

Package ‘BAMMtools’

July 10, 2014

Type Package

Title Analysis and visualization of macroevolutionary dynamics on phylogenetic trees

Version 2.0.2

Date 2014-07-09

Author Dan Rabosky, Mike Grundler, Pascal Title, Carlos Anderson, Jeff Shi, Joseph Brown, Huteng Huang

Maintainer Dan Rabosky <dan.rabosky@gmail.com>

Depends ape

Description BAMMtools provides functions for analyzing and visualizing complex macroevolutionary dynamics on phylogenetic trees. It is a companion package to the command line program BAMM (Bayesian Analysis of Macroevolutionary Mixtures) and is entirely oriented towards the analysis, interpretation, and visualization of evolutionary rates. Functionality includes visualization of rate shifts on phylogenies, estimating evolutionary rates through time, comparing posterior distributions of evolutionary rates across clades, comparing diversification models using Bayes factors, and more.

License GPL (>= 2)

URL <http://www.bamm-project.org>

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-07-10 05:11:24

R topics documented:

BAMMtools-package	2
addBAMMshifts	3
assignColorBreaks	5
BAMMtools-data	6

BAMMtools-internal	7
bayesFactorBranches	8
cohorts	9
computeBayesFactors	10
credibleShiftSet	12
distinctShiftConfigurations	16
dtRates	17
getBestShiftConfiguration	19
getBranchShiftPriors	20
getCladeRates	22
getCohortMatrix	24
getEventData	25
getMarginalBranchRateMatrix	28
getMeanBranchLengthTree	29
getmrca	30
getRateThroughTimeMatrix	31
getShiftNodesFromIndex	33
getTipRates	34
maximumShiftCredibility	36
plot.bammdata	37
plot.bammshifts	41
plot.credibleshiftset	43
plotRateThroughTime	44
samplingProbs	47
setBAMMpriors	48
ShiftProbsTree	50
speciesByRatesMatrix	51
subsetEventData	53
subtreeBAMM	54
summary.bammdata	55
summary.branchprior	56
summary.credibleshiftset	56
testTimeVariableBranches	58

Index	61
--------------	-----------

BAMMtools-package	<i>BAMMtools</i>
-------------------	------------------

Description

An R package for the analysis and visualization of complex macroevolutionary dynamics. Functions in BAMMtools are oriented entirely around analysis of results obtained using the BAMM software (bamm-project.org).

Details

Package: BAMMtools
 Type: Package
 Version: 1.0.1
 Date: 2014-03-07
 License: GPL (>= 2)

Author(s)

Dan Rabosky, Mike Grundler, Pascal Title, Carlos Anderson, Jeff Shi, Joseph Brown, Huateng Huang
 Maintainer: Dan Rabosky <dan.rabosky@gmail.com>

References

bamm-project.org

Rabosky, D. L. 2014. Automatic detection of key innovations, rate shifts, and diversity-dependence on phylogenetic trees. PLoS ONE 9: e89543

Rabosky, D. L., F. Santini, J. T. Eastman, S. A. Smith, B. L. Sidlauskas, J. Chang, and M. E. Alfaro. 2013. Rates of speciation and morphological evolution are correlated across the largest vertebrate radiation. Nature Communications DOI: 10.1038/ncomms2958

addBAMMshifts

Add BAMM-inferred rate shifts to a phylogeny plot

Description

Adds symbols to a plotted tree to mark the location(s) where there is a shift in the macroevolutionary dynamics of diversification or trait evolution.

Usage

```

addBAMMshifts(ephy, index = 1, method = 'phylogram',
cex = 1, pch = 21, col = 1, bg = 2, msp = NULL,
shiftnodes = NULL, par.reset = TRUE)
  
```

Arguments

ephy	an object of class bammdata.
method	a character string indicating the method used in plotting. Must be "polar" or "phylogram".
index	an integer indicating which posterior sample to use for adding shifts to the plotted tree.

<code>cex</code>	a numeric indicating the character expansion ("size") of the plotted points.
<code>pch</code>	an integer indicating the choice of plotting symbol.
<code>col</code>	an integer or character string indicating the border color of the plotting symbol.
<code>bg</code>	an integer or character string indicating the background color of the plotting symbol.
<code>msp</code>	if not NULL, an object of class <code>phylo</code> where each branch length is equal to the marginal probability of a shift occurring on that branch. Plotted points corresponding to shifts will be sized by these probabilities.
<code>shiftnodes</code>	an optional vector of node numbers indicating the locations of shifts to plot.
<code>par.reset</code>	a logical indicating whether to reset the graphical parameters before exiting.

Details

Any given sample from the posterior distribution sampled using BMM contains a potentially unique configuration of rate shifts and associated parameters. There is no single "best" rate shift, but rather a set of shift configurations (and associated parameters) - along with their relative probabilities - sampled with MCMC. This function enables the user to plot the locations of shifts sampled with BMM for a given sample from the posterior.

If the `bamdata` object contains just a single sample, these shifts will be plotted regardless of the value of `index`.

Note

If a `shiftnodes` argument is passed care should be taken to ensure that the nodes are in the same order as in the event data for the sample `index`.

Author(s)

Mike Grundler

See Also

[getShiftNodesFromIndex](#), [plot.bamdata](#)

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25)

#adding shifts to tree for specific posterior samples
plot(ed, method="polar")
addBAMMshifts(ed, index=5, "polar")

#multi-panel plotting and adding shifts
par(mfrow=c(2,3),mar=c(5,1,1,1))
samples = sample(1:length(ed$eventData), 6)
for (i in 1:6) {
```

```
sed <- subsetEventData(ed, samples[i])
plot(sed, par.reset=FALSE)
addBAMMshifts(sed, index=1, method="phylogram", par.reset=FALSE)
}
```

assignColorBreaks *Map macroevolutionary rates to colors*

Description

Maps macroevolutionary rates to a set of NCOLORS.

Usage

```
assignColorBreaks(rates, NCOLORS, spex, log = FALSE)
```

Arguments

rates	a numeric vector of phenotypic rates or a list of numeric vectors of speciation and extinction rates.
NCOLORS	an integer number of colors to use for the mapping. Larger numbers do not necessarily result in smoother looking color ramps. The default is 64 and is probably sufficient for most purposes.
spex	a character string. "s" means that speciation rates are used to make the map, "e" means that extinction rates are used. Any other string, e.g. "se", means that diversification rates are used. Ignored for BAMM trait data.
log	should the natural logarithm of rates be used for the color map.

Details

Macroevolutionary rates are binned into NCOLORS+1 percentiles and rates in each bin are mapped to a color determined by the pal argument in plot.bamdata.

Value

A numeric vector of rate percentiles.

Author(s)

Mike Grundler

See Also

[plot.bamdata](#)

Examples

```

data(whales, events.whales)
ed <- getEventData(whales,events.whales,0.2)

ed <- dtRates(ed, 0.01)
colors <- assignColorBreaks(ed$dtrates$rates, spex="s") #speciation rates
#colors <- assignColorBreaks(ed$dtrates$rates[[1]])
#this also works for speciation rates

plot(ed, colorbreaks = colors, spex="s")
colors <- assignColorBreaks(ed$dtrates$rates, spex="se")
#diversification rates

#colors <- assignColorBreaks(ed$dtrates$rates[[1]] - ed$dtrates$rates[[2]])
#this also works for diversification rates

plot(ed, colorbreaks = colors, spex="se")

```

BAMMtools-data

BAMMtools datasets

Description

Example datasets and sample BAMM output for the package BAMMtools

Usage

