

Package ‘laser’

August 30, 2009

Version 2.3

Date 2009-08-29

Title Likelihood Analysis of Speciation/Extinction Rates from Phylogenies

Author Dan Rabosky

Maintainer Dan Rabosky <DLR32@cornell.edu>

Depends R(>= 2.6), ape, geiger

Suggests

Description laser implements maximum likelihood methods based on the birth-death process to test whether diversification rates have changed over time and whether rates vary among lineages.

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-08-30 14:38:19

R topics documented:

laser-package	2
agamids	3
bd	3
birthdeathSim	5
calcLHbd	6
DensityDependent	7
fitdAICrc	8
fitdAICrc.batch	10
fitNDR_1rate.Rd	11
fitNDR_2rate.Rd	13
fitSPVAR	16
gamStat	17
getBtimes	18
getBtimes.batch	19

getTipdata.Rd	21
MagallonSanderson.Rd	22
mccrTest.Rd	23
plethodon	25
plotLtt	25
plotNodeNumbers.phylo.Rd	26
plotRate	26
pureBirth	27
RandomTrees	28
scaleBranchingtimes	29
skinkdiversity	30
skinktree	30
truncateTree	31
warblers	32
yule-n-rate	33
yuleSim	35
yuleWindow	36

Index	38
--------------	-----------

laser-package	<i>LASER</i>
---------------	--------------

Description

Maximum likelihood-based methods for analyzing lineage diversification rates

Details

Package:	laser
Type:	Package
Version:	2.2
Date:	2008-09-21
License:	GPL 2.0 ?
LazyLoad:	yes

Author(s)

Dan Rabosky Maintainer: Dan Rabosky <DLR32@cornell.edu>

Examples

```
data(warblers)
fitSPVAR(warblers)
fitEXVAR(warblers)
gamStat(warblers)
```

```
mccrTest(CladeSize=50, NumberMissing=10, NumberOfReps=50, ObservedGamma=-2.1)
```

agamids

Ultrametric Phylogeny of Australian Agamid Lizards

Description

Ultrametric phylogenetic tree for 69 Australian Agamid lizards. Tree was constructed by maximum likelihood from 1800 bp of mtDNA using the GTR + I + G model of sequence evolution and made ultrametric using Non-Parametric Rate Smoothing.

Usage

```
data(agamids)
```

Source

Harmon, L. J., J. A. Schulte, A. Larson, and J. B. Losos. 2003. Tempo and mode of evolutionary radiation in Iguanian lizards. *Science* 301:961-964.

References

Rabosky, D. L. 2006. Likelihood methods for inferring temporal shifts in diversification rates. *Evolution* 60:1152-1164.

Examples

```
data(agamids)
agbtimes <- getBtimes(string = agamids)
gamStat(agbtimes)
plotLtt(agbtimes)
```

bd

Fit Rate-Constant Birth-Death Model to Branching Times

Description

Finds maximum likelihood estimates of the net diversification rate r (speciation rate S minus the extinction rate E) and the extinction fraction $a = E/S$, using branching times derived from an ultrametric phylogenetic tree.

Usage

```
bd(x, ai = c(0.1, 0.5, 0.9))
```

Arguments

<code>x</code>	a numeric vector of branching times
<code>ai</code>	a vector of initial a parameterizations for the optimization algorithm

Details

Non-linear optimization can be exceedingly difficult, and the algorithms used here can become trapped on local (rather than global) optima. The default 'ai' parameters specified above fit the constant-rate birth-death model to branching times using three initial a values. You should check your results against those obtained using the [pureBirth](#) model. If the log-likelihood under 'bd' is less than 'pureBirth', you should explore alternative initial parameterizations. For example, '`ai = seq(0.05, 0.99, length.out = 20)`' would attempt the optimization with 20 equally spaced a values on the interval (0.05, 0.99).

I have found the default option to be satisfactory for all phylogenies I have examined.

Value

a list with the following components:

<code>LH</code>	the log-likelihood at the maximum
<code>aic</code>	the Akaike Information Criterion
<code>r</code>	the net diversification rate giving the maximum log-likelihood
<code>a</code>	the extinction fraction giving the maximum log-likelihood

Author(s)

Dan Rabosky (DLR32@cornell.edu)

References

- Kendall, D. G. 1948. On the generalized "birth-and-death" process. *Ann. Math. Stat.* 19:1-15.
- Nee, S., E. C. Holmes, R. M. May, and P. H. Harvey. 1994a. Extinction rates can be estimated from molecular phylogenies. *Philos. Trans. R. Soc. Lond. B* 344:77-82.
- Nee, S., R. M. May, and P. H. Harvey. 1994b. The reconstructed evolutionary process. *Philos. Trans. R. Soc. Lond. B* 344:305-311.

See Also

[pureBirth](#), [fitdAICrc](#), [yule-n-rate](#)

Examples

```
data(plethodon)
result <- bd(plethodon)
```

birthdeathSim *birthdeathSim*

Description

Simulate Branching Times Under the birth-death model and/or with incomplete sampling

Usage

```
birthdeathSim(b, d, CladeSize, NumberMissing, NumberOfReps)
```

Arguments

`b` the speciation rate
`d` the extinction rate
`CladeSize` the true clade size you wish to simulate
`NumberMissing` the number of taxa missing from the 'real' tree
`NumberOfReps` the number of phylogenies to simulated

Details

This generates a matrix of branching times that can be used with `fitdAICrc.batch` under any parameterization of a general birth-death model with or without incomplete sampling.

To simulate a clade with incomplete sampling, note that `CladeSize` is the true size of a phylogeny, and `NumberMissing` is the number of missing taxa, so if you specified `CladeSize = 100` and `NumberMissing = 20`, your resulting trees would have 80 tips (Trees with 100 tips would be simulated, but then 20 taxa would be dropped randomly to simulate incomplete sampling).

This function is basically a wrapper for the `birthdeath.tree` function from Geiger, but makes it more amenable to calculation of the delta AIC test statistic for detecting temporal changes in diversification rates.

Value

an $n \times m$ matrix of branching times, where n is size `CladeSize - NumberMissing` and m is size `NumberOfReps`.

Author(s)

Dan Rabosky (DLR32@cornell.edu)

`calcLHbd`*Calculate Likelihood of Branching Times Under Birth-Death Model*

Description

Calculates the likelihood of branching times given a vector of branching times and parameters r (net diversification rate; speciation rate S - extinction rate E) and a (the extinction fraction, E/S)

Usage

```
calcLHbd(x, r, a)
```

Arguments

<code>x</code>	a numeric vector of branching times
<code>r</code>	the net diversification rate, $S - E$
<code>a</code>	the extinction fraction, E/S

Details

A function that can be called to explore alternative parameterizations of the birth-death process, to bootstrap likelihood confidence regions, or to generate plots of likelihood surfaces (as in Nee et al. 1994).

Value

the likelihood of the branching times given a and r

Author(s)

Dan Rabosky (DLR32@cornell.edu)

References

Kendall, D. G. 1948. On the generalized "birth-and-death" process. *Ann. Math. Stat.* 19:1-15.

Nee, S., E. C. Holmes, R. M. May, and P. H. Harvey. 1994a. Extinction rates can be estimated from molecular phylogenies. *Philos. Trans. R. Soc. Lond. B* 344:77-82.

