma^2:
#OUwie(tree,trait,model=c("OUMV"),root.station=TRUE)

#Fit an OU model based on a clade of interest:
#OUwie(tree,trait,model=c("OUMV"), root.station=TRUE, clade=c("t50", "t64"))
```

OUwie.anc

Estimate ancestral states given a fitted OUwie model

Description

Fits ancestral states (a joint estimate) given a fitted OUwie model. Currently, only works for trees with regimes painted on as node labels (rather than a simmap tree). The intended use case is just to visualize what the model is saying about evolution to help intuition (is the model something you can believe in?) rather than a firm estimate you should use to say, Yes, 56.4 MY, the ancestral body size was 17.34 mm. It could likely be anywhere from 1 to 100 mm with about equal chance. Please read details before using this function.

Usage

```
OUwie.anc(fitted.OUwie.object, opts = list("algorithm"="NLOPT_LN_SBPLX",
"maxeval"="1000", "ftol_rel"=.Machine$double.eps^0.5), knowledge=FALSE)
```

Arguments

fitted.OUwie.object	an object returned from the OUwie function
opts	a list of options to pass to nloptr for the ancestral state optimization
knowledge	a logical indicating whether or not you have read the documentation The default is FALSE.

Details

You probably DON'T want to use this function for anything more serious than poking around to make sure the data and model look right (oh, golly: my tips are all between 5 and 15 mm in size, and the ancestral states are 674 mm.). The request to implement ancestral state estimation resulted in many important comments from experts in the public R-SIG-PHYLO discussion forum. Here is a sampling; to see them all, go to [R-SIG-PHYLO](#).

So in short, yes, you can do it, with any number of methods. But why? If you can answer your biological question with methods that do not involve estimation of a parameter that is inherently fraught with error, it might be better to go another way. Bottom line - use caution and be thoughtful!
– Marguerite Butler

I would add an extra caveat to Marguerite's excellent post: Most researchers work with extant taxa only, ignoring extinction. This causes a massive ascertainment bias, and the character states of the extinct taxa can often be very different to the ancestral state reconstructions, particularly if the evolutionary model is wrong. Eg. there has been an evolutionary trend for example. Ancestral state reconstructions based only on extant taxa should be treated as hypotheses to be tested with fossil data. I wouldn't rely on them for much more. – Simone Blomberg

While I am at it, let me echo Simone and Marguerite's warnings. The predicted ancestral states will reflect the process you assumed to predict them. Hence, if you use them to make inferences about evolution, you will recover your own assumptions. I.e. if you predict from a model with no trend, you will find no trend, etc. Many comparative studies are flawed for this reason. – Thomas Hansen

Let me add more warnings to Marguerite and Thomas's excellent responses. People may be tempted to infer ancestral states and then treat those inferences as data (and also to infer ancestral environments and then treat those inferences as data). In fact, I wonder whether that is not the main use people make of these inferences. But not only are those inferences very noisy, they are correlated with each other. So if you infer the ancestral state for the clade (Old World Monkeys, Apes) and also the ancestral state for the clade (New World Monkeys, (Old World Monkeys, Apes)) the two will typically not only be error-prone, but will also typically be subject to strongly correlated errors. Using them as data for further inferences is very dubious. It is better to figure out what your hypothesis is and then test it on the data from the tips of the tree, without the intermediate step of taking ancestral state inferences as observations. The popular science press in particular demands a fly-on-the-wall account of what happened in evolution, and giving them the ancestral state inferences as if they were known precisely is a mistake. – Joe Felsenstein

The minor twist I would throw in is that it's difficult to make universal generalizations about the quality of ancestral state estimation. If one is interested in the ancestral state value at node N, it might be reasonably estimated if it is nested high up within the phylogeny, if the rates of change aren't high, etc. And (local) trends etc might well be reliably inferred. We are pretty confident that the common ancestor of humans and chimps was larger than many deeper primate ancestors, for

instance. If N is the root of your available phylogeny, however, you have to be much more cautious.
– Nick Matzke

I'll also add that I think there's a great deal to be skeptical of ancestral trait reconstruction even when large amounts of fossil data is available. You can try the exercise yourself: simulate pure BM on a non-ultrametric tree with lots of 'extinct' tips, and you'll still find pretty large confidence intervals on the estimates of the trait values. What does it mean to do ancestral trait reconstruction, if our calculations of uncertainty are that broad? – Dave Bapst

These are some of the people who best know the power and limitations of the OU model in phylogenetics. Heed them!

To ensure that you've read this before use, please pass `knowledge=TRUE` as an argument to the function.

Value

`OUwie.anc` returns an object of class `OUwie.anc`. This is an `OUwie` object but with terminal species added at each node representing the ancestral states at each node (which are also included in the data object). There is also a `NodeRecon` element in the list that has the optimal ancestral states. There is not currently an estimate of uncertainty, but it is substantial.

Author(s)

Brian C. O'Meara

Examples