```

data(whales)
data(primates)
data(mcmc.whales)
data(mcmc.primates)
data(prior.primates)
data(prior.whales)
data(events.whales)
data(events.primates)
data(mass.primates)

```

Details

This includes both the raw data and the BAMM output for two example analyses. The first is an analysis of speciation and extinction rates during the radiation of modern whales, using a time-calibrated tree from Steeman et al. (2009). The second is a BAMM analysis of phenotypic evolutionary rates (body mass) during the radiation of extant primates, taken from Vos and Mooers (2006) and Redding et al. (2010).

Dataset whales is the raw time-calibrated tree that was analyzed with BAMM, and primates is the corresponding time-calibrated phylogeny of 233 primate species. Log-transformed body masses for primates are in dataset mass.primates.

The MCMC output files (`mcmc.whales` and `mcmc.primates`) are dataframes containing the raw MCMC output as generated by BAMM. Column headers in the dataframes includes the sampling generation, the current number of shifts in the simulation (`N_shifts`), the log-prior density of the parameters (`logPrior`), the log-likelihood of the data (`logLik`), the current parameter of the Poisson process governing the number of regime shifts (`eventRate`), and the MCMC acceptance rate (`acceptRate`). This is the file that would typically be analyzed as a first step towards assessing MCMC convergence (e.g., analyzing effective sample sizes of `logLik` and `N_shifts`).

The prior datasets (`prior.primates` and `prior.whales`) are the corresponding prior densities for the number of shifts generated by BAMM. These dataframes include only 3 columns: `generation`, `N_shifts`, and `logPrior`. Their utility lies in enabling the user to estimate Bayes factors for model comparison (see function `computeBayesFactors`).

The "core" BAMM output is included in the *event data* files (`events.whales` and `events.primates`). These are all the parameters sampled with MCMC that are relevant to reconstructing the nature and location of evolutionary rate dynamics across a phylogeny. Please refer to BAMM documentation for a detailed overview of this output, but a brief description is as follows:

`generation`: The index value of the state in the MCMC simulation (the "generation").

`leftchild`, `rightchild`: This defines a unique topological location where a rate shift was sampled. Specifically, for given right-left pair, the shift is sampled on the branch leading to the node from which `rightchild` and `leftchild` are descended (these two taxa are part of the spanning set of taxa for the node). If `leftchild` is "NA", this simply means that the shift was sampled on a terminal branch.

`abstime`: The absolute occurrence time of the shift, assuming that the time of the root node is 0.0.

`lambdainit`, `lambdashift`: for speciation extinction model, the initial speciation rate and rate change parameter for the process.

`muinit`: for speciation extinction model, the extinction rate (time-invariant)

`betainit`, `betashift`: for phenotypic evolutionary model, the initial (`betainit`) rate of phenotypic evolution and the rate change parameter (`betashift`).

References

- Vos R.A., Mooers A.O. 2006. A new dated supertree of the Primates. Chapter 5. In *Inferring large phylogenies: the big tree problem* (R Vos, Ph.D. thesis) Simon Fraser University.
- Redding D.W., DeWolff C., Mooers A.O. 2010. Evolutionary distinctiveness, threat status and ecological oddity in primates. *Conserv. Biol.* 24:1052-1058.
- Steehan, M. E., M. B. Hebsgaard, R. E. Fordyce, S. W. Y. Ho, D. L. Rabosky, R. Nielsen, C. Rahbek, H. Glenner, M. V. Sorensen, E. Willerslev. 2009. Evolution and radiation of extant whales. *Systematic Biology.* 58:573-585.

BAMMtools-internal *Internal BAMMtools functions*

Description

BAMMtools-internal functions are not routinely called by the user and are undocumented.

Details

Internal **BAMMtools** functions, either not intended to be called directly by the user or otherwise undocumented.

bayesFactorBranches *Bayes factor evidence for rate shifts on individual branches*

Description

Computes Bayes factor evidence for a rate shift on every branch of a phylogenetic tree

Usage

```
bayesFactorBranches(ephy, priordata)
```

Arguments

ephy	an object of class bammdata
priordata	an object of class branchprior

Details

This function returns a copy of a phylogenetic tree where each branch length is equal to the Bayes factor evidence in favor of a rate shift on a particular branch. Here, we imagine that there are two models: one with a rate shift on a given branch, and one without. Treating the "no shift" model as something akin to a null model, the Bayes factor can be computed as the ratio of posterior to prior odds of the model with a shift on a given branch, divided by the ratio of posterior to prior odds of the model without a shift.

By computing branch-specific marginal shift probabilities (using [marginalShiftProbsTree](#)), you can immediately extract the posterior probabilities of the relevant models. The branch length returned by this function is the posterior probability of a rate shift (*prob_shift*), and the posterior probability of the model with no shift is simply $1 - prob_shift$.

Likewise, you can compute the prior odds in much the same way. The function [getBranchShiftPriors](#) computes the *prior* probability of a rate shift on each branch. As above, the prior probability of the *no shift* model is just $1 - prob_noshift$. The posterior-to-prior odds ratio for the shift model is $(posterior\ prob\ of\ shift) / (prior\ prob\ of\ shift)$. Simply multiplying this term by $(prior\ prob\ of\ no\ shift) / (posterior\ prob\ of\ no\ shift)$ gives us the Bayes factor.

In general, Bayes factors greater than 10 imply evidence in support of a rate shift, and Bayes factors greater than 100 imply very strong evidence for a shift on a particular branch.

Value

A object of class `phylo` but where each branch length is equal to the Bayes factor evidence in favor of a rate shift on a particular branch.

Author(s)

Dan Rabosky

See Also[getBranchShiftPriors](#), [distinctShiftConfigurations](#), [credibleShiftSet](#)**Examples**

```

data(whales, events.whales, prior.whales)
ed <- getEventData(whales, events.whales, burnin=0.1)
pdata <- getBranchShiftPriors(whales, prior.whales)
bftree <- bayesFactorBranches(ed, pdata)
margprobs <- marginalShiftProbsTree(ed)
plot.new()
par(mfcol=c(1,3))
# Original tree:
plot.phylo(whales, show.tip.label=FALSE)
mtext(side=3, text="original tree", cex=1)

plot.phylo(margprobs, show.tip.label=FALSE)
mtext(side=3, text="Marginal probs", cex=1)

plot.phylo(bftree, show.tip.label=FALSE)
mtext(side=3, text="Bayes factors", cex=1)

```

cohorts

*Visualize macroevolutionary cohorts***Description**

Plots the matrix of pairwise correlations in rate regimes between all tips in a phylogeny

Usage

```
cohorts(x, ephy, col, pal, lwd = 1, ofs = 0, use.plot.bamdata = TRUE,
useraster = FALSE, LARGE = 500)
```

Arguments

<code>x</code>	a matrix of pairwise correlations generated by <code>getCohortMatrix</code> .
<code>ephy</code>	an object of class <code>bamdata</code> .
<code>col</code>	a vector of colors passed to the function <code>image</code> . These will be used to color the values in <code>x</code> . See documentation for <code>image</code> .
<code>pal</code>	the palette to use if <code>use.plot.bamdata=TRUE</code> . See options documented in the help file for <code>plot.bamdata</code> .

lwd	a numeric indicating the width of branches in the phylogeny.
ofs	a numeric controlling the offset of the phylogeny from the matrix plot. Appropriate values will probably be in the interval [0,0.1].
use.plot.bammdata	logical. should a phylorate plot be generated?
useraster	a logical indicating whether the function image should plot the matrix as a raster.
LARGE	an integer. If trees have more tips than LARGE, useraster will be coerced to TRUE.

Details

The plotting function creates an image of the BAMM correlation matrix between tip lineages of the phylogeny. Each correlation is the posterior frequency with which a pair of lineages occurs in the same macroevolutionary rate regime. Correlations are mapped to a set of colors, with warmer colors corresponding to higher correlations than cooler colors. The set of colors is specified by the `col` argument and a legend is plotted to guide interpretation of the color-correlation map. Trees are plotted on the margins of the matrix image. The correlation between any two tips can be inferred by finding their intersection within the matrix image.

Author(s)

Mike Grundler

See Also

[plot.bammdata](#), [getCohortMatrix](#), [image](#)

Examples

```
library(BAMMtools)
data(whales, events.whales)
ed <- getEventData(whales, events.whales, 0.1)
x <- getCohortMatrix(ed)
cohorts(x, ed)
cohorts(x, ed, col=richColors(64))
cohorts(x, ed, ofs=0.05, col=richColors(64))
cohorts(x, ed, pal="temperature", col=richColors(64))
# gray scale
cohorts(x, ed, col=gray(seq(0.2,0.9,length.out=128)), use.plot.bammdata=FALSE)
```

computeBayesFactors *computeBayesFactors*

Description

Computes pairwise Bayes factors for a set of macroevolutionary models sampled using BAMM, using MCMC simulation output.

Usage

```
computeBayesFactors(postdata, priordata, burnin = 0.1, modelset = NULL, ...)
```

Arguments

postdata	Filename for the MCMC output file from a BMM run. Alternatively, a dataframe containing this information.
priordata	Filename for the BMM-generated output file containing a simulation of shifts under the prior. Alternatively, a dataframe containing this information.
burnin	What fraction of samples to discard from postdata as burnin?
modelset	Integer set of models for which you wish to compute Bayes factors. An integer value of k implies a model with k rate shifts.
...	Additional arguments to computeBayesFactors

Details

This function returns a matrix of pairwise Bayes factors, where the Bayes factor is the ratio of marginal likelihoods between two models M_i and M_j . Numerator models are given as rows, and denominator models as columns. Row names and column names give the number of shifts in the corresponding model. Suppose you have an output matrix with row and column names 0:3 (0, 1, 2, 3). Model 0 is a model with just a single process (starting at the root), and no among-lineage rate heterogeneity.

If `computeBayesFactors` gives a matrix `mm`, and `mm[2, 1]` is 10.0, this implies Bayes factor evidence of 10 in favor of the 2nd row model (a model with 1 process; e.g., `rownames(mm)[2]`) over the first column model (a model with a single process).

It can be very difficult to approximate either the posterior or prior probability of models that are very rarely sampled. Suppose you have a simulation of a prior distribution that samples the models (M0, M1, M2, M3). Now, suppose your posterior distribution samples the models (M4, M5, M6, M7). There is no overlap between these sets of models. In other words, models sampled during the simulation of the prior are *so improbable* that they were not sampled during simulation of the posterior. Hence, we can't even estimate their posterior probabilities with any degree of accuracy. Likewise, the models sampled during simulation of the posterior are very unlikely under the prior, and we can't estimate their prior probabilities with any degree of accuracy. The function `computeBayesFactors` will automatically check and make sure that it is only computing Bayes factors for models where a non-zero probability (posterior or prior) can be estimated.

This is likely to change in the future with more robust methods for estimating these quantities in the tails of the distributions. However, keep in mind that if your posterior is very far from your prior, such that these models don't overlap, you actually have strong evidence in favor of rate shifts (if the mean of the posterior distribution is greater than the prior mean) - this implies that the models expected under the prior alone have effectively zero posterior probability.

Value

A matrix of pairwise Bayes factors between models.

Author(s)

Dan Rabosky

Examples