Nee, S., R. M. May, and P. H. Harvey. 1994b. The reconstructed evolutionary process. *Philos. Trans. R. Soc. Lond. B* 344:305-311.

See Also

`bd`, `pureBirth`

Examples

```

#plot a likelihood surface for the plethodon dataset
data("plethodon")
rvector <- seq(.001, .04, length.out = 100)
avector <- seq(0, 0.99, length.out = 100)
#calculating likelihoods:
lmatrix <- matrix(0, length(rvector), length(avector))
for (i in 1:length(rvector)){
  for (j in 1:length(avector)){
    lmatrix[i,j] <- calcLHbd(plethodon, rvector[i], avector[j])
  }
}

#Now to plot surface:
lmax <- max(lmatrix) #maximum calculated LH
filled.contour(rvector, avector, lmatrix,
              levels = seq(lmax-5, lmax, length.out = 20),
              col = heat.colors(20), xlab = "Net Diversification Rate",
              ylab = "Extinction Fraction", key.title = title("Log-LH"))

# plots the surface. See ?filled.contour for help on this plotting function.

```

DensityDependent *Fit Density Dependent Speciation Model to Branching Times*

Description

Functions that fit density-dependent speciation rate models to branching times derived from phylogenetic data. 'DDX' and 'DDL' fit exponential and logistic variants of the density-dependent speciation rate model.

Usage

```
DDX(x)
DDL(x)
```

Arguments

`x` a numeric vector of branching times

Details

'DDX' models the speciation rate as a function of the number of extant lineages at any point in time, $r(t) = r_0 * (Nt \text{ \textasciicircum (-x)})$, where r_0 is the initial speciation rate, Nt is the number of lineages at some time t , and x is a parameter controlling the magnitude of the rate change.

'DDL' models the speciation rate as $r(t) = r_0 * (1 - Nt / K)$, where r_0 is the initial speciation rate, Nt is the number of extant lineages at some time t , and K is analogous to the 'carrying capacity' parameter of population ecology.

Value

a list with the following components:

LH	The log-likelihood at the maximum
aic	the Akaike Information Criterion
r1	the initial speciation rate
kparam	the K parameter in the logistic density dependent model
xparam	the x parameter in the density-dependent exponential model

Author(s)

Dan Rabosky (DLR32@cornell.edu)

References

- Nee, S., R. M. May, and P. H. Harvey. 1994b. The reconstructed evolutionary process. *Philos. Trans. R. Soc. Lond. B* 344:305-311.
- Nee, S., A. Mooers, and P. H. Harvey. 1992. Tempo and mode of evolution revealed from molecular phylogenies. *Proc. Natl. Acad. Sci. USA* 89:8322-8326.

See Also

[yule-n-rate](#), [fitdAICrc](#)

Examples

```
data(plethodon)
resX <- DDX(plethodon)
resL <- DDL(plethodon)
```

fitdAICrc

Test for Rate Variation Using delta-AICrc Test Statistic

Description

Fits a specified set of rate-variable and rate-constant variants of the birth-death model to branching times from phylogenetic data. The test statistic dAICrc is the difference in AIC scores between the best rate-constant and rate-variable models.

Usage

```
fitdAICrc(x, modelset = c("pureBirth", "bd", "DDL", "DDX", "yule2rate"), ints = NU
```

Arguments

<code>x</code>	a numeric vector of branching times
<code>modelset</code>	the set of rate-constant and rate-variable candidate models to be fitted
<code>ints</code>	the number of intervals. See 'Details'

Details

'fitdAICrc' implements the dAICrc test statistic for temporal variation in diversification rates as described in Rabosky (2006).

`modelset` is a list of the rate-constant and rate-variable models to consider. You should include both rate-constant models ([pureBirth](#) and [bd](#)), as well as one or more candidate rate-variable models. Available options are [DDX](#), [DDL](#), [yule2rate](#), and [yule3rate](#). See full descriptions of each of these models this document.

'ints' is used in determining the number of shift points to consider. If 'ints = NULL' (the default), the model will consider only observed branching times as possible shift points. See [yule-n-rate](#) for additional discussion of the 'ints' option.]

Note that the rvbd function is no longer supported ('rate variable birth death').

Value

a dataframe with the number of rows equal to the number of candidate models. Columns include likelihoods, parameters, and AIC scores for each model. The first column contains the model names. If a parameter is not present in a particular model, it will have an entry of 'NA' in the column for that parameter. Parameter names follow conventions for model descriptions in other parts of this document. For example, parameter 'r1' is the initial net diversification rate for all models (note that this will be the only rate for the 'pureBirth' model).

The full set of columns if all available models are included in the candidate set will consist of the following:

<code>model</code>	the model name for row <i>i</i> in the dataframe
<code>params</code>	the free parameters for model[i]
<code>np</code>	the number of free parameters in mode[i]
<code>mtype</code>	either 'RC' for rate-constant or 'RV' for rate-variable
<code>LH</code>	the log-likelihood under model[i]
<code>r1, r2, r3</code>	net diversification rates, as applicable; r1 is always the initial rate, and r3 is always the final rate
<code>a</code>	the extinction fraction <i>E/S</i> if applicable
<code>xp</code>	the x-parameter from the DDX model
<code>k</code>	the k-parameter from the DDL model
<code>st1, st2</code>	shift-times, if applicable. st1 is always the first shift point
<code>AIC</code>	the Akaike Information Criterion for model[i]
<code>dAIC</code>	delta-AIC; the difference in AIC scores between model[i] and the overall best-fit model

Author(s)

Dan Rabosky <DLR32@cornell.edu>

References

Nee, S., R. M. May, and P. H. Harvey. 1994b. The reconstructed evolutionary process. *Philos. Trans. R. Soc. Lond. B* 344:305-311.

Rabosky, D. L. 2006. Likelihood methods for inferring temporal shifts in diversification rates. *Evolution* 60:1152-1164.

Examples

```
data(agamids)
agbtimes <- getBtimes(string = agamids)
#agbtimes is now a vector of branching times from the agamid phylogeny

#here we fit 2 rate-constant and 3 rate-variable models
# to the agamid data:
result <- fitdAICrc(agbtimes, modelset = c("pureBirth", "bd",
      "DDX", "DDL", "yule2rate"), ints = 100)

# this outputs summaries of parameters and likelihoods to screen;
# object 'result' is a dataframe containing all parameter estimates,
# likelihoods, and AIC scores
```

fitdAICrc.batch *Generate Null Distribution of dAICrc*

Description

Generates null distribution of dAICrc test statistic and determines significance of dAICrc indices calculated for 'real' phylogenies.

Usage

```
fitdAICrc.batch(x, observed=NULL, ...)
```

Arguments

x	a numeric matrix or dataframe of branching times
observed	the dAICrc statistic for the 'real' phylogeny
...	other parameters, e.g., ints for yule2rate

Details

For details on input format, see `getBtimes.batch`.

This fits the following models: pure birth, birth death, density-dependent (exponential), density-dependent (logistic), yule-2-rate

'observed' is the calculated dAICrc statistic obtained for a test phylogeny for which you would like to obtain a p-value (using `fitdAICrc`).

Value

A dataframe with N rows, where N is the number of sets of branching times analyzed. Columns of the data frame are the AIC scores for each of the models. The final column, with name dAIC, is the deltaAICrc test statistic for that particular tree.