```
data(tworegime)
fitted <- OUwie(tree,trait,model=c("OUMV"),root.station=TRUE)
recon <- OUwie.anc(fitted, knowledge=FALSE) # read the description above to make this work.
plot(recon)
```

OUwie.boot

Parametric bootstrap function

Description

A function that performs a parametric bootstrap for a set of user-specified model parameters

Usage

```
OUwie.boot(phy, data, model=c("BM1", "BMS", "OU1", "OUM", "OUMV", "OUMA", "OUMVA"),
  nboot=100, alpha, sigma.sq, theta, theta0, simmap.tree=FALSE, root.age=NULL,
  scaleHeight=FALSE, root.station=TRUE, clade=NULL, mserr="none", diagn=FALSE,
  quiet=TRUE, warn=FALSE)
```

Arguments

phy	a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.
data	a data matrix containing species information.
model	models to fit to comparative data.
nboot	The number of bootstrap replicates.
alpha	a numeric vector giving the values of α for each selective regime.
sigma.sq	a numeric vector giving the values of σ^2 for each selective regime.
theta	a numeric vector giving the values of θ for each selective regime.
theta0	a numeric indicating the starting state, θ_0
simmap.tree	a logical indicating whether the input tree is in SIMMAP format. The default is FALSE.
root.age	indicates the age of the tree. This is to be used in cases where the "tips" are not contemporary, such as in cases for fossil trees. Default is NULL meaning latest tip is modern day.
scaleHeight	a logical indicating whether the total tree height should be scaled to 1 (see Details). The default is FALSE.
root.station	a logical indicating whether the starting state, θ_0 , should be estimated (see Details).
clade	a list containing a pair of taxa whose MRCA is the clade of interest.
mserr	designates whether a fourth column in the data matrix contains measurement error for each species value ("known"). The measurement error is assumed to be the standard error of the species mean. The default is "none".
diagn	a logical indicating whether the full diagnostic analysis should be carried out. The default is FALSE.
quiet	a logical indicating whether progress should be written to the screen. The default is TRUE.
warn	a logical indicating whether a warning should be printed if the number of parameters exceeds ntips/10. The default is FALSE.

Details

A simple function for conducting a parametric bootstrap on parameters estimated in OUwie. As before, the input is a tree and a data file. The tree must be of class “phylo” and if `simmap=FALSE` must contain the ancestral selective regimes as internal node labels. The data file is a dataframe that must have column entries in the following order: [,1] species names and [,2] their current selective regime. The user specifies the simulated parameter values (i.e. α , σ^2 , θ_0 , θ), which is assumed to be the maximum likelihood estimates obtained from an OUwie run.

Note that if `root.station` is TRUE (the default), θ_0 was dropped from the model. In this case, then, θ_0 should be set to the value of the selective regime mapped at the root (i.e., state 1 in the “tworegime” example dataset).

Value

OUwie.boot returns an object of class OUwie.boot. This is a matrix of column length equal to the number of parameters, and row length of the number of bootstrap replicates specified.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu J.M., Jhwueng D.C., Boettiger C., and O'Meara B.C. 2012. Modeling stabilizing selection: Expanding the Ornstein-Uhlenbeck model of adaptive evolution. *Evolution* 66:2369-2383.

O'Meara B.C., Ane C., Sanderson P.C., Wainwright P.C. 2006. Testing for different rates of continuous trait evolution using likelihood. *Evolution* 60:922-933.

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

Examples