```
data(mcmc.whales)
data(prior.whales)
computeBayesFactors(mcmc.whales, prior.whales, burnin=0.1)
```

credibleShiftSet	<i>Credible set of macroevolutionary rate shift configurations from BAMM results</i>
------------------	--

Description

Computes the 95% (or any other %) credible set of macroevolutionary rate shift configurations from a bammdata object. These results can be analyzed further and/or plotted.

Usage

```
credibleShiftSet(ephy, prior, BFcriterion = 5, set.limit = 0.95, ...)
```

Arguments

ephy	an object of class bammdata
prior	An object of class branchprior. See getBranchShiftPriors
BFcriterion	The Bayes factor criterion for including nodes as core shifts in the enumeration of distinct shift configurations.
set.limit	the desired limit to the credible set. A value of 0.95 will return the 95% credible set of shift configurations.
...	Other arguments to credibleShiftSet

Details

Computes the 95% credible set (or XX% credible set, depending on set.limit) of diversification shift configurations sampled using BAMM. This is analogous to a credible set of phylogenetic tree topologies from a Bayesian phylogenetic analysis.

To understand how this calculation is performed, one must first distinguish between "core" and "non-core" rate shifts. A "core shift" is a rate shift with a marginal probability that is substantially elevated above the probability expected on the basis of the prior alone. With BAMM, every branch in a phylogenetic tree is associated with some non-zero prior probability of a rate shift. Typically this is a very low per-branch shift probability (this prior is determined by the value of the "poisson-RatePrior" parameter in a BAMM analysis).

If we compute distinct shift configurations with every sampled shift (including those shifts with very low marginal probabilities), the number of distinct shift configurations will be overwhelmingly high. However, most of these configurations include shifts with marginal probabilities that are expected

even under the prior alone. Hence, using these shifts to identify distinct shift configurations simply generates noise and isn't particularly useful.

The solution adopted in BAMMtools is, for each branch in the phylogeny, to compute both the posterior and prior probabilities of a rate shift occurring. The ratio of these probabilities is a branch-specific Bayes factor: it is the posterior probability of a rate shift normalized by the corresponding prior probability. Hence, any branch with a Bayes factor value of 1.0 is one where the observed (posterior) odds of a rate shift are no different from the prior odds. A value of 10 implies that the posterior probability is 10 times the prior probability.

The user of `credibleShiftSet` must specify a `BFcriterion` argument. This is simply a Bayes factor cutoff value for identifying "important" shifts for the purposes of identifying distinct shift configurations. See the online documentation on this topic available at www.bamm-project.org for more information. If you specify `BFcriterion = 5` as an argument to `credibleShiftSet`, the function will ignore all shifts with Bayes factors less than 5 during the enumeration of topologically distinct shift configurations. Only shifts with a Bayes factor value greater than or equal to `BFcriterion` will be treated as core shifts for the purposes of identifying distinct shift configurations.

For each shift configuration in the credible set, this function will compute the average diversification parameters. For example, the most frequent shift configuration (the maximum a posteriori shift configuration) might have 3 shifts, and 150 samples from your posterior (within the `bammdata` object) might show this shift configuration. However, the parameters associated with each of these shift configurations (the actual evolutionary rate parameters) might be different for every sample. This function returns the mean set of rate parameters for each shift configuration, averaging over all samples from the posterior that can be assigned to a particular shift configuration.

See the examples below for a worked example.

Value

A class `credibleshiftset` object with many components. Most components are an ordered list of length `L`, where `L` is the number of distinct shift configurations in the credible set. The first list element in each case corresponds to the shift configuration with the maximum a posteriori probability.

<code>frequency</code>	A vector of frequencies of shift configurations, including those that account for <code>set.limit</code> (typically, 0.95 or 0.99) of the probability of the data. The index of the <i>i</i> 'th element of this vector is the <i>i</i> 'th most probable shift configuration (excepting ties).
<code>shiftnodes</code>	A list of the "core" rate shifts (marginal probability > threshold) that occurred in each distinct shift configuration in the credible set. The <i>i</i> 'th vector from this list gives the core shift nodes for the <i>i</i> 'th shift configuration. They are sorted by frequency, so <code>x\$shiftnodes[[1]]</code> gives the shift nodes that occurred together in the shift configuration with the highest posterior probability.
<code>indices</code>	A list of vectors containing the indices of samples in the <code>bammdata</code> object that are assigned to a given shift configuration. All are sorted by frequency.
<code>cumulative</code>	Like <code>frequency</code> , but contains the cumulative frequencies
<code>BFcriterion</code>	The Bayes factor criterion used during enumeration of distinct shift configurations.

<code>number.distinct</code>	Number of distinct shift configurations in the credible set
<code>set.limit</code>	which credible set is this (0.9, 0.95, etc)?
<code>coreshifts</code>	A vector of node numbers corresponding to the core shifts. All of these nodes have a Bayes factor of at least <code>BFcriterion</code> supporting a rate shift.

In addition, a number of components that are defined similarly in class `phylo` or class `bammdata` objects:

<code>edge</code>	See documentation for class <code>phylo</code> in package <code>ape</code> .
<code>Nnode</code>	See documentation for class <code>phylo</code> in package <code>ape</code> .
<code>tip.label</code>	See documentation for class <code>phylo</code> in package <code>ape</code> .
<code>edge.length</code>	See documentation for class <code>phylo</code> in package <code>ape</code> .
<code>begin</code>	The beginning time of each branch in absolute time (the root is set to time zero)
<code>end</code>	The ending time of each branch in absolute time.
<code>numberEvents</code>	An integer vector with the number of core events contained in the <code>bammdata</code> object for each shift configuration in the credible set. The length of this vector is equal to the number of distinct shift configurations in the credible set.
<code>eventData</code>	A list of dataframes. Each element holds the average rate and location parameters for all samples from the posterior that were assigned to a particular distinct shift configuration. Each row in a dataframe holds the data for a single event. Data associated with an event are: <code>node</code> - a node number. This identifies the branch where the event originates. <code>time</code> - this is the absolute time on that branch where the event originates (with the root at time 0). <code>lam1</code> - an initial rate of speciation or trait evolution. <code>lam2</code> - a decay/growth parameter. <code>mu1</code> - an initial rate of extinction. <code>mu2</code> - a decay/growth parameter. <code>index</code> - a unique integer associated with the event. See 'Details' in the documentation for <code>getEventData</code> for more information.
<code>eventVectors</code>	A list of integer vectors. Each element is for a single shift configuration in the posterior. For each branch in the <code>bammdata</code> object, gives the index of the event governing the (tipwards) end of the branch. Branches are ordered increasing here and elsewhere.
<code>eventBranchSegs</code>	A list of matrices. Each element of the list is a single distinct shift configuration. Each matrix has four columns: Column 1 identifies a node in <code>phy</code> . Column 2 identifies the beginning time of the branch or segment of the branch that subtends the node in Column 1. Column 3 identifies the ending time of the branch or segment of the branch that subtends the node in Column 1. Column 4 identifies the index of the event that occurs along the branch or segment of the branch that subtends the node in Column 1.
<code>tipStates</code>	A list of integer vectors. Each element is a single distinct shift configuration. For each tip the index of the event that occurs along the branch subtending the tip. Tips are ordered increasing here and elsewhere.
<code>tipLambda</code>	A list of numeric vectors. Each element is a single distinct shift configuration. For each tip is the average rate of speciation or trait evolution at the end of the terminal branch subtending that tip (averaged over all samples that are assignable to this particular distinct shift configuration).

tipMu	A list of numeric vectors. Each element is a single distinct shift configuration. For each tip the rate of extinction at the end of the terminal branch subtending that tip. Meaningless if working with BMM trait results.
type	A character string. Either "diversification" or "trait" depending on your BMM analysis.
downseq	An integer vector holding the nodes of phy. The order corresponds to the order in which nodes are visited by a pre-order tree traversal.
lastvisit	An integer vector giving the index of the last node visited by the node in the corresponding position in downseq. downseq and lastvisit can be used to quickly retrieve the descendants of any node. e.g. the descendants of node 89 can be found by <code>downseq[which(downseq==89):which(downseq==lastvisit[89])]</code> .

Author(s)

Dan Rabosky

References

bamm-project.org

See Also

[distinctShiftConfigurations](#), [plot.bammshifts](#), [summary.credibleShiftSet](#), [plot.credibleShiftSet](#), [getBranchShiftPriors](#)

Examples