Note

Make sure that you are using the exact same set of models for the real phylogeny and for the simulated phylogenies. If not, your results are invalid.

Author(s)

Dan Rabosky <DLR32@cornell.edu>

References

Rabosky, D. L. 2006. Likelihood methods for inferring temporal shifts in diversification rates. *Evolution* 60:1152-1164.

fitNDR_1rate.Rd *Estimate diversification rate from combined taxonomic/phylogenetic data*

Description

Find maximum likelihood estimate of net diversification rate from combined taxonomic/phylogenetic data.

Usage

```
fitNDR_1rate(phy, eps=0, rbounds=c(0.0001, .5), combined=TRUE)
```

Arguments

phy	a class 'phylo' phylogenetic tree with an additional component 'phenotype'; phy\$phenotype specifies the number of tips per terminal taxon. The phenotype data is added with getTipdata
eps	the relative extinction rate, or μ / λ
rbounds	The upper and lower limits for the 1-dimensional optimization of the net diversification rate
combined	Should likelihoods be calculated using both combined phylogenetic and taxonomic data, or taxonomic data only?

Details

fitNDR_1rate finds the maximum likelihood estimate of the net diversification rate using phylogenetic and taxonomic data. This method is best applied to phylogenetic trees for higher taxonomic levels where you have incomplete sampling but know the approximate species diversities of each terminal taxon in the tree (e.g., a phylogenetic tree of arthropod families).

eps allows you to estimate the speciation rate under any assumed (constant) relative extinction rate, where the relative extinction rate is μ / λ .

rbounds is an important argument, and the default may not work. Basically, you are trying to find the ML estimate of the net diversification rate; however, the optimization algorithm requires that you specify min and max values for the search-space. Sometimes the function cannot be evaluated at these bounds or the optimization will fail for other reasons. Thus, you might wish to start with a vary small range (e.g., rbounds = c(0.001, 0.05)) and increase the range until you are confident of finding the optimum. Most important: if the maximum diversification rate appears to be on a boundary, you have a problem. For example, if you specify rbounds = c(0.001, 0.05), and find the r value returned by the function to be 0.05 at the maximum, then this is almost certainly not valid. You will need to repeat your analysis, expanding the range as necessary, until you find a maximum that does not lie on a boundary. Also be wary of multiple optima, though I haven't personally encountered them.

The combined argument asks whether you wish to use an estimator based on (i) combined taxonomic and phylogenetic data, or (ii) taxonomic data only. There are good reasons for trying both approaches: when combined = TRUE, both the taxonomic/species richness and phylogenetic backbone of a tree contribute heavily to the overall likelihood estimate. This occurs in spite of the fact that most of the species diversity is actually in the tips. When combined = FALSE, the internal phylogenetic structure of the tree does not contribute to the likelihood; you are finding the maximum likelihood estimate of the speciation rate λ from the species richness data plus stem clade ages alone. I recommend checking estimates under both combined = TRUE and combined = FALSE

Value

a dataframe with the following components:

LH	The log-likelihood at the maximum
aic	the Akaike Information Criterion
r	the net diversification rate, $\lambda - \mu$, at the maximum

lambda the ML estimate of the speciation rate
 eps the value of eps used

Author(s)

Dan Rabosky <DLR32@cornell.edu>

References

Rabosky, D. L., S. C. Donnellan, A. L. Talaba, and I. J. Lovette. 2007. Exceptional among-lineage variation in diversification rates during the radiation of Australia's largest vertebrate clade. *Proc. Roy. Soc. Lond. Ser. B* 274:2915-2923.

See Also

[fitNDR_2rate](#), [getTipdata](#), [lambda.stem.ml](#)

Examples

```
data(skinktree);
data(skinkdiversity);
skinktree <- getTipdata(skinkdiversity, skinktree);

#first we fit the one rate model
fitNDR_1rate(skinktree, eps=0);

#here we fit the 2 rate model
res <- fitNDR_2rate(skinktree, eps=0);

#extracting the node most likely to have undergone rate shift:
subset(res, res$LH==max(res$LH));

#this function plots the node numbers on the tree:
plotNodeNumbers.phylo(skinktree);

# and you can see that the 'best' rate shift
# location is the MRCA of the genera Ctenotus and Lerista
```

fitNDR_2rate.Rd *Fit 2-rate diversification model to combined phylogenetic/taxonomic data*

Description

Finds likelihoods and parameter estimates for combined phylogenetic/taxonomic data under a two-rate diversification model.

Usage

```
fitNDR_2rate(phy, eps=0, combined=TRUE, rate.decrease=FALSE, rbounds=c(.001, .5))
```

Arguments

<code>phy</code>	a class 'phylo' phylogenetic tree with an additional component 'phenotype'; <code>phy\$phenotype</code> specifies the number of tips per terminal taxon. The phenotype data is added with <code>getTipData</code>
<code>eps</code>	the relative extinction rate, or μ / λ
<code>combined</code>	Should likelihoods be calculated using both combined phylogenetic and taxonomic data, or taxonomic data only?
<code>rate.decrease</code>	Constrain the analysis such that the highest diversification rate occurs in the tree bipartition containing the root node?
<code>rbounds</code>	The upper and lower limits for the 1-dimensional optimization of the net diversification rate. See details.

Details

`fitNDR_2rate` fits a 2-speciation rate model to combined phylogenetic/taxonomic data. Suppose you have a higher-level phylogeny for some group of organisms (e.g., beetle families), where you also know the approximate species diversities for each terminal. `fitNDR_2rate` assumes that at some point in the tree, an ancestral speciation rate `lambda1` shifts to a new speciation rate `lambda2`.

The model is fitted by iteratively splitting the tree at each node and fitting a birth-death model to each of the resulting bipartitions. Thus, for each node, you obtain (i) the likelihood of a rate shift at that position in the tree, and (ii) the estimated speciation rates for each bipartition. The function returns a dataframe giving the likelihoods of rate shift at each node as well as the parameter estimates.

`eps` allows you to estimate the speciation rate under any assumed (constant) relative extinction rate, where the relative extinction rate is μ / λ .

`rbounds` is an important argument, and the default may not work. Basically, you are trying to find the ML estimate of the net diversification rate; however, the optimization algorithm requires that you specify min and max values for the search-space. Sometimes the function cannot be evaluated at these bounds or the optimization will fail for other reasons. Thus, you might wish to start with a very small range (e.g., `rbounds = c(0.001, 0.05)`) and increase the range until you are confident of finding the optimum. Most important: if the maximum diversification rate appears to be on a boundary, you have a problem. For example, if you specify `rbounds = c(0.001, 0.05)`, and find the `r` value returned by the function to be 0.05 at the maximum, then this is almost certainly not valid. You will need to repeat your analysis, expanding the range as necessary, until you find a maximum that does not lie on a boundary. Also be wary of multiple optima, though I haven't personally encountered them.

Specifying `combined = TRUE` estimates `lambda` from combined taxonomic and phylogenetic data. Thus, the likelihood is a function of the internal branching structure of the tree and the species richness/taxonomic data. If `combined=FALSE`, likelihoods are estimated from the species richness/taxonomic data only, and the internal branching structure of the tree does not contribute.

Option `rate.decrease` fits a model where the highest speciation rate must occur in the tree bipartition containing the root node.