```
data(tworegime)

##First step is estimate parameters under a particular model:
#pp <- OUwie(tree,trait,model=c("OUMV"),root.station=TRUE)

##Second step is to run bootstrap replicates:
#boot.reps <- OUwie.boot(tree,trait,model="OUMV", nboot=10, alpha=pp$solution[1,],
#sigma.sq=pp$solution[2,],theta=pp$theta[1,], theta0=pp$theta[1,1])

##Finally summarize to obtain the desired confidence -- here is the 95% CI:
#apply(boot.reps, 2, quantile, probs=c(0.025,0.975))
```

OUwie.fixed

Generalized Hansen model likelihood calculator

Description

Allows the user to calculate the likelihood given a specified set of parameter values

Usage

```
OUwie.fixed(phy, data, model=c("BM1", "BMS", "OU1", "OUM", "OUMV", "OUMA", "OUMVA"),
simmap.tree=FALSE, root.age=NULL, scaleHeight=FALSE, root.station=TRUE, alpha=NULL,
sigma.sq=NULL, theta=NULL, clade=NULL, mserr="none", quiet=FALSE)
```

Arguments

phy	a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes
data	a dataframe containing species information (see Details)
model	models to fit to comparative data (see Details).
simmap.tree	a logical indicating whether the input tree is in SIMMAP format. The default is FALSE.
root.age	indicates the age of the tree. This is to be used in cases where the "tips" are not contemporary, such as in cases for fossil trees. Default is NULL meaning latest tip is modern day.
scaleHeight	a logical indicating whether the total tree height should be scaled to 1 (see Details). The default is FALSE.
root.station	a logical indicating whether the starting state, θ_0 , should be estimated.
alpha	a numeric vector giving the values of α for each selective regime
sigma.sq	a numeric vector giving the values of σ^2 for each selective regime
theta	a numeric vector giving the values of θ for each selective regime
clade	a list containing a pair of taxa whose MRCA is the clade of interest.
mserr	designates whether a fourth column in the data matrix contains measurement error for each species value ("known"). The measurement error is assumed to be the standard error of the species mean. The default is "none".
quiet	a logical indicating whether or not to print progress to the screen. The default is "FALSE".

Details

The input is a tree and a data file. The tree must be of class “phylo” and must contain the ancestral selective regimes as internal node labels. The data file is a data.frame that must have column entries in the following order: [,1] species names and [,2] their current selective regime. The user specifies the parameter values (i.e. α , σ^2 , and θ).

Value

OUwie.fixed returns an object of class OUwie.fixed. This is a list with elements:

\$loglik	the maximum log-likelihood.
\$AIC	Akaike information criterion.
\$AICc	Akaike information criterion corrected for sample-size.
\$model	The model being fit
\$param.count	The number of parameters counted in the model.
\$solution	a matrix containing the maximum likelihood estimates of α and σ^2 .
\$theta	a matrix containing the maximum likelihood estimates of θ .
\$tot.state	A vector of names for the different regimes

<code>\$index.mat</code>	The indices of the parameters being estimated are returned. The numbers correspond to the row in the <code>eigvect</code> and can be useful for identifying the parameters that are causing the objective function to be at a saddlepoint (see Details)
<code>\$simmmap.tree</code>	A logical indicating whether the input phylogeny is a SIMMAP formatted tree.
<code>\$root.age</code>	The user-supplied age at the root of the tree.
<code>\$data</code>	User-supplied dataset
<code>\$phy</code>	User-supplied tree
<code>\$root.station</code>	A logical indicating whether the starting state, θ_0 , was estimated
<code>\$res</code>	A vector of residuals from the model fit. The residuals are ordered in the same way as the tips in the tree.

Author(s)

Jeremy M. Beaulieu

Examples