```
data(events.whales)
data(whales)
data(prior.whales)
ed <- getEventData(whales, events.whales)

# Here we get the prior distribution on branch-specific shifts:
priordist <- getBranchShiftPriors(whales, prior.whales)

cset <- credibleShiftSet(ed, prior = priordist, BFcriterion = 5)

# Here is the total number of samples in the posterior:
length(ed$eventData)

# And here is the number of distinct shift configurations:
cset$number.distinct

# here is the summary statistics:
summary(cset)

# Accessing the raw frequency vector for the credible set:
cset$frequency

#The cumulative frequencies:
```

```

cset$cumulative

# The first element is the shift configuration with the maximum
# a posteriori probability. We can identify all the samples from
# posterior that show this shift configuration:

cset$indices[[1]]

# Now we can plot the credible set:
plot.credibleshiftset(cset, plotmax=4)

```

distinctShiftConfigurations

Identify distinct rate shift configurations

Description

Identify topologically distinct rate shift configurations that were sampled with BAMM, and assign each sample in the posterior to one of the distinct shift configurations.

Usage

```
distinctShiftConfigurations(ephy, prior, BFcriterion, ...)
```

Arguments

ephy	an object of class bammdata.
prior	an object of class branchprior
BFcriterion	Numeric value for the Bayes factor threshold used to identify branches with elevated shift probabilities relative to prior
...	Other arguments to distinctShiftConfigurations (possibly deprecated args)

Value

an object of class bammshifts. This is a list with the following components

marg.probs	A list of the marginal probability of a shift occurring at each node of the phylogeny for each distinct rate configuration.
bayesfactors	Bayes factor evidence for a rate shift on each branch
shifts	A list of the set of shift nodes for each distinct rate configuration.
samplesets	A list of sample indices that reduce to each of the unique shift sets.
frequency	A vector of frequencies of each distinct shift configuration .
coreshifts	A vector of node numbers corresponding to the core shifts. All of these nodes have a Bayes factor of at least BFcriterion supporting a rate shift.

BFcriterion A single numeric value giving the Bayes factor threshold used during enumeration of distinct shift configurations

Results are sorted by frequency: `$frequency[1]` gives the most common shift configuration sampled. `$shifts[[1]]` gives the corresponding node indices for that configuration. `$samplesets[[1]]` gives the indices of samples with this configuration.

Author(s)

Dan Rabosky

See Also

[plot.bammshifts](#), [credibleShiftSet](#)

Examples

```
data(whales, events.whales, prior.whales)

# Get the prior distribution on the number of shifts per branch:
bp <- getBranchShiftPriors(whales, prior.whales)

ed <- getEventData(whales, events.whales, burnin=0.25)

sc <- distinctShiftConfigurations(ed, bp, 10)

plot.bammshifts(sc, ed, rank=1)
```

dtRates	<i>Calculate macroevolutionary rate changes on a phylogeny from BAMB output</i>
---------	---

Description

`dtRates` calculates the mean of the marginal posterior density of the rate of speciation/extinction or trait evolution for small segments along each branch in a phylogeny.

Usage

```
dtRates(ephy, tau, ism = NULL, tmat = FALSE)
```

Arguments

ephy	an object of class <code>bamdata</code> .
tau	a numeric that specifies the size (as a fraction of tree height) of the segments that each branch will be discretized into.
ism	an integer vector indexing which posterior samples to include in the calculation.
tmat	a logical. If TRUE the matrix of branch segments is returned.

Details

dtRates bins the phylogeny into windows of time and calculates average rates of speciation/extinction or phenotypic evolution along each segment of a branch within a window. The width of each window is determined by tau. tau is a fraction of the root to tip distance so a value of tau = 0.01 bins the phylogeny into 100 time windows of equal width.

Value

A bammdata object with a new component named "dtrates", which is a list with two or three components:

tau	the parameter value of tau used in the calculation.
rates	if ephy\$type = "trait": a numeric vector with the phenotypic rates of each segment on each branch. if ephy\$type = "diversification": a list with two components. The first component is a numeric vector of speciation rates. The second component is a numeric vector of extinction rates.
tmat	a matrix of the starting and ending times of the segments on each branch. Only if tmat = TRUE.

Note

If there are zero length branches in the input tree NAs will result.

Author(s)

Mike Grudler

References

bamm-project.org

See Also

[plot.bammdata](#)

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25)

#use all posterior samples
ed <- dtRates(ed, tau=0.01)

#use specified range of posterior samples
ed <- dtRates(ed, tau=0.01, ism=50:150)
```

`getBestShiftConfiguration`*Get the best (sampled) rate shift configuration from a BAMM analysis*

Description

Get the rate shift configuration with the maximum a posteriori probability, e.g., the shift configuration that was sampled most frequently with BAMM.

Usage

```
getBestShiftConfiguration(x, prior, BFcriterion = 5)
```

Arguments

<code>x</code>	either a <code>bammdata</code> object or a <code>credibleshiftset</code> object
<code>prior</code>	An object of class <code>branchprior</code> . See getBranchShiftPriors
<code>BFcriterion</code>	The Bayes factor criterion for including nodes as core shifts in the enumeration of distinct shift configurations.

Details

This function estimates the rate shift configuration with the highest maximum a posteriori (MAP) probability. It returns a `bammdata` object with a single sample. This can be plotted with [plot.bammdata](#), and individual rate shifts can then be added with [addBAMMshifts](#).

The parameters of this object are averaged over all samples in the posterior that were assignable to the MAP shift configuration. All non-core shifts have been excluded, such that the only shift information contained in the object is from the "significant" rate shifts, as determined by the relevant Bayes factor criterion.

You can extract the same information from the credible set of shift configurations. See [credibleShiftSet](#) for more information.

Value

A class `bammdata` object with a single sample, corresponding to the diversification rate shift configuration with the maximum a posteriori probability. See [getEventData](#) for details.

Author(s)

Dan Rabosky

See Also

[getEventData](#), [credibleShiftSet](#), [plot.credibleshiftset](#), [plot.bammdata](#)

Examples

```

data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.1)

# Get prior distribution on shifts-per-branch:
data(prior.whales)
bp <- getBranchShiftPriors(whales, prior.whales)

# Pass the event data object in to the function:
best <- getBestShiftConfiguration(ed, bp, BFcriterion = 5)
plot.bammdata(best, lwd=2)
addBAMMshifts(best, cex=2)

# Now we can also work with the credible shift set:
css <- credibleShiftSet(ed, bp, BFcriterion = 5)

summary(css)

# examine model-averaged shifts from MAP configuration-
# This gives us parameters, times, and associated nodes
# of each evolutionary rate regime
# (note that one of them corresponds to the root)
css$eventData[[1]];

# Get bammdata representation of MAP configuration:
best <- getBestShiftConfiguration(css, bp, BFcriterion = 5);

plot.bammdata(best)
addBAMMshifts(best)

```

getBranchShiftPriors *Compute prior odds of a rate shift on each branch of a phylogeny from BMM output*

Description

Computes the prior probability of a rate shift event for each branch. These results are important for identifying topological rate shift locations on phylogenies with marginal probabilities that exceed those predicted under the prior alone.

Usage

```
getBranchShiftPriors(phy, priordata)
```

Arguments

phy	an object of class phylo
priordata	Filename for the BMM-generated output file containing a simulation of shifts under the prior. Alternatively, a dataframe containing this information.

Details

This function computes the prior odds on the distribution of numbers of rate shift events per branch under the prior. It returns an object of class 'branchprior', which is nothing more than a copy of the original phylogenetic tree but where each branch length has been replaced by the prior probability of a rate shift on each branch.

The significance of this function is that it lets us explicitly determine which branches have shift probabilities that are elevated relative to the prior expectation. The user does not work with the output of this function directly; it is passed to other functions [credibleShiftSet](#) and [distinctShiftConfigurations](#).

Value

A class `branchprior` with all the components of the original class `phylo` object, with the following changes:

`edge.length` branch lengths now represent the prior probability of a rate shift on each branch

Author(s)

Dan Rabosky

References

bamm-project.org

See Also

[distinctShiftConfigurations](#), [plot.bammshifts](#), [summary.credibleshiftset](#), [plot.credibleshiftset](#), [credibleShiftSet](#)

Examples

```
data(events.whales)
data(whales)
data(prior.whales)
ed <- getEventData(whales, events.whales)

# Get branch-specific cutoffs for identifying branches with
# elevated shift probabilities:
bprior <- getBranchShiftPriors(whales, prior.whales)
cset <- credibleShiftSet(ed, bprior)

# Here is the total number of samples in the posterior:
length(ed$eventData)
```

```

# And here is the number of distinct shift configurations:
cset$number.distinct

# here is the summary statistics:
summary(cset)

# Accessing the raw frequency vector for the credible set:
cset$frequency

#The cumulative frequencies:
cset$cumulative

# The first element is the shift configuration with the maximum
# a posteriori probability. We can identify all the samples from
# posterior that show this shift configuration:

cset$indices[[1]]

```

getCladeRates

Compute clade-specific mean rates

Description

Computes marginal clade-specific rates of speciation, extinction, or (if relevant) trait evolution from BAMM output.