Value

a dataframe with the following components:

<code>node</code>	The node defining the tree bipartition
<code>LH</code>	The log-likelihood at the maximum
<code>aic</code>	the Akaike Information Criterion
<code>r.1</code>	the net diversification rate, $\lambda - \mu$, for the partition containing the root node
<code>lambda.1</code>	the speciation rate for the for the partition containing the root node
<code>LH.1</code>	the log-likelihood for partition 1 at the maximum
<code>r.2</code>	the net diversification rate, $\lambda - \mu$, for the partition NOT containing the root node, e.g., partition 2
<code>lambda.2</code>	the speciation rate for the for the partition 2
<code>LH.2</code>	the log-likelihood for partition 2 at the maximum
<code>eps</code>	the value of <code>eps</code> used

Author(s)

Dan Rabosky (DLR32@cornell.edu)

References

Rabosky, D. L., S. C. Donnellan, A. L. Talaba, and I. J. Lovette. 2007. Exceptional among-lineage variation in diversification rates during the radiation of Australia's largest vertebrate clade. *Proc. Roy. Soc. Lond. Ser. B* 274:2915-2923.

See Also

[fitNDR_1rate](#), [getTipdata](#), [lambda.stem.ml](#)

Examples

```
data(skinktree);
data(skinkdiversity);
skinktree <- getTipdata(skinkdiversity, skinktree);

#first we fit the one rate model
fitNDR_1rate(skinktree, eps=0);

#here we fit the 2 rate model
res <- fitNDR_2rate(skinktree, eps=0);

#extracting the node most likely to have undergone rate shift:
subset(res, res$LH==max(res$LH));
```

```
#this function plots the node numbers on the tree:
plotNodeNumbers.phylo(skinktree);
```

fitSPVAR	<i>Fit model with continuous-time varying speciation/extinction rates to phylogeny</i>
----------	--

Description

Functions that fit time-varying speciation and extinction models to branching times derived from phylogenetic data. ‘fitSPVAR’ fits a model with an exponentially declining speciation rate through time and constant extinction. ‘fitEXVAR’ fits a model with exponentially increasing extinction and constant speciation. ‘fitBOTHVAR’ fits a model where both speciation and extinction rates can vary through time.

Usage

```
fitSPVAR(bt, init=c(2, .2, .1))
fitEXVAR(bt, init=c(.3, .01, 1))
fitBOTHVAR(bt, init=c(.3, .5, .1, .5))
```

Arguments

bt	a numeric vector of branching times
init	a vector of starting parameter values for the optimization algorithm. Note that there are 3 parameters in SPVAR and EXVAR models, but 4 parameters in BOTHVAR

Details

These functions are described in Rabosky & Lovette (2008, *Evolution*). There is a high likelihood that the optimization will fail across a large span of parameter space, so you should not expect the default parameters to work. I recommend varying the first parameter (e.g., `init[1]`) before changing anything else.

The three models return the log-likelihood, the AIC, and the parameter estimates. Parameters correspond exactly to those described in Rabosky and Lovette (2008), equations 7-11. The speciation rate is specified by parameters `lam0` and `k`, and extinction through time is described by `mu0` and `z`. `lam0` and `mu0` are the initial speciation and final extinction rates, respectively. `k` and `z` control the rate of decrease / increase in speciation and extinction, respectively.

Because the underlying mathematical model of speciation and extinction requires that the extinction rate not exceed the speciation rate, the models have been reparameterized as follows: for the SPVAR model, optimization is performed on the parameter set `c(X, k, mu0)`, where `k` and `mu0` are defined as above and `X` corresponds the net diversification rate. This provides for a fixed minimum bound for the optimization interval (because `X` must always be greater than 0). This should be irrelevant to the usage of the function. However, if you are having a difficult time getting optimization to work

without an error, bear in mind that the initial parameters you are supplying correspond to: SPVAR: X, k, mu0; EXVAR: X, mu0, z; and BOTHVAR, X, k, mu0, z. A good solution might be to start with initial parameter values under the pure birth or constant rate birth death model. You will need to estimate these in light of equations 7-11 in Rabosky & Lovette (2008).

Optimization uses the L-BFGS-B algorithm as implemented in the function `optimize`. Numerical integration uses `integrate`.

You can use the equations from Rabosky & Lovette (2008) with your parameter estimates to generate a plot of the speciation and extinction through time curves, or you can use the function [plotRate](#)

Value

<code>model</code>	The name of the model
<code>LH</code>	Maximum log-likelihood of the fitted model
<code>aic</code>	AIC
<code>lam0</code>	The initial speciation rate
<code>k</code>	Parameter of the exponential change in speciation rate
<code>mu0</code>	the final extinction rate
<code>z</code>	Parameter of the exponential change in extinction rate

Author(s)

Dan Rabosky (DLR32@cornell.edu)

References

Rabosky, D. L. and I. J. Lovette. 2008. Explosive evolutionary radiations: decreasing speciation or increasing extinction through time? *Evolution*, doi:10.1111/j.1558-5646.2008.00409.x

Examples

```
data(warblers)
fitSPVAR(warblers)
fitEXVAR(warblers)
fitBOTHVAR(warblers)
```

gamStat

gamStat

Description

Computes the Gamma Statistic of Pybus and Harvey (2000), using branching times

Usage

```
gamStat(x, return.list=TRUE)
```

Arguments

`x` a numeric vector of branching times
`return.list` logical: return a list with pvalue and the statistic?

Value

If `return.list = TRUE`, a list containing the following elements:

`gamstat` the calculated gamma statistic
`pval` One-tailed p-value

If `return.list = FALSE`, it simply returns a single number, the gamma statistic

Author(s)

Dan Rabosky (DLR32@cornell.edu)

References

Pybus, O. G., Harvey, P. H. 2000. New uses for incomplete phylogenies.

Examples

```
data("plethodon")
pleth <- plethodon
gamStat(pleth)
```

getBtimes

Get Branching Times From Parenthetic-Format Tree ('Newick')

Description

Reads a file containing an ultrametric tree in parenthetic format and returns a numeric vector of branching times, sorted from earliest to most recent.

Usage

```
getBtimes(file = NULL, string=NULL)
```

Arguments

`file` a file containing a single tree in 'newick' (parenthetic) format
`string` the name of an object containing a 'newick' tree character string

Details

This function requires the package ‘ape’. If ‘ape’ has not been attached to the search path but exists in the R directory, it will automatically be loaded. If ‘ape’ does not exist, you must install it by typing ‘install.packages("ape")’ at the R prompt.

If tree is not ultrametric, an error message will be generated.

Value

a numeric vector of branching times

Author(s)

Dan Rabosky <DLR32@cornell.edu>

See Also

[getBtimes.batch](#)

Examples

```
data(agamids)
write.table(agamids, file = 'example.tre', quote=FALSE,
            row.names = FALSE, col.names = FALSE)
# the preceding lines generate a file 'example.tre' that can be
# read by 'getBtimes'

agbtimes <- getBtimes(file = 'example.tre')
#or alternatively:
agbtimes <- getBtimes(string = agamids)
#agbtimes is now a numeric vector of branching times

plotLtt(agbtimes) #plot log-lineages through time for agamid data
gamStat(agbtimes) #calculate gamma statistic for agamid data
unlink('example.tre') #clean-up.
```

getBtimes.batch *Branching Times for Batch of Phylogenies*

Description

Processes files containing multiple sets of phylogenetic trees in parenthetic (‘Newick’) format and returns a matrix of branching times to generate distributions of diversification rate test statistics.