```
data(tworegime)

#Calculate the likelihood based on known values of
#alpha, sigma^2, and theta:
alpha=c(0.5632459,0.1726052)
sigma.sq=c(0.1064417,0.3461386)
theta=c(1.678196,0.4185894)

OUwie.fixed(tree,trait,model=c("OUMVA"), simmap.tree=FALSE, scaleHeight=FALSE,
clade=NULL, alpha=alpha,sigma.sq=sigma.sq,theta=theta)
```

OUwie.joint

Joint optimization for multiple traits

Description

Fits generalized Ornstein-Uhlenbeck-based Hansen models for multiple continuous characters evolving under discrete selective regimes.

Usage

```
OUwie.joint(phy, data, model=c("BM1", "BMS", "OU1", "OUM", "OUMV",
"OUMVr", "OUMA", "OUMAr", "OUMVA", "OUMVAr"), ntraits, allfree=TRUE,
simmmap.tree=FALSE, root.age=NULL, scaleHeight=FALSE,
root.station=TRUE, mserr="none", diagn=FALSE, quiet=FALSE)
```

Arguments

phy	a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.
data	a dataframe containing species information (see Details).
model	models to fit to comparative data (see Details).
ntraits	a numeric indicating the number of traits to analyze.
allfree	a logical indicating whether the parameters vary across the multiple traits (see Details). The default is TRUE.
simmmap.tree	a logical indicating whether the input tree is in SIMMAP format. The default is FALSE.
root.age	indicates the age of the tree. This is to be used in cases where the "tips" are not contemporary, such as in cases for fossil trees. Default is NULL meaning latest tip is modern day.
scaleHeight	a logical indicating whether the total tree height should be scaled to 1 (see Details). The default is FALSE.
root.station	a logical indicating whether the starting state, θ_0 , should be estimated (see Details).
mserr	designates whether the data matrix contains measurement error for each species value ("known"). The default is "none".
diagn	a logical indicating whether the full diagnostic analysis should be carried out. The default is FALSE.
quiet	a logical indicating whether progress should be written to the screen. The default is FALSE.

Details

This function jointly estimates parameters under various likelihood models for several continuous characters evolving under discrete selective regimes. As with the regular OUwie function all that is required is a tree and a trait data.frame. The tree must be of class “phylo” and must contain the ancestral selective regimes as internal node labels. Internal node labels can be applied manually or from some sort of ancestral state reconstruction procedure (BayesTraits, ape, diversitree, SIMMAP, etc.), which would then be brought into OUwie. The trait data.frame must have column entries in the following order: [,1] species names, [,2] current selective regime, with all other columns containing the set of continuous traits to optimize jointly. Currently, this function does not allow for measurement error to be incorporated, but future versions will (mserr="known"). Finally, a user can simply specify a particular clade as being in a different selective regime, by inputting a pair of species whose mrca is the root of the clade of interest [e.g., clade=c("taxaA", "taxaB")]. OUwie will automatically assign internal node labels and update the data matrix according to this clade designation.

Possible models are the same as before: single-rate Brownian motion (model=BM1), Brownian motion with different rate parameters for each state on a tree (model=BMS), Ornstein-Uhlenbeck model with a single optimum for all species (model=OU1), Ornstein-Uhlenbeck model with different state means and a single α and σ^2 acting all selective regimes (model=OUM), and new Ornstein-Uhlenbeck models that assume different state means as well as either multiple σ^2 (model=OUMV), multiple α

(`model=OUMA`), or multiple α and σ^2 per selective regime (`model=OUMVA`). We also allow for several constrained models: a model that assumes different state means as well as multiple σ^2 but keeps α constant across all traits (`model=OUMVr`); a model that assumes different state means as well as either multiple α but keeps σ^2 constant across all traits (`model=OUMAr`). Finally, when `allfree=FALSE`, the model specified will be considered global and applied to all traits rather than estimating the same model separately for each trait.

Value

`OUwie.joint` returns an object of class `OUwie.joint`. This is a list with elements:

<code>\$loglik</code>	the maximum log-likelihood.
<code>\$AIC</code>	Akaike information criterion.
<code>\$AICc</code>	Akaike information criterion corrected for sample-size.
<code>\$model</code>	The model being fit
<code>\$param.count</code>	The number of parameters counted in the model.
<code>\$solution</code>	a matrix containing the maximum likelihood estimates of α and σ^2 .
<code>\$theta</code>	a matrix containing the maximum likelihood estimates of θ and its standard error.
<code>\$tot.state</code>	A vector of names for the different regimes
<code>\$index.mat</code>	The indices of the parameters being estimated are returned. The numbers correspond to the row in the <code>eigvect</code> and can be useful for identifying the parameters that are causing the objective function to be at a saddlepoint (see Details)
<code>\$simmap.tree</code>	A logical indicating whether the input phylogeny is a SIMMAP formatted tree.
<code>\$root.age</code>	The user-supplied age at the root of the tree.
<code>\$opts</code>	Internal settings of the likelihood search
<code>\$data</code>	User-supplied dataset
<code>\$phy</code>	User-supplied tree
<code>\$root.station</code>	A logical indicating whether the starting state, θ_0 , was estimated
<code>\$lb</code>	The lower bound set
<code>\$ub</code>	The upper bound set
<code>\$iterations</code>	Number of iterations of the likelihood search that were executed
<code>\$ntraits</code>	Number of traits optimized

Author(s)

Jeremy M. Beaulieu

References

- Leslie, A.B., Beaulieu, J.M., Crane, P.R., and Donoghue, M.J. 2014. Cone size is related to branching architecture in conifers. *New Phytologist* doi:10.1111/nph.12864.
- Beaulieu J.M., Jhwueng D.C., Boettiger C., and O'Meara B.C. 2012. Modeling stabilizing selection: Expanding the Ornstein-Uhlenbeck model of adaptive evolution. *Evolution* 66:2369-2383.

Examples