Usage

```
getCladeRates(ephy, node = NULL, nodetype = "include", verbose = FALSE)
```

Arguments

ephy	an object of class bammdata
node	If computing rates for a specific portion of tree, the node subtending the relevant subtree
nodetype	Either "include" or "exclude". If nodetype = "include", the rates returned by the function will be for the subtree defined by node. If nodetype = "exclude", will compute mean rates for the tree after excluding the subtree defined by node.
verbose	If TRUE, will print the sample index as mean rates are computed for each sample from posterior. Potentially useful for extremely large trees.

Details

Computes the time-weighted mean evolutionary rate for a given clade. Conversely, one can compute the rate for a given phylogeny while excluding a clade; this operation will give the "background" rate. It is important to understand several aspects of these mean rates. First, rates in the BAMM framework are not constant through time. Hence, the function computes the mean time-integrated rates across the subtree. Operationally, this is done by integrating the speciation rate with respect to time along each branch in the subtree. These time-integrated rates are then summed, and the sum is divided by the total sum of branch lengths for the subtree.

The function computes a rate for each sample in the posterior, and returns a list of rate vectors. Each rate in the corresponding vector is a mean rate for a particular sample from the posterior. Hence, you can think of the return value for this function as an estimate of the marginal distribution of rates for the focal clade. You can compute means, medians, quantiles, etc from these vectors.

Value

A list with the following components:

lambda	A vector of speciation rates (if applicable), with the <i>i</i> 'th rate corresponding to the mean rate from the <i>i</i> 'th sample in the posterior
mu	A vector of extinction rates (if applicable), with the <i>i</i> 'th rate corresponding to the mean rate from the <i>i</i> 'th sample in the posterior
beta	A vector of phenotypic rates (if applicable), with the <i>i</i> 'th rate corresponding to the mean rate from the <i>i</i> 'th sample in the posterior

Author(s)

Dan Rabosky

References

bamm-project.org

Examples

```
data(events.whales)
data(whales)
ed <- getEventData(whales, events.whales)
all_rates <- getCladeRates(ed)

mean(all_rates$lambda)
mean(all_rates$mu)
# joint density of mean speciation and extinction rates:
plot(all_rates$mu ~ all_rates$lambda)

# clade specific rates: here for Dolphin subtree:
dol_rates <- getCladeRates(ed, node=140)
mean(dol_rates$lambda)
mean(dol_rates$mu)
```

getCohortMatrix	<i>Compute the pairwise correlation in rate regimes between all tips in a bammdata object</i>
-----------------	---

Description

Takes a bammdata object and computes the pairwise correlation in evolutionary rate regimes between all tips in the phylogeny. This can be used to identify cohorts of taxa that share common macroevolutionary rate parameters. It can also be used to construct a correlation matrix for GLS analyses using BAMM-estimated tip rates of speciation, extinction, or phenotypic evolution.

Usage

```
getCohortMatrix(ephy)
```

Arguments

ephy an object of class bammdata

Details

The cohort matrix is important for interpreting and visualizing macroevolutionary dynamics. Each entry $[i, j]$ of the cohort matrix is the probability that taxon i and taxon j share a common macroevolutionary rate regime. To compute this, we simply tabulate the percentage of samples from the posterior where taxon i and taxon j were placed in the same rate regime. If there is no rate heterogeneity in the dataset (e.g., the data are best explained by a single rate regime), then all species will tend to share the same rate regime and all values of the cohort matrix will approach 1.

A value of 0 between any two taxa means that at least one rate shift occurred on the nodal path connecting them in 100% of samples from the posterior. A value of 0.50 would imply that 50% of samples from the posterior included a rate shift on the path connecting taxa i and j . See below (examples) for an illustration of this.

Value

A numeric matrix of dimension $k \times k$, where k is the number of species in the phylogeny included in the bammdata object. Species names are included as row names and column names. The matrix is symmetric, such that the values for entry $[i, j]$ will mirror those for $[j, i]$.

Author(s)

Dan Rabosky

References

www.bamm-project.org

Examples

```

data(whales)
data(events.whales)
ed <- getEventData(whales, events.whales)

cormat <- getCohortMatrix(ed)

dim(cormat)
hist(cormat, breaks=50)

```

getEventData

Create bammdata object from MCMC output

Description

getEventData reads shift configuration data (the "event data" output) from a BMM analysis and creates a bammdata object. The bammdata object is fundamental for extracting information about macroevolutionary rate variation through time and among lineages.

Usage

```

getEventData(phy, eventdata, burnin = 0, nsamples = NULL,
  verbose = FALSE, type = "diversification")

```

Arguments

phy	an object of class phylo - specifically, the time calibrated tree that was analyzed with BMM.
eventdata	a character string specifying the path to a BMM event-data file. Alternatively, an object of class data.frame that includes the event data from a BMM run.
burnin	a numeric indicating the fraction of posterior samples to discard as burn-in.
nsamples	an integer indicating the number of posterior samples to include in the bammdata object. May be NULL.
verbose	a logical. If TRUE progress is outputted to the console. Defaults to FALSE.
type	a character string. Either "diversification" or "trait" depending on your BMM analysis.

Details

In the BMM framework, an "event" defines a macroevolutionary process of diversification or trait evolution. Every sample from the posterior includes at least one process, defined by such an "event". If a given sample includes just a single event, then the dynamics of diversification or trait evolution can be described entirely by a single time-constant or time-varying process that begins at the root of the tree. Any sample from the posterior distribution may include a complex mixture of distinct processes. To represent temporal heterogeneity in macroevolutionary rates, BMM models a rate R ,

e.g. speciation, as a function that changes exponentially with time: $R(t) = R(0) * exp(b * t)$. Here $R(0)$ is the initial rate and b is a parameter determining how quickly that rate grows or decays with time.

The `eventdata` file (or data frame) is a record of events and associated parameters that were sampled with BMM during simulation of the posterior with reversible jump MCMC. This complex, information-rich file is processed into a `bammdata` object, which serves as the core data object for numerous downstream analyses. From a `bammdata` object, you can summarize rate variation through time, among clades, extract locations of rate shifts, summarize clade-specific rates of speciation and extinction, and more.

In general, the user does not need to be concerned with the details of a `bammdata` object. The object is used as input by a number of `BMMtools` functions.

The parameter `nsamples` can be used to reduce the total amount of data included in the raw event-data output from a BMM run. The final `bammdata` object will consist of all data for `nsamples` from the posterior. These `nsamples` are equally spaced after discarding some burnin fraction as "burn-in". If `nsamples` is set to NULL, the `bammdata` object will include all samples in the posterior after discarding the burnin fraction.

Value

A list with many components:

<code>edge</code>	See documentation for class <code>phylo</code> in package <code>ape</code> .
<code>Nnode</code>	See documentation for class <code>phylo</code> in package <code>ape</code> .
<code>tip.label</code>	See documentation for class <code>phylo</code> in package <code>ape</code> .
<code>edge.length</code>	See documentation for class <code>phylo</code> in package <code>ape</code> .
<code>begin</code>	The beginning time of each branch in absolute time (the root is set to time zero)
<code>end</code>	The ending time of each branch in absolute time.
<code>numberEvents</code>	An integer vector with the number of events contained in <code>phy</code> for each posterior sample. The length of this vector is equal to the number of posterior samples in the <code>bammdata</code> object.
<code>eventData</code>	A list of dataframes. Each element is a single posterior sample. Each row in a dataframe holds the data for a single event. Data associated with an event are: <code>node</code> - a node number. This identifies the branch where the event originates. <code>time</code> - this is the absolute time on that branch where the event originates (with the root at time 0). <code>lam1</code> - an initial rate of speciation or trait evolution. <code>lam2</code> - a decay/growth parameter. <code>mu1</code> - an initial rate of extinction. <code>mu2</code> - a decay/growth parameter. <code>index</code> - a unique integer associated with the event. See 'Details'.
<code>eventVectors</code>	A list of integer vectors. Each element is a single posterior sample. For each branch in <code>phy</code> the index of the event that occurs along that branch. Branches are ordered increasing here and elsewhere.
<code>eventBranchSegs</code>	A list of matrices. Each element is a single posterior sample. Each matrix has four columns: Column 1 identifies a node in <code>phy</code> . Column 2 identifies the beginning time of the branch or segment of the branch that subtends the node in Column 1. Column 3 identifies the ending time of the branch or segment of

	the branch that subtends the node in Column 1. Column 4 identifies the index of the event that occurs along the branch or segment of the branch that subtends the node in Column 1.
tipStates	A list of integer vectors. Each element is a single posterior sample. For each tip the index of the event that occurs along the branch subtending the tip. Tips are ordered increasing here and elsewhere.
tipLambda	A list of numeric vectors. Each element is a single posterior sample. For each tip the rate of speciation or trait evolution at the end of the terminal branch subtending that tip.
tipMu	A list of numeric vectors. Each element is a single posterior sample. For each tip the rate of extinction at the end of the terminal branch subtending that tip. Meaningless if working with BAMM trait results.
meanTipLambda	For each tip the mean of the marginal posterior density of the rate of speciation or trait evolution at the end of the terminal branch subtending that tip.
meanTipMu	For each tip the mean of the marginal posterior density of the rate of extinction at the end of the terminal branch subtending that tip. Meaningless if working with BAMM trait results.
type	A character string. Either "diversification" or "trait" depending on your BAMM analysis.
downseq	An integer vector holding the nodes of phy. The order corresponds to the order in which nodes are visited by a pre-order tree traversal.
lastvisit	An integer vector giving the index of the last node visited by the node in the corresponding position in downseq. downseq and lastvisit can be used to quickly retrieve the descendants of any node. e.g. the descendants of node 89 can be found by <code>downseq[which(downseq==89):which(downseq==lastvisit[89])]</code> .