Usage

```
getBtimes.batch(fname, basal = NULL)
```

Arguments

`fname` filename where the parenthetic trees are stored
`basal` scales all trees to same basal divergence time. See 'details'.

Details

'basal' will scale all of your trees to the same basal divergence time. This situation could arise in practice if you are interested in the posterior distributions of diversification rate parameters estimated under one or more models in this package. For example, you might have the output file from a run of MrBayes (the *.t* file), with trees generated under a clock constraint. Suppose you wished to examine the posterior distribution of speciation and/or extinction rates under a constant rate birth death model. Since all of the trees in the posterior distribution should be calibrated to the same basal divergence, you may specify '`basal = value`', where value is the inferred time of the basal divergence.

Value

A matrix of branching times, where rows are different trees or datasets, and columns are branching times. Thus, if you have N trees and K taxa, you will have a matrix of N rows and K-1 columns, since the number of branching times is one less than the number of taxa in a phylogenetic tree.

Suppose you have a file 'trees.tre', in newick format. '`Res <-getBtimes.batch(fname = 'trees.tre')`' returns the matrix of branching times. You can access the j'th tree as '`Res[j,]`'. Thus, '`plotLtt(Res[5,])`' would generate a log-lineages through time plot for the 5th tree in the file.

Note

To analyze a single tree, see [getBtimes](#)

Author(s)

Dan Rabosky <DLR32@cornell.edu>

See Also

[getBtimes](#)

Examples

```
data("rtrees50")
write.table(rtrees50, file = 'temp.txt', quote=FALSE,
           row.names = FALSE, col.names = FALSE)

#creates a temporary file with trees in Newick format, identical to
# output from PHYLOGEN & other software

btimes <- getBtimes.batch(fname = "temp.txt")

# now btimes is a matrix of branching times. Rows are different trees;
```

```

# columns are branching times.
# To verify that this has correctly read the tree, we can plot the log-
# lineages through time for the first tree:

plotLtt(btrees[1,])

# And we can compute the gamma statistic for this set of branching times:
gamStat(btrees[1,])

# or if you wanted to compute the gamma statistic for each tree:
result <- as.numeric(apply(btrees, MARGIN=1, gamStat, return.list=FALSE))
hist(result) #plot histogram of gamma stat values for trees

unlink("temp.txt") #clean up; delete temp file.

```

getTipdata.Rd

Add species richness data to tips of phylogenetic tree

Description

Adds species richness (taxonomic) data to phylogenetic tree

Usage

```
getTipdata(tipdata, phy)
```

Arguments

tipdata	either of the following: a vector of species diversities, where the name of each element corresponds to the name of a terminal in a phylogenetic tree, or a data frame with a single column, where the row names are the names of the terminals
phy	a class 'phylo' phylogenetic tree, perhaps obtained by <code>read.tree</code>

Details

`getTipdata` takes species richness data and adds a new component `phenotype` to your phylogenetic tree. The phenotype data could in principle be anything, but the function is designed to add species richness data to the tree to facilitate analysis by `fitLambda_1rate` and `fitNDR_2rate`.

This function requires that the names in `tipdata` exactly match `phy$tip.label`; if they don't, you will (probably) receive an error message. If you fail to supply a vector with names or a dataframe with `row.names`, the input vector will be matched exactly to the tips in the order in which they are stored in your tree (e.g., `tipdata[1]` will correspond to `phy$tip.label[1]`). This is not recommended.

Value

a phylogenetic tree with the standard components of class 'phylo' plus the following component:

phenotype Species diversities associated with each terminal taxon

Author(s)

Dan Rabosky (DLR32@cornell.edu)

References

Rabosky, D. L., S. C. Donnellan, A. L. Talaba, and I. J. Lovette. 2007. Exceptional among-lineage variation in diversification rates during the radiation of Australia's largest vertebrate clade. *Proc. Roy. Soc. Lond. Ser. B* 274:2915-2923.

See Also

[fitNDR_2rate](#), [getTipdata](#), [lambda.stem.ml](#), [fitNDR_1rate](#)

MagallonSanderson.Rd

Estimate speciation rates for crown and stem groups

Description

Estimate speciation rates under any relative extinction rate, given a crown or stem group age and a number of extant species

Usage

```
lambda.crown.ms01(n, tb, eps=0)
lambda.stem.ms01(n, tb, eps=0)
lambda.stem.ml(n, tb, eps=0)
lambda.stem.ci(tb, r, eps=0)
```

Arguments

n	the number of extant species in a clade
tb	the stem or crown group age
eps	the relative extinction rate, or μ / λ
r	the net diversification rate, r (<code>lambda.stem.ci</code> only)

Details

`lambda.crown.ms01` estimates speciation rates assuming some value of `eps` and a crown group age after Magallon & Sanderson (2001), eqn A7.

`lambda.stem.ms01` estimates speciation rates assuming a known stem clade age. Same as eqn A6 in Magallon & Sanderson 2001.

`lambda.stem.ml` finds maximum likelihood estimate of speciation rate given stem age and `eps`. This is given in Magallon & Sanderson (2001), eqns 1-2, Raup (1985) eqn A18, and others.

`lambda.stem.ci` gives 95 percent confidence intervals on expected species diversity after `tb` time units given net diversification rate `r` and `eps`

Value

a list with the following components (for speciation rate estimators)

<code>lambda</code>	the estimated speciation rate
<code>r</code>	the net diversification rate
<code>upper</code>	upper bound of 95 percent confidence interval on expected species diversity
<code>lower</code>	lower bound of 95 percent confidence interval on expected diversity

Author(s)

Dan Rabosky (DLR32@cornell.edu)

References

Magallon, S., and M. J. Sanderson. 2001. Absolute diversification rates in angiosperm clades. *Evolution* 55:1762-1780.

Raup, D. M. 1985. Mathematical models of cladogenesis. *Paleobiology* 11:42-52.

See Also

[fitNDR_2rate](#), [getTipdata](#), [lambda.stem.ml](#), [fitNDR_1rate](#)

mccrTest.Rd

Conduct monte carlo constant rates (gamma statistic) test

Description

Conducts gamma statistic analysis (Pybus and Harvey 2000) for incompletely sampled phylogenies. Phylogenies are simulated to the full clade size under the null hypothesis (constant rate pure birth diversification process) and taxa are randomly pruned from the tree to mimic incomplete sampling. The null distribution of the gamma statistic is then tabulated from these phylogenies.

Usage

```
mccrTest(CladeSize, NumberMissing, NumberOfReps, ObservedGamma = NULL, fname=NULL)
```

Arguments

CladeSize	The TRUE clade diversity
NumberMissing	The number of missing species, e.g., CladeSize minus the number of taxa in your tree
NumberOfReps	The number of Monte Carlo simulations to conduct. Recommend ≥ 5000
ObservedGamma	The observed gamma statistic value for the empirical tree. Optional.
fname	An optional filename where simulated trees are stored, if you generated the trees in another program, such as phylogen

Details

If ObservedGamma is supplied, `mccrTest` returns the p-value.