```
#data(tworegime)

#trait1 <- OUwie.sim(tree,trait, alpha=pp$solution[1,],
#sigma.sq=pp$solution[2,],theta=pp$theta[,1], theta0=pp$theta[1,1])
#trait2 <- OUwie.sim(tree,trait, alpha=pp$solution[1,],
#sigma.sq=pp$solution[2,],theta=pp$theta[,1], theta0=pp$theta[1,1])
#trait.multi <- cbind(trait[,1:2],trait1[,3],trait2[,3])

##Fit a global OUMV model applied to all traits:
#oumv.global <- OUwie.joint(tree,trait.multi, model="OUMV", ntraits=2, allfree=FALSE)

##Now fit a OUMV model applied to each separately, but optimized jointly:
#oumv.joint <- OUwie.joint(tree,trait.multi, model="OUMV", ntraits=2, allfree=TRUE)
```

OUwie.sim

*Generalized Hansen model simulator***Description**

Simulates the Ornstein-Uhlenbeck process of continuous characters evolving under discrete selective regimes.

Usage

```
OUwie.sim(phy=NULL, data=NULL, simmap.tree=FALSE, root.age=NULL, scaleHeight=FALSE,
alpha=NULL, sigma.sq=NULL, theta0=NULL, theta=NULL, mserr="none", fitted.object=NULL)
```

Arguments

phy	a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes
data	a dataframe containing species information (see Details). Not necessary to include if simmap=TRUE.
simmap.tree	a logical indicating whether the input tree is in SIMMAP format. The default is FALSE.
root.age	indicates the age of the tree. This is to be used in cases where the "tips" are not contemporary, such as in cases for fossil trees. Default is NULL meaning latest tip is modern day.
scaleHeight	a logical indicating whether the total tree height should be scaled to 1 (see Details). The default is FALSE.
alpha	a numeric vector giving the values of α for each selective regime (see Details)
sigma.sq	a numeric vector giving the values of σ^2 for each selective regime (see Details)
theta0	a numeric indicating the starting state, θ_0

theta	a numeric vector giving the values of θ for each selective regime (see Details)
mserr	designates whether a third column in the data matrix contains measurement error for each species value ("known"). The measurement error is assumed to be the standard error of the species mean. The default is "none".
fitted.object	a model fit from OUwie to use for simulation.

Details

The input is a tree and a data file OR a fitted OUwie object. The tree must be of class "phylo" and if `simmmap=FALSE` must contain the ancestral selective regimes as internal node labels. The data file is a dataframe that must have column entries in the following order: [,1] species names and [,2] their current selective regime. If `mserr="known"` then a third column can be added which contains the measurement error for each species. Note that if `simmmap=TRUE` no data file is needed. The user specifies the simulated parameter values (i.e. α , σ^2 , θ_0 , θ). Assuming two selective regimes, possible models can be specified as follows (Note that this assumes a stationary distribution at the root):

- Single rate Brownian motion (BM1): `alpha=c(1e-10,1e-10); sigma.sq=c(0.45,0.45); theta0=1.0; theta=c(0,0)`.
- Brownian motion with different rate parameters for each state on a tree (BMS): `alpha=c(1e-10,1e-10) sigma.sq=c(0.45,0.90); theta0=1.0; theta=c(0,0)`.
- Ornstein Uhlenbeck with a single optimum for all species (OU1): `alpha=c(0.1,0.1); sigma.sq=c(0.9,0.9); theta0=1; theta=c(1.0,1.0)`.
- Ornstein Uhlenbeck model that assumes different state means and a single α and σ^2 (OUM): `alpha=c(1.0,1.0); sigma.sq=c(0.45,0.45); theta0=1.0; theta=c(1.0,2.0)`.
- Ornstein Uhlenbeck model that assumes different state means and multiple σ^2 (OUMV): `alpha=c(1.0,1.0); sigma.sq=c(0.45,0.90); theta0=1.0; theta=c(1.0,2.0)`.
- Ornstein Uhlenbeck model that assumes different state means and multiple α (OUMA): `alpha=c(1.0,0.5); sigma.sq=c(0.45,0.45); theta0=1.0; theta=c(1.0,2.0)`.
- Ornstein Uhlenbeck model that assumes different state means and multiple σ^2 and α (OUMVA): `alpha=c(1.0,0.5); sigma.sq=c(0.45,0.9); theta0=1.0; theta=c(1.0,2.0)`.

With a fitted OUwie model, it just uses the parameters from that, ignoring any alpha, theta, etc. set in the function.

Value

A dataframe containing, as column entries, [,1] species names, [,2] current regime, [,3] simulated continuous trait, x.

Author(s)

Jeremy M. Beaulieu and Brian C. O'Meara

Examples

```

data(sim.ex)

#Simulate an Ornstein-Uhlenbeck model with different state means
#and a separate alpha and sigma^2 per selective regime
alpha=c(1.0,0.5)
sigma.sq=c(0.45,0.9)
theta0=1.0
theta=c(1.0,2.0)

sim.data<-OUwie.sim(tree,trait,simmap.tree=FALSE,scaleHeight=FALSE,
alpha=alpha,sigma.sq=sigma.sq,theta=theta0,theta=theta)

#Now fit a model to this and simulate from the fitted results
result <- OUwie(tree, sim.data, model="OUMVA", simmap.tree=FALSE,scaleHeight=FALSE)
sim.data.2 <- OUwie.sim(fitted.object=result)

```

OUwie.slice

Generalized Hansen models with time slices

Description

Fits generalized Ornstein-Uhlenbeck-based Hansen models of continuous characters before and after time slices.

Usage