Note

Currently the function does not check for duplicate tip labels in phy, which may cause the function to choke.

Author(s)

Dan Rabosky, Mike Grudler

References

bamm-project.org

Rabosky, D. L. 2014. Automatic detection of key innovations, rate shifts, and diversity-dependence on phylogenetic trees. PLoS ONE 9:e89543

Rabosky, D. L., F. Santini, J. T. Eastman, S. A. Smith, B. L. Sidlauskas, J. Chang, and M. E. Alfaro. 2013. Rates of speciation and morphological evolution are correlated across the largest vertebrate radiation. Nature Communications DOI: 10.1038/ncomms2958

See Also

[summary.bammdata](#), [plot.bammdata](#), [dtRates](#).

Examples

```
data(primates)
data(events.primates)
xx <- getEventData(primates, events.primates, burnin=0.25, type = 'trait')

# compute mean phenotypic rate for primate body size evolution:
brates <- getCladeRates(xx)
mean(brates$beta)

# Plot rates:
plot.bammdata(xx)
```

```
getMarginalBranchRateMatrix
      compute mean branch rates for bammdata object
```

Description

For each sample in the posterior, computes the mean rate for each branch in the focal phylogeny (speciation, extinction, trait evolution). If the bammdata object contains *nsamples* samples and the target phylogeny has *nbranches* branches, the function will compute a matrix of *nbranches* x *nsamples*.

Usage

```
getMarginalBranchRateMatrix(ephy, verbose = FALSE)
```

Arguments

ephy	an object of class bammdata
verbose	print progress during processing of bammdata object

Details

If a type = 'diversification' bammdata object is passed as an argument, the function will return matrices for both speciation and extinction. If type = 'trait' object, the matrix will simply be the corresponding phenotypic rates. Branch-specific rates are the mean rates computed by integrating the relevant rate-through-time function along each branch, then dividing by the length of the branch.

Value

Returns a list with the following components:

```
lambda_branch_matrix
      A nbranches x nsamples matrix of mean speciation rates for each branch.
```

```
mu_branch_matrix      A nbranches x nsamples matrix of mean extinction rates for each branch.
beta_branch_matrix    A nbranches x nsamples matrix of mean phenotypic rates for each branch.
```

Author(s)

Dan Rabosky

Examples

```
data(whales)
data(events.whales)
ed <- getEventData(whales, events.whales, nsamples = 10)
mbr <- getMarginalBranchRateMatrix(ed)
dim(mbr$lambda_branch_matrix)
```

```
getMeanBranchLengthTree
```

Compute phylogeny with branch lengths equal to corresponding macroevolutionary rate estimates

Description

Takes a bammdata object and computes a phylogenetic tree where branch lengths are equal to the mean of the marginal distributions of rates on each branch. This tree can be plotted to visualize rate variation across a phylogeny.

Usage

```
getMeanBranchLengthTree(ephy, rate = "speciation")
```

Arguments

ephy	an object of class bammdata
rate	the type of rate-tree to be computed. Options: "speciation" (default), "extinction", "ndr" (net diversification), and "trait"

Value

A list with the following components:

phy	A phylogenetic tree, topologically identical to the model tree, but with branch lengths replaced by the mean (marginal) rates on each branch as estimated from the posterior samples in the bammdata object.
mean	The mean rate over all branches
median	the median rate over all branches

Author(s)

Dan Rabosky

References

www.bamm-project.org

See Also

[plot.bammdata](#)

Examples

```
data(whales)
data(events.whales)
ed <- getEventData(whales, events.whales)
ed2 <- subsetEventData(ed, index = 1:20)
ratetree <- getMeanBranchLengthTree(ed2, rate='speciation')
plot(ratetree$phy, show.tip.label=FALSE)
```

getmrca

Find most recent common ancestors

Description

Calculates the most recent common ancestor for each pair of tips. Used internally by [getEventData](#)

Usage

```
getmrca(phy, t1, t2)
```

Arguments

phy	an object of class phylo.
t1	a vector of mode integer or character corresponding to tips in phy.
t2	a vector of mode integer or character corresponding to tips in phy.

Details

Finds the most recent common ancestor for each pair of tips where pairs are defined as (t1[1], t2[1]), (t1[2], t2[2]), ... , (t1[i], t2[i]), ... ,(t1[n], t2[n]).

Value

A vector of node numbers of the common ancestor for each pair of tips.

Author(s)

Mike Grudler

getRateThroughTimeMatrix

Generate rate-through-time matrix from bammdata object

Description

Computes a matrix of macroevolutionary rates at specified timepoints from a bammdata object. These rates can be used for plotting speciation rates (and other rates) through time.

Usage

```
getRateThroughTimeMatrix(ephy, start.time = NULL,
end.time = NULL, nslices = 100, node = NULL, nodetype = "include")
```

Arguments

ephy	an object of class bammdata
start.time	the start time (in units before present) for the time sequence over which rates should be computed. If NULL, starts at the root
end.time	the end time (in units before present) for the time sequence over which rates should be computed. If NULL, ends in the present
nslices	The number of time points at which to compute rate estimates (between start.time and end.time)
node	Allows user to extract rate-through-time information for the subtree descended from a specific node. Alternatively, a specified subtree can be excluded from the rate matrix calculations.
nodetype	Two options: "include" and "exclude". If "include", computes rate matrix only for the descendants of subtree defined by node. If "exclude", computes rate matrix for all background lineages in tree after excluding the subtree defined by node. Ignored if node = NULL.

Details

Computes evolutionary rates for each sample in the posterior included as part of the bammdata object. Rates are computed by draping an imaginary grid over the phylogeny, where the grid begins at start.time and ends at end.time, with nslices vertical lines through the phylogeny. The mean rate at each point in time (for a given sample from the posterior) is simply the mean rate at that time for all branches that are intersected by the grid (see the grid plot in the examples section).

This function is used by [plotRateThroughTime](#), but the user can work directly with the `bamm-ratematrix` object for greater control in plotting rate-through-time trajectories for individual clades. See [examples](#) for an example of how this can be used to plot confidence intervals on a rate trajectory using shaded polygons.

The node options are particularly useful. If you have run BMM on a large phylogeny, you can easily generate the rate-through-time data for a particular subtree by specifying the node number along with `nodetype = "include"`. Likewise, if you want to look at just the background rate - excluding some particular lineage - just specify `nodetype = "exclude"`.

Value

An object of class `bamm-ratematrix` with the following components:

<code>lambda</code>	A <code>nsamples x nslices</code> matrix of speciation rates, where <code>nsamples</code> is the number of posterior samples in the <code>bammdata</code> object.
<code>mu</code>	A <code>nsamples x nslices</code> matrix of extinction rates
<code>beta</code>	A <code>nsamples x nslices</code> matrix of phenotypic rates (if applicable)
<code>times</code>	A vector of timepoints where rates were computed
<code>times</code>	A vector of timepoints where rates were computed (see example)
<code>type</code>	Either "diversification" or "trait", depending on the input data.

Author(s)

Dan Rabosky

See Also

[plotRateThroughTime](#)

Examples

```
##NOT RUN##
# Plot a rate-through-time curve with
# confidence intervals for the whale dataset:

# data(whales, events.whales)
# ed <- getEventData(whales, events.whales)

# rmat <- getRateThroughTimeMatrix(ed)

# plot.new()
# plot.window(xlim=c(0, 36), ylim=c(0, .7))

## Speciation quantiles: plot 90% CIs
# qq <- apply(rmat$lambda, 2, quantile, c(0.05, 0.5, 0.95))

# xv <- c(rmat$times, rev(rmat$times))
# yv <- c(qq[1,], rev(qq[3,]))

## Add the confidence polygon on rate distributions:
# polygon(xv, yv, col='gray80', border=FALSE)

## Add the median rate line:
# lines(rmat$times, qq[2,], lwd=3, col='red')

# Add axes
# axis(1, at=seq(-5, 35, by=5))
# axis(2, at=seq(-0.2, 1, by=0.2), las=1)

##### Now we will show the actual grid used for rate calculations:
```



```

# plot(whales, show.tip.label=FALSE)
# axisPhylo()

# mbt <- max(branching.times(whales))
## tvec <- mbt - rmat$times;
# tvec <- rmat$times;

# for (i in 1:length(tvec)){
#   lines(x=c(tvec[i], tvec[i]), y=c(0, 90), lwd=0.7, col='gray70')
# }

## This shows the grid of time slices over the phylogeny
##END##

```

getShiftNodesFromIndex

Identify nodes associated with rate shifts from bammdata object

Description

Find the node numbers associated with rate shifts for a specified sample from the posterior distribution contained in a bammdata object.