Value

A list with the following components:

<code>null.gamma</code>	The null distribution of gamma. You can plot a histogram or otherwise inspect these values...
<code>critical.value</code>	The 0.05 percentile of the null distribution. This is the value corresponding to $\alpha = 0.05$
<code>p.value</code>	The actual p-value, only returned if ObservedGamma is supplied by user

Author(s)

Dan Rabosky <DLR32@cornell.edu>

References

Pybus, O. G. and Harvey, P. H. 2000. Testing macro-evolutionary models using incomplete molecular phylogenies. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 267, 2267-2272.

See Also

[gamStat](#)

Examples

```
mccrTest(CladeSize=25, NumberMissing=5, NumberOfReps=50);
```

`plethodon`*plethodon*

Description

Branching times for plethodontid salamanders from Highton and Larson (1979) and analyzed by Nee et al. 1994 and Nee 2001

Usage

```
data(plethodon)
```

Format

The data are represented by a numeric vector of branching times

Source

Highton, R., and A. Larson. 1979. The genetic relationships of the salamanders of the genus *Plethodon*. *Syst. Zool.* 28:579 - 599.

References

Nee, S., E. C. Holmes, R. M. May, and P. H. Harvey. 1994a. Extinction rates can be estimated from molecular phylogenies. *Philos. Trans. R. Soc. Lond. B* 344:77-82.

Nee, S. 2001. Inferring speciation rates from phylogenies. *Evolution* 55:661-668.

Examples

```
data(plethodon)
plotLtt(plethodon)
```

`plotLtt`*plotLtt*

Description

Plots log-lineages through time given a vector of branching times.

Usage

```
plotLtt(x)
```

Arguments

`x` a numeric vector of branching times

Value

a plotting function with no return value

Author(s)

Dan Rabosky <DLR32@cornell.edu>

Examples

```
data(warblers)
plotLtt(warblers);
```

```
plotNodeNumbers.phylo.Rd
```

Plot phylogenetic tree with node numbers

Description

Wrapper for plot.phylo; plots class 'phylo' trees and displays internal and terminal node numbers

Usage

```
plotNodeNumbers.phylo(phy)
```

Arguments

phy a class 'phylo' phylogenetic tree

Author(s)

Dan Rabosky <DLR32@cornell.edu>

```
plotRate
```

Plot speciation and extinction rates through time

Description

Plots diversification rates through time estimated from fits of exponential speciation and extinction models (Rabosky & Lovette 2008).

Usage

```
plotRate(bt, pars)
```

Arguments

bt	A numeric vector of branching times
pars	a list of parameters, with named components lam0, k, mu0, and z

Details

Plots speciation and extinction curves through time for models fitted using [fitSPVAR](#), [fitEXVAR](#), and [fitBOTHVAR](#). If you want to plot the rate from SPVAR, note that this corresponds to assuming a very large z parameter; Try supplying z = 10000 or similar. Likewise, if you want to plot the fit from EXVAR, supply k = 0.

Value

Just plots the rates.

Author(s)

Dan Rabosky <DLR32@cornell.edu>

See Also

[fitSPVAR](#), [fitEXVAR](#), [fitBOTHVAR](#)

pureBirth

pureBirth

Description

Fits pure birth (Yule) model to set of branching times

Usage

```
pureBirth(x)
```

Arguments

x	a numeric vector of branching times
---	-------------------------------------

Value

A list containing the following elements:

LH	the log-likelihood at the maximum
aic	the Akaike Information Criterion
r1	the speciation rate giving the maximum log-likelihood

Author(s)

Dan Rabosky <DLR32@cornell.edu>

References

Nee, S. 2001. Inferring speciation rates from phylogenies. *Evolution* 55:661-668.

Yule, G. U. 1924. A mathematical theory of evolution based on the conclusions of Dr. J. C. Willis. *Phil. Trans. R. Soc. Lond. B* 213:21-87.

See Also

[yule-n-rate](#), [bd](#), [yuleWindow](#)

Examples

```
data("plethodon")
### loads branching times for plethodontid salamander dataset
pureBirth(plethodon)
```

RandomTrees

Random Phylogenetic Trees

Description

'rtrees5' and 'rtrees50' are sets of 5 and 50 phylogenetic trees of 25 taxa generated under a stochastic pure-birth (Yule) process using the program *Phyl-O-Gen*

Usage

```
data(rtrees50)
```

Source

Data generated using *Phyl-O-Gen* (<http://evolve.zoo.ox.ac.uk/software/PhyloGen/main.html>)

Examples

```
data(rtrees50)
```

```
scaleBranchingtimes
```

Scale Branching Times

Description

Scales a vector of branching times by a known basal divergence time.

Usage

```
scaleBranchingtimes(x, basal = 100)
```

Arguments

x	a numeric vector of branching times
basal	estimated time of the basal divergence for the clade

Details

This function can be used when analyzing branching times for which it is possible to estimate the basal divergence time. If branch lengths in a specified phylogenetic tree are given in units of genetic distance, but the basal divergence is estimated at Z mya, call `'res <- scaleBranchingtimes(x, basal = Z)'`. Rate parameters estimated by fitting variants of birth-death models will then be in units of lineage births or deaths per million years.

Value

a vector of scaled branching times

Author(s)

Dan Rabosky (DLR32@cornell.edu)

Examples

```
data(plethodon)
# suppose plethodon basal divergence occurred 23 mya:
svec <- scaleBranchingtimes(plethodon, basal = 23)
plotLtt(svec)
# plots lineages through time in units of time, rather than genetic
# distance

pureBirth(svec)
# returns speciation rate in units of lineages/million years
```

skinkdiversity *Species richness for Australian sphenomorphine skink genera*

Description

Species richness data for genera in `skinktree`; also data used for analysis in Rabosky et al. 2007

Usage

```
data(skinkdiversity)
```

Source

Rabosky, D. L., S. C. Donnellan, A. L. Talaba, and I. J. Lovette. 2007. Exceptional among-lineage variation in diversification rates during the radiation of Australia's largest vertebrate clade. *Proc. Roy. Soc. Lond. Ser. B* 274:2915-2923.

References

Rabosky, D. L., S. C. Donnellan, A. L. Talaba, and I. J. Lovette. 2007. Exceptional among-lineage variation in diversification rates during the radiation of Australia's largest vertebrate clade. *Proc. Roy. Soc. Lond. Ser. B* 274:2915-2923.

skinktree *Phylogenetic tree of Australian sphenomorphine skink genera*

Description

Phylogenetic tree of 23 Australian sphenomorphine skinks - including at least one representative of all genera.

Usage

```
data(skinktree)
```

Source

Rabosky, D. L., S. C. Donnellan, A. L. Talaba, and I. J. Lovette. 2007. Exceptional among-lineage variation in diversification rates during the radiation of Australia's largest vertebrate clade. *Proc. Roy. Soc. Lond. Ser. B* 274:2915-2923.

References

Rabosky, D. L., S. C. Donnellan, A. L. Talaba, and I. J. Lovette. 2007. Exceptional among-lineage variation in diversification rates during the radiation of Australia's largest vertebrate clade. *Proc. Roy. Soc. Lond. Ser. B* 274:2915-2923.

<code>truncateTree</code>	<i>Truncate a Set of Branching Times</i>
---------------------------	--

Description

This function is used to truncate a set of branching times derived from a phylogenetic tree. Incomplete taxon sampling causes a spurious decline in the rate of lineage accumulation over time, and this effect becomes more severe towards the present. Likewise, in the absence of dense phylogeographic sampling, it may be desirable to omit the final one or more nodes from the tree. `truncateTree` permits the user to omit k final nodes, or the final t time units.