```

OUwie.slice(phy, data, model=c("BMS","OUM","OUMV","OUMA","OUMVA"),
timeslices=c(NA), root.age=NULL, scaleHeight=FALSE, root.station=TRUE,
mserr="none", slice.lower.bound=NULL, starting.vals=NULL, diagn=FALSE,
quiet=FALSE, warn=TRUE)

```

Arguments

phy	a phylogenetic tree, in ape "phylo" format and with internal nodes labeled denoting the ancestral selective regimes.
data	a data.frame containing species information (see Details).
model	models to fit to comparative data (see Details).
timeslices	specifies the value and number fixed timeslices, timeslices to be estimated, or both (see Details).
root.age	indicates the age of the tree. This is to be used in cases where the "tips" are not contemporary, such as in cases for fossil trees. Default is NULL meaning latest tip is modern day.

scaleHeight	a logical indicating whether the total tree height should be scaled to 1 (see Details). The default is FALSE.
root.station	a logical indicating whether the starting state, θ_0 , should be estimated (see Details).
mserr	designates whether a fourth column in the data matrix contains measurement error for each species value ("known"). The measurement error is assumed to be the standard error of the species mean. The default is "none".
slice.lower.bound	specifies the value of the lower bound when estimating time slices. The default is NULL, meaning the value is $\exp(-21)$.
starting.vals	a vector of initial values for the optimization search. For OU models, two must be supplied, with the first being the initial alpha value and the second being the initial sigma squared. For BM models, just a single value is needed.
diagn	a logical indicating whether the full diagnostic analysis should be carried out. The default is FALSE.
quiet	a logical indicating whether progress should be written to the screen. The default is FALSE.
warn	a logical indicating whether a warning should be printed if the number of parameters exceeds <code>ntips/10</code> . The default is TRUE.

Details

This function fits various likelihood models for continuous characters evolving under discrete selective regimes that defined by a time slice (i.e., before and after K-P event). Timeslices can be fixed, estimated from the data, or some combination of both. Here it is assumed that the present is $T=0$ and the root is $\max(\text{branching.times}(\text{phy}))$. Also note that this function is still under development and so far it seems that when specifying an OU model there is a narrow range of conditions in which meaningful parameter estimates are obtained. Thus, use this function at your own risk. However, I would like to personally thank Graham Slater for vetting the function in great detail.

The trait data.frame must have column entries in the following order: [,1] species names, and [,2] the continuous trait of interest. Alternatively, if the user wants to incorporate measurement error (`mserr="known"`), then a third column, [,3], must be included that provides the standard error estimates for each species mean. However, a global measurement error for all taxa can be estimated from the data (`mserr="est"`); is not well tested, so use at your own risk.