Usage

```
getShiftNodesFromIndex(ephy, index)
```

Arguments

ephy	a bammdata object
index	The index value of the posterior sample from which you want to identify shift nodes. This is <i>not</i> the same as the actual generation number of the MCMC sample. If your bammdata object contains 100 samples from the posterior distribution, the value of index must range from 1 to 100.

Value

A vector of nodes (excluding the root) that define branches on which shifts occurred for the specified sample from the posterior. Will return a numeric of length 0 if no non-root shifts occur in the specified sample.

Author(s)

Dan Rabosky

See Also

[addBAMMshifts](#), [plot.bammdata](#), [maximumShiftCredibility](#)

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales)

# Get the maximum shift credibility configuration:
msc <- maximumShiftCredibility(ed)

# GEt the nodes at which shifts occurred in the
# maximum shift credibility configuration:

getShiftNodesFromIndex(ed, index=msc$sampleindex)
```

getTipRates

Compute tip-specific macroevolutionary rates from bammdata object

Description

Return speciation, extinction, net diversification, or Brownian motion trait rates for all species in the phylogeny from BAMM output.

Usage

```
getTipRates(ephy, returnNetDiv = FALSE, statistic = 'mean')
```

Arguments

ephy	an object of class bammdata.
returnNetDiv	logical. If TRUE, then net diversification rates are returned, if FALSE, then both speciation and extinction rates are returned. If ephy is of type trait, then this is ignored.
statistic	logical: determines how the average tip rates should be calculated. Can be either mean or median.

Value

Returns a list with the following elements: If ephy type is 'diversification':

lambda	A matrix of tip speciation rates with species as rows, and posterior samples as columns.
mu	A matrix of tip extinction rates with species as rows, and posterior samples as columns.

`lambda.avg` A vector of average tip speciation rates, averaged with mean or median, depending on selected option for `statistic`. The vector is named with species names.

`mu.avg` A vector of average tip extinction rates, averaged with mean or median, depending on selected option for `statistic`. The vector is named with species names.

If ephy type is 'diversification' and `returnNetDiv = TRUE`:

`netdiv` A matrix of tip net diversification rates with species as rows, and posterior samples as columns.

`netdiv.avg` A vector of average tip net diversification rates, averaged with mean or median, depending on selected option for `statistic`. The vector is named with species names.

If ephy type is 'trait':

`beta` A matrix of tip phenotypic rates with species as rows, and posterior samples as columns.

`beta.avg` A vector of average tip phenotypic rates, averaged with mean or median, depending on selected option for `statistic`. The vector is named with species names.

Author(s)

Pascal Title

See Also

Requires an object of class `bammdata` as obtained with [getEventData](#).

Examples

```
data(whales, events.whales)
ephy <- getEventData(whales, events.whales, burnin=0.25)

#return a vector of average species-specific speciation rates.
meanlam <- getTipRates(ephy, returnNetDiv = FALSE, statistic = 'mean')$lambda.avg
meanlam

#return a vector of median species-specific net diversification rates.
ndr <- getTipRates(ephy, returnNetDiv = TRUE, statistic = 'median')$netdiv.avg

#Return mean species-specific speciation rates from all posterior
# samples in the \code{bamm-data} object.
lam <- getTipRates(ephy, returnNetDiv = FALSE, statistic = 'mean')$lambda
rowMeans(lam)
```

```
maximumShiftCredibility
```

Estimate maximum shift credibility configuration

Description

This is one estimate of the "best" rate shift configuration, considering only those shift configurations that were actually sampled using BAMM's reversible jump MCMC simulator. This is analogous to the "maximum clade credibility tree" from a Bayesian phylogenetic analysis. It is not necessarily the same as the shift configuration with the maximum a posteriori probability.

Usage

```
maximumShiftCredibility(ephy, maximize = "product")
```

Arguments

ephy	an object of class BAMMdata
maximize	Maximize the marginal probability of the product or sum of branch-specific shifts

Details

This is one point estimate of the overall "best" rate shift configuration. Following an MCMC simulation, the marginal shift probabilities on each individual branch are computed using [marginal-ShiftProbsTree](#). The shift configuration that maximizes the product (or sum, if specified) of these marginal branch-specific shift probabilities is the *maximum shift credibility configuration*.

This option is only recommended if you have no clear "winner" in your credible set of shift configurations (see [credibleShiftSet](#)). If you have a number of largely-equiprobable shift configurations in your 95% credible set, you may wish to try this function as an alternative for identifying a single best shift configuration. Otherwise, it is recommended that you present the shift configuration with the maximum a posteriori probability (see [getBestShiftConfiguration](#)).

Value

A list with the following components:

bestconfigs	A vector of the index values of MCMC samples with shift configurations equal to the maximum. Usually, more than one state sampled during the MCMC simulation will have an identical (maximized) marginal probability. All samples given in this vector will have an identical shift configuration.
scores	The optimality score (product or sum of marginal shift probabilities) for all sampled shift configurations in the BAMMdata object
optimalityType	Whether the product or sum of marginal shift probabilities was used to compute the maximum shift credibility configuration
sampleindex	A representative sample that is equal to the maximum shift credibility configuration (e.g., this can be plotted with addBAMMshifts)

Author(s)

Dan Rabosky

See Also

[marginalShiftProbsTree](#), [addBAMMshifts](#), [cumulativeShiftProbsTree](#), [credibleShiftSet](#), [getBestShift-Configuration](#)

Examples

```
data(whales)
data(events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25)
best_config <- maximumShiftCredibility(ed)
plot.bammdata(ed)
addBAMMshifts(ed, method='phylogram', index=best_config$sampleindex)
```

plot.bammdata

*Plot BAMM-estimated macroevolutionary rates on a phylogeny***Description**

plot.bammdata plots a phylogenetic tree from a bammdata object and colors each branch by the estimated rate of speciation, extinction, or trait evolution. Rates are not assumed to be constant in time, and the function can plot continuously-varying rates along individual branches.

Usage