Usage

```
truncateTree(x, omit.time = NULL, omit.nodes = NULL, batch = FALSE)
```

Arguments

<code>x</code>	a numeric vector of branching times
<code>omit.time</code>	remove the final 'omit.time' time units before present from tree
<code>omit.nodes</code>	remove the final 'omit.nodes' nodes before present
<code>batch</code>	if TRUE, processes batch of branching times. FALSE assumes as single set of branching times

Details

Consider a set of branching times `x = (100, 80, 50, 40, 30, 20, 10, 5)`. If you wanted to analyze only the first half of the tree, perhaps due to concern about incomplete sampling, calling `truncateTree(x, omit.time = 50)` would return a vector of branching times `x1 = (50, 30, 0)`. Likewise, if you wished to omit the final 2 branching times, `truncateTree(x, omit.nodes = 2)` would return `x2 = (90, 70, 40, 30, 20)`.

'batch' implies that you are entering a matrix or dataframe of branching times for multiple trees, as returned by `getBtimes.batch`. The 'omit.time' option is not available for batch processing.

Value

a numeric vector of branching times, or if `batch = TRUE`, a matrix of branching times.

Note

In the absence of dense phylogeographic sampling, it may be desirable to omit the final few nodes.

Author(s)

Dan Rabosky (DLR32@cornell.edu)

Examples

```
data(plethodon)
pleth2 <- truncateTree(plethodon, omit.nodes = 2)
#omits final 2 branching times

plotLtt(pleth2)
```

warblers	<i>Branching times from phylogeny of North American Dendroica warblers</i>
----------	--

Description

Branching times from ultrametric phylogenetic tree for Dendroica wood-warblers

Usage

```
data(warblers)
```

Source

Lovette, I.J., E. Bermingham. 1999. Explosive speciation in the New World Dendroica warblers. *Proc. Roy. Soc. B. Lond.* 266:1629-1636.

References

Lovette, I.J., E. Bermingham. 1999. Explosive speciation in the New World Dendroica warblers. *Proc. Roy. Soc. B. Lond.* 266:1629-1636.

Rabosky, D. L., Lovette, I. J. 2008. Explosive evolutionary radiations: decreasing speciation or increasing extinction through time? *Evolution* In press.

Examples

```
data(warblers)
plotLtt(warblers)
```

yule-n-rate	<i>yule-n-rate</i>
-------------	--------------------

Description

Fits multi-rate variants of the pure birth (Yule) model to branching times derived from phylogenetic data. For example, the *yule2rate* model assumes that the clade has diversified under speciation rate $r1$ until some time st , at which point the speciation rate shifts to a new rate $r2$. The shift point(s) are found by optimizing parameters and computing likelihoods for a set of possible shift times and selecting the parameter combinations giving the maximum log-likelihood.

Usage

```
yule2rate(x, verbose = FALSE, ints = NULL, file = "out_yule2rate.txt")
yule3rate(x, ints = NULL, verbose = FALSE, file = "out_yule3rate.txt")
yule4rate(x, ints = NULL)
yule5rate(x, ints = NULL)
```

Arguments

x	a numeric vector of branching times
verbose	if 'verbose = TRUE', writes likelihoods and parameter estimates for all shift points considered to the specified file. Default is FALSE. Only available for <i>yule2rate</i> and <i>yule3rate</i> models
ints	the number of intervals. See details
file	a filename for output if 'verbose = TRUE'

Details

'verbose' - for *yule2rate* and *yule3rate* models, maximum log-likelihoods and parameter estimates for each shift time under consideration will be output to file. The file can then be loaded to examine the likelihood of a rate shift at different points in time.

'ints' is used in determining the number of shift points to consider. If 'ints = NULL' (the default), the model will consider only observed branching times as possible shift points. Suppose we have a small dataset with the following branching times: (100, 80, 50, 40, 30, 20, 10, 5, 2). Under the *yule2rate* model, we assume that the clade has diversified under some rate $r1$ until some time ts , at which point the rate simultaneously shifts in all lineages to a new rate $r2$. In this example, if 'ints = NULL', we would use the set of observed branching times only, but omitting the first and final branching times (thus, we would be considering only $st = (80, 50, 40, 30, 20, 10, 5)$ as possible shift points. If 'ints = 100', we would consider 100 evenly spaced shift points on the interval between the 2nd branching time and the N-1 branching time (e.g., on (80, 5)). 'ints' works well for *yule2rate* and *yule3rate* models, but can result in high computational times for *yule4rate* and *yule5rate* models.

Value

a data frame containing the following elements:

LH	the maximum log-likelihood
AIC	the Akaike Information Criterion
r1	the first (earliest) speciation rate giving the maximum log-likelihood
r2	the ML estimate of the second speciation rate (in the case of <i>yule2rate</i> , this will be the final speciation rate)
st1	the earliest shift point (for <i>yule2rate</i> , you will only have a single shift point)
...	speciation rates and shift points for models other than <i>yule2rate</i> are abbreviated by <i>r3</i> , <i>st2</i> , etc, as described above

Note

The total number of parameters for each model is equal to the number of speciation rates and shift points subject to optimization. Thus, the *yule3rate* model has 3 speciation rates and 2 shift times, for a total of 5 parameters. Strictly speaking, it may be inappropriate to treat the shift time *st* as a free parameter, as it can only take on a limited set of values. However, in practice, it appears to work well; in many cases, using observed shift times can give higher likelihoods than when 'ints' are specified. There seems to be little improvement in the log-likelihood with 'ints' greater than 1000, at least for phylogenies with fewer than 100 tips.

Note that shift times, like branching times, are given in divergence units before present. Thus, if you have scaled a set of branching times to a basal divergence of 30 million years before present, you would interpret 'st1 = 19.5' as an inferred shift point 19.5 million years before present.

Author(s)

Dan Rabosky (DLR32@cornell.edu)

References

Barracough, T. G., and A. P. Vogler. 2002. Recent diversification rates in North American tiger beetles estimated from a dated mtDNA phylogenetic tree. *Mol. Biol. Evol.* 19:1706-1716.

Rabosky, D. L. 2006. Likelihood methods for inferring temporal shifts in diversification rates. *Evolution* 60:1152-1164.

See Also

`bd` , `fitdAICrc` , `yuleWindow` , `pureBirth`

Examples

```
data(plethodon)

### fitting a 3-rate Yule model to the Plethodon data:
```

```

result <- yule3rate(plethodon)

### gives data frame with maximum log-likelihood and parameter estimates
### at the max.
### In this case, we would access individual parameters as
### result$LH (the max), result$st1 (first shift time), result$st2
### (the second shift time), and result$r1, result$r2, and result$r3
### for the speciation rates.

### Here we will use 'yule2rate' to output maximum log-likelihoods
### for each shift point considered, then load the file and plot
### log-likelihoods of a rate shift against 'time from basal divergence'
### to graphically explore the tempo of diversification

# result <- yule2rate(plethodon, ints = NULL,
#                   verbose = TRUE, file = 'out.txt')
# LHtable <- read.table(file = 'out.txt', header = TRUE)

### 'header = TRUE' ensures that variable names are correctly read

### rescaling shift times so that they reflect 'time from basal divergence':

# LHtable$st1 <- plethodon[1] - LHtable$st1
# plot(LHtable$LH~LHtable$st1, xlab = 'Time From Basal Divergence',
#      ylab = 'Log-likelihood')

```

yuleSim

yuleSim

Description

Simulate Branching Times Under the Pure Birth Model

Usage

```
yuleSim(ntaxa, nsets, lambda = 0.01)
```

Arguments

ntaxa	number of taxa in each set of branching times
nsets	number of datasets of size ntaxa to be simulated
lambda	speciation rate (default = 0.01)

Value

an $n \times m$ matrix of branching times, where n is size ntaxa and m is size nsets

Author(s)

Dan Rabosky (DLR32@cornell.edu)

Examples

```
testdata <- yuleSim(25, 50, lambda = .001)
```

yuleWindow

Fit Yule Model (Pure Birth) to Temporal Window of Branching Times

Description

This function fits the Yule model to any temporal window that includes at least one branching time.