Also note, when specifying the BMS model be mindful of the `root.station` flag. When `root.station=FALSE`, the non-censored model of O'Meara et al. 2006 is invoked (i.e., a single regime at the root is estimated), and when `root.station==TRUE` the group mean model of Thomas et al. 2006 (i.e., the number of means equals the number of regimes). The latter case appears to be a strange special case of OU, in that it behaves similarly to the OUMV model, but without selection. I would say that this is more consistent with the censored test of O'Meara et al. (2006), as opposed to having any real connection to OU. In any case, more work is clearly needed to understand the behavior of the group means model, and therefore, I recommend setting `root.station=FALSE` in the BMS case.

Value

`OUwie.slice` returns an object of class `OUwie.slice`. This is a list with elements:

\$loglik	the maximum log-likelihood.
\$AIC	Akaike information criterion.
\$AICc	Akaike information criterion corrected for sample-size.
\$model	The model being fit
\$param.count	The number of parameters counted in the model.
\$solution	a matrix containing the maximum likelihood estimates of α and σ^2 .
\$theta	a matrix containing the maximum likelihood estimates of θ and its standard error.
\$solution.se	a matrix containing the approximate standard errors of α and σ^2 . The standard error is calculated as the diagonal of the inverse of the Hessian matrix.
\$timeslices	a vector of timeslices either based on fixed age specified by the user, estimated from the data, or both.
\$tot.state	A vector of names for the different regimes
\$index.mat	The indices of the parameters being estimated are returned. The numbers correspond to the row in the eigvect and can be useful for identifying the parameters that are causing the objective function to be at a saddlepoint (see Details)
\$simmap.tree	A logical indicating whether the input phylogeny is a SIMMAP formatted tree.
\$root.age	The user-supplied age at the root of the tree.
\$opts	Internal settings of the likelihood search
\$data	User-supplied dataset
\$phy	User-supplied tree
\$root.station	A logical indicating whether the starting state, θ_0 , was estimated
\$starting.vals	A vector of user-supplied initial search parameters.
\$lb	The lower bound set
\$ub	The upper bound set
\$iterations	Number of iterations of the likelihood search that were executed
\$mserr.est	The estimated measurement error if mserr="est". Otherwise, the value is NULL.
\$res	A vector of residuals from the model fit. The residuals are ordered in the same way as the tips in the tree.
\$eigval	The eigenvalues from the decomposition of the Hessian of the likelihood function. If any eigval < 0 then one or more parameters were not optimized during the likelihood search (see Details)
\$eigvect	The eigenvectors from the decomposition of the Hessian of the likelihood function is returned (see Details)

Author(s)

Jeremy M. Beaulieu

References

- Beaulieu J.M., and O'Meara B.C. In prep.
- Beaulieu J.M., Jhwueng D.C., Boettiger C., and O'Meara B.C. 2012. Modeling stabilizing selection: Expanding the Ornstein-Uhlenbeck model of adaptive evolution. *Evolution* 66:2369-2383.
- O'Meara B.C., Ane C., Sanderson P.C., Wainwright P.C. 2006. Testing for different rates of continuous trait evolution using likelihood. *Evolution* 60:922-933.
- Butler M.A., King A.A. 2004. Phylogenetic comparative analysis: A modeling approach for adaptive evolution. *American Naturalist* 164:683-695.
- Thomas G.H., Freckleton R.P., and Szekely T. 2006. Comparative analysis of the influence of developmental mode on phenotypic diversification rates in shorebirds. *Proceedings of the Royal Society, B*. 273:1619-1624.

Examples