```
## S3 method for class 'bammdata'
plot(x, tau = 0.01, method = "phylogram",
     vtheta = 5, rbf = 0.001, show = TRUE, labels = FALSE,
     legend = FALSE, spex="s", lwd = 1, cex = 1, pal = "RdYlBu",
     mask = integer(0), mask.color = gray(0.5), colorbreaks = NULL, logcolor = FALSE,
     par.reset = TRUE, direction = "rightwards", ...)
```

Arguments

x	an object of class bammdata.
tau	a numeric indicating the grain size for the calculations. See documentation for dtRates.
method	a character string indicating the method for plotting the phylogenetic tree. method = "phylogram" (default) plots the phylogenetic tree using rectangular coordinates. method = "polar" plots the phylogenetic tree using polar coordinates.
vtheta	a numeric indicating the angular separation (in degrees) of the first and last terminal nodes. Ignored if method = "phylogram".
rbf	a numeric indicating the length of the root branch as a fraction of total tree height. Ignored if method = "phylogram".

show	a logical indicating whether or not to plot the tree. Defaults to TRUE.
labels	a logical indicating whether or not to plot the tip labels. Defaults to FALSE.
legend	a logical indicating whether or not to plot a legend for interpreting the mapping of evolutionary rates to colors. Defaults to FALSE.
spex	a character string indicating what type of macroevolutionary rates should be plotted. "s" (default) indicates speciation rates, "e" indicates extinction rates, and any other character, e.g. "se", indicates net diversification rates. Ignored if <code>ephy\$type = "trait"</code> .
lwd	a numeric specifying the line width for branches.
cex	a numeric specifying the size of tip labels.
pal	a character string or vector of mode character that describes the color palette. See Details for explanation of options.
mask	an optional integer vector of node numbers specifying branches that will be masked with <code>mask.color</code> when plotted.
mask.color	the color for the mask.
colorbreaks	a numeric vector of percentiles delimiting the bins for mapping rates to colors. If NULL (default) bins are calculated from the rates that are passed with the <code>bammdata</code> object.
logcolor	should colors be plotted on a log scale.
par.reset	a logical indicating whether or not to reset the graphical parameters when the function exits. Defaults to TRUE. If you notice weird placement of points or labels after the function exits retry with this argument set to FALSE.
direction	a character string. Options are "rightwards", "leftwards", "upwards", and "downwards", which determine the orientation of the tips when the phylogeny plotted.
...	further arguments passed to <code>par</code> .

Details

To calculate rates, each branch of the phylogeny is discretized into a number of small segments, and the mean of the marginal posterior density of the rate of speciation/extinction or trait evolution is calculated for each such segment. Rates are mapped to colors such that cool colors represent slow rates and warm colors represent fast rates. When the tree is plotted each of these small segments is plotted so that changes in rates through time and shifts in rates are visible as gradients of color. The `spex` argument determines the type of rate that will be calculated. `spex = "s"` will plot speciation rates, `spex = "e"` will plot extinction rates, and `spex = "se"` will plot diversification rates (speciation - extinction). Note that if `x$type = "trait"` the `spex` argument is ignored and rates of phenotypic evolution are plotted instead. If `legend = TRUE` the function will plot a legend with the (kernel density estimated) distribution of rates. This contains the mapping of colors to numerical values as well as a comparison of their frequencies.

A number of color palettes come built in with BMMtools. Color-blind friendly options include

BrBG
PiYG
PRGn

PuOr
 RdBu
 RdYlBu
 BuOr
 BuOrRd
 DkRdBu
 BuDkOr
 GnPu

Some color-blind unfriendly options include

RdYlGn
 Spectral
 temperature
 terrain

For more information about these color palettes visit colorbrewer2.org and geography.uoregon.edu/datagraphics/color_scales.htm or use the help files of the R packages `RColorBrewer` and `dichromat`.

Additionally, any vector of valid named colors may also be used. The only restriction is that the length of this vector be greater than or equal to three (you can provide a single color, but in this case the entire tree will be assigned the same color). The colors should be ordered from cool to warm as the colors will be mapped from low rates to high rates in the order supplied (e.g. `pal=c("darkgreen","yellow2","red")`). The option `pal = "temperature"` uses the `rich.colors` function written by Arni Magnusson for the R package `gplots`.

Internally `plot.bammdata` checks whether or not rates have been calculated by looking for a component named "dtrates" in the `bammdata` object. If rates have not been calculated `plot.bammdata` calls `dtRates` with `tau`. Specifying smaller values for `tau` will result in smoother-looking rate changes on the tree. Note that smaller values of `tau` require more computation. If the `colorbreaks` argument is `NULL` a map of rates to colors is also made by calling `assignColorBreaks` with `NCOLORS = 64`. A user supplied `colorbreaks` argument can be passed as well. This allows one to plot parts of a tree while preserving the map of rates to colors that was made using rates for the entire tree.

If `plot.bammdata` is called repeatedly with the same `bammdata` object, computation can be reduced by first calling `dtRates` in the global environment.

Value

Returns (invisibly) a list with three components.

<code>coords</code>	A matrix of plot coordinates. Rows correspond to branches. Columns 1-2 are starting (x,y) coordinates of each branch and columns 3-4 are ending (x,y) coordinates of each branch. If <code>method = "polar"</code> a fifth column gives the angle (in radians) of each branch.
<code>colorbreaks</code>	A vector of percentiles used to group macroevolutionary rates into color bins.

colordens A matrix of the kernel density estimates (column 2) of evolutionary rates (column 1) and the color (column 3) corresponding to each rate value.

Note

If arguments to `par` are specified in ... the function resets `par` variables before returning unless `multi = TRUE`.

Author(s)

Mike Grundler

Source

[colorbrewer2.org, geography.uoregon.edu/datagraphics/color_scales.htm](http://colorbrewer2.org,geography.uoregon.edu/datagraphics/color_scales.htm)

See Also

[dtRates](#), [addBAMMshifts](#), [assignColorBreaks](#), [subtreeBAMM](#), [colorRampPalette](#)

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25)

#the first call to plot.bammdata. No calculations or assignments of rates have been made
plot(ed, lwd = 3, spex = "s") #internally calls dtRates and assignColorBreaks

##NOT RUN##
## now plot.bammdata no longer calls dtRates
# ed <- dtRates(ed, tau = 0.01)
# xx <- plot(ed, lwd = 3, spex = "s")
#
## you can plot subtrees while preserving the original
## rates to colors map by passing the colorbreaks object as an argument
# sed <- subtreeBAMM(ed, node = 103)
# plot(sed, lwd = 3, colorbreaks = xx$colorbreaks)
# sed <- subtreeBAMM(ed, node = 140)
# plot(sed, lwd = 3, colorbreaks = xx$colorbreaks)
## note how if we do not pass colorbreaks the map is
## no longer relative to the rest of the tree and the plot is quite
## distinct from the original
# plot(sed, lwd = 3)
#
## if you want to change the value of tau and the rates to colors map for the entire tree
# ed <- dtRates(ed, tau = 0.002)
# xx <- plot(ed, lwd = 3, spex = "s")
## now you can re-plot the subtrees using this finer tau partition
# sed <- subtreeBAMM(ed, node = 103)
# sed <- dtRates(sed, 0.002)
# plot(sed, lwd = 3, colorbreaks = xx$colorbreaks)
# sed <- subtreeBAMM(ed, node = 140)
```



```

# sed <- dtRates(sed, 0.002)
# plot(sed, lwd = 3, colorbreaks = xx$colorbreaks)
#
## multi-panel plotting and adding shifts of specific posterior samples
# par(mfrow=c(2,3),mar=c(5,1,1,1))
# samples <- sample(1:length(ed$eventData), 6)
# ed <- dtRates(ed, 0.005)
# xx <- plot(ed, show=FALSE) #individual plots will have a color map relative to the mean
# for (i in 1:6) {
#   ed <- dtRates(ed, 0.005, samples[i])
#   plot(ed, par.reset=FALSE, colorbreaks=xx$colorbreaks)
#   addBAMMshifts(ed,index=samples[i],method="phylogram",par.reset=FALSE)
# }
# dev.off()
#
## color options
# ed <- dtRates(ed,0.01)
# plot(ed, pal="temperature",lwd=3)
# plot(ed, pal="terrain",lwd=3)
# plot(ed, pal=c("darkgreen","yellow2","red"),lwd=3)
# plot(ed,method="polar",pal="Spectral", lwd=3)
# plot(ed,method="polar",pal="RdYlBu", lwd=3)
##END##

```

plot.bammshifts

Plot distinct rate shift configurations on a phylogeny

Description

Plots a random distinct rate shift configuration sampled by BAMM on a phylogeny.

Usage

```

## S3 method for class 'bammshifts'
plot(x, ephy, method="phylogram",pal="RdYlBu",
     rank=NULL, index=NULL, spex="s",
     legend=TRUE, add.freq.text=TRUE,...)

```

Arguments

x	an object of class <code>bammshifts</code> .
ephy	an object of class <code>bammdata</code> .
method	a character string for which plotting method to use. "phylogram" uses rectangular coordinates. "polar" uses polar coordinates.
pal	the color palette to use in <code>plot.bammdata</code> .
rank	the rank of the core shift configuration to plot. For the default (NULL) a random configuration is chosen.
index	the posterior sample to plot. For the default (NULL) a random sample is chosen.

<code>spex</code>	a character string indicating what type of macroevolutionary rates should be plotted. "s" (default) indicates speciation rates, "e" indicates extinction rates, and any other character, e.g. "se", indicates net diversification rates. Ignored if <code>ephy\$type = "trait"</code> .
<code>legend</code>	a logical indicating whether to plot a legend.
<code>add.freq.text</code>	a logical indicating whether the frequency of each sampled shift configuration should be added to each plot.
<code>...</code>	other arguments to <code>plot.bammdata</code> .

Details

A rate shift configuration is the set of nodes of the phylogeny where a shift occurs in the macroevolutionary rate dynamic of diversification or trait evolution. Each posterior sample is a potentially distinct rate shift configuration. Different configurations may imply different macroevolutionary scenarios. This function helps visualize the different distinct rate shift configurations sampled by BAMM.

A core shift configuration is defined by a set of nodes that have shift probabilities that are substantially elevated relative to what you expect under the prior alone. These core configurations are inferred in `distinctShiftConfigurations`. It is almost certain that more than one core shift configuration will be sampled by BAMM. Moreover, each core shift configuration may contain many subconfigurations. A subconfiguration contains the core shift node configuration and zero or more additional shift nodes that occur with low marginal probability.

Points are added to the branches subtending the nodes of each rate configuration. The size of the point is proportional to the marginal probability that a shift occurs on a specific branch. If the instantaneous rate at a shift's origin represents an initial increase above the ancestral instantaneous rate the point is red. If the instantaneous rate at a shift's origin represents an initial decrease below the ancestral instantaneous rate the point is blue.

Author(s)

Mike Grundler, Dan Rabosky

References

bamm-project.org

See Also

[distinctShiftConfigurations](#), [plot.bammdata](#)

Examples

```
data(whales, events.whales, prior.whales)

ed <- getEventData(whales, events.whales, burnin=0.25)

# Here we get prior for identifying "significant" (core) rate shifts
priordist <- getBranchShiftPriors(whales, prior.whales)
```

```
sc <- distinctShiftConfigurations(ed, prior = priordist, BFcriterion = 5)

plot(sc, ed)
```

plot.credibleShiftset *Plot credible set of rate shift configurations from BAMM analysis*

Description

Plots the credible set of rate shift configurations from a BAMM analysis on a phylogeny

Usage

```
## S3 method for class 'credibleShiftset'
plot(x, plotmax=9, method="phylogram