Usage

```
yuleWindow(x, st1 = x[1], st2 = 0)
```

Arguments

x	a numeric vector of branching times
st1	the start of the interval you wish to examine
st2	the end of the interval you wish to examine

Details

'st1' and 'st2' are given in divergence units before present. If we had a set of branching times with an initial divergence 100 mya, 'yuleWindow(x, 75, 25)' would fit the pure birth model to the portion of the tree between 75 and 25 mya. Calling 'yuleWindow(x, x[1], 0)' will fit the model to the entire tree (identical to 'pureBirth(x)').

Note that st1 must be greater than st2, because they are given in units of divergence before present.

Value

a list with the following components:

LH	the log-likelihood at the maximum
smax	the speciation rate giving the maximum log-likelihood

Note

This can be used in conjunction with other models in this package to test *a priori* hypotheses of rate variation. If, for example, it was hypothesized that a particular climatic event occurring T mya shifted diversification rates, you can use 'yuleWindow(x, x[1], T)' and 'yuleWindow(x, T, 0)' to obtain the log-likelihoods for these two temporal windows. The log-likelihoods can then be summed to obtain the likelihood of the full set of branching times under a 2-rate Yule model with an *a priori* hypothesized rate-shift. This model would only have two free parameters, in contrast to the [yule2rate](#) model, with three parameters (and thus, the AIC would be computed as $-2 * (\text{sum of log-likelihoods}) + 4$).

Author(s)

Dan Rabosky <DLR32@cornell.edu>

References

Nee, S., R. M. May, and P. H. Harvey. 1994b. The reconstructed evolutionary process. *Philos. Trans. R. Soc. Lond. B* 344:305-311.

Nee, S. 2001. Inferring speciation rates from phylogenies. *Evolution* 55:661-668.

See Also

[fitdAICrc](#), [pureBirth](#), [bd](#)

Examples

```
data(agamids)
agbtimes <- getBtimes(string = agamids)
yuleWindow(agbtimes, 0.22, 0.10)
# fits Yule model to temporal window between 0.22 and 0.10 divergence
# units before present
```

Index

- *Topic **IO**
 - getBtimes, 18
 - getBtimes.batch, 19
- *Topic **aplot**
 - plotLtt, 25
- *Topic **datagen**
 - birthdeathSim, 4
 - yuleSim, 34
- *Topic **datasets**
 - agamids, 2
 - plethodon, 24
 - RandomTrees, 28
 - skinkdiversity, 29
 - skinktree, 30
 - warblers, 31
- *Topic **hplot**
 - plotLtt, 25
- *Topic **htest**
 - fitdAICrc, 8
 - fitdAICrc.batch, 10
 - gamStat, 17
 - mccrTest.Rd, 23
 - yuleWindow, 35
- *Topic **manip**
 - getBtimes, 18
 - getBtimes.batch, 19
 - scaleBranchingtimes, 28
 - truncateTree, 30
- *Topic **models**
 - bd, 3
 - calcLHbd, 5
 - DensityDependent, 6
 - fitdAICrc, 8
 - fitdAICrc.batch, 10
 - fitNDR_1rate.Rd, 11
 - fitNDR_2rate.Rd, 13
 - fitSPVAR, 15
 - getTipdata.Rd, 20
 - MagallonSanderson.Rd, 21
 - plotNodeNumbers.phylo.Rd, 25
 - plotRate, 26
 - pureBirth, 27
 - yule-n-rate, 32
 - yuleWindow, 35
- *Topic **package**
 - laser-package, 1
- *Topic **univar**
 - gamStat, 17
 - mccrTest.Rd, 23
- agamids, 2
- bd, 3, 8, 27, 36
- birthdeathSim, 4
- calcLHbd, 5
- DDL, 8, 9
- DDL (*DensityDependent*), 6
- DDX, 8, 9
- DDX (*DensityDependent*), 6
- DensityDependent, 6
- fitBOTHVAR, 26
- fitBOTHVAR (*fitSPVAR*), 15
- fitdAICrc, 4, 7, 8, 10, 36
- fitdAICrc.batch, 10
- fitEXVAR, 26
- fitEXVAR (*fitSPVAR*), 15
- fitNDR_1rate, 14, 21, 22
- fitNDR_1rate (*fitNDR_1rate.Rd*), 11
- fitNDR_1rate.Rd, 11
- fitNDR_2rate, 12, 21, 22
- fitNDR_2rate (*fitNDR_2rate.Rd*), 13
- fitNDR_2rate.Rd, 13
- fitSPVAR, 15, 26
- gamStat, 17, 24
- getBtimes, 18, 19, 20
- getBtimes.batch, 10, 18, 19, 31

getTipdata, [12](#), [14](#), [21](#), [22](#)
getTipdata (*getTipdata.Rd*), [20](#)
getTipdata.Rd, [20](#)

lambda.crown.ms01
 (*MagallonSanderson.Rd*), [21](#)
lambda.stem.ci
 (*MagallonSanderson.Rd*), [21](#)
lambda.stem.ml, [12](#), [14](#), [21](#), [22](#)
lambda.stem.ml
 (*MagallonSanderson.Rd*), [21](#)
lambda.stem.ms01
 (*MagallonSanderson.Rd*), [21](#)
laser (*laser-package*), [1](#)
laser-package, [1](#)

MagallonSanderson.Rd, [21](#)
mccrTest (*mccrTest.Rd*), [23](#)
mccrTest.Rd, [23](#)

plethodon, [24](#)
plotLtt, [25](#)
plotNodeNumbers.phylo
 (*plotNodeNumbers.phylo.Rd*),
 [25](#)
plotNodeNumbers.phylo.Rd, [25](#)
plotRate, [16](#), [26](#)
pureBirth, [3](#), [4](#), [8](#), [27](#), [36](#)

RandomTrees, [28](#)
rtrees5 (*RandomTrees*), [28](#)
rtrees50 (*RandomTrees*), [28](#)

scaleBranchingtimes, [28](#)
skinkdiversity, [29](#)
skinktree, [30](#)

truncateTree, [30](#)

warblers, [31](#)

yule-n-rate, [4](#), [7](#), [8](#), [27](#)
yule-n-rate, [32](#)
yule2rate, [8](#), [36](#)
yule2rate (*yule-n-rate*), [32](#)
yule3rate, [8](#)
yule3rate (*yule-n-rate*), [32](#)
yule4rate (*yule-n-rate*), [32](#)
yule5rate (*yule-n-rate*), [32](#)
yuleSim, [34](#)
yuleWindow, [27](#), [35](#)