```
data(tworegime)

##Here we want a fixed slice at T=2, assuming the present is T=0:
#library(phytools)
#max.height <- max(nodeHeights(tree))
#timeslices <- max.height - 2
#timeslices <- c(0,timeslices)
#phy.sliced<-make.era.map(tree,timeslices)
#leg<-c("blue3","red3")
#names(leg)<-c(1,2)
#plotSimmap(phy.sliced,leg, pts=FALSE, ftype="off", lwd=1)

##Now fit an BMS model with a single fixed timeslice at time=2:
#ppBM<-OUwie.slice(tree,trait[,c(1,3)],model=c("BMS"), root.station=TRUE, timeslices=c(2))

##Fit an OU model with a single fixed timeslice:
#ppOUM<-OUwie.slice(tree,trait[,c(1,3)],model=c("OUM"), root.station=TRUE, timeslices=c(2))

##Fit an BMS model with an unknown timeslice:
#ppBM<-OUwie.slice(tree,trait[,c(1,3)],model=c("BMS"), root.station=TRUE, timeslices=c(NA))

##Fit an BMS model with an unknown and a fixed timeslice:
#ppBM<-OUwie.slice(tree,trait[,c(1,3)],model=c("BMS"), root.station=TRUE, timeslices=c(NA,2))
```

Index

*Topic **datasets**

Example, [2](#)

*Topic **models**

OUwie, [2](#)

OUwie.boot, [8](#)

OUwie.joint, [12](#)

OUwie.slice, [17](#)

Example, [2](#)

OUwie, [2](#)

OUwie.anc, [6](#)

OUwie.boot, [8](#)

OUwie.fixed, [10](#)

OUwie.joint, [12](#)

OUwie.sim, [15](#)

OUwie.slice, [17](#)

trait (Example), [2](#)

tree (Example), [2](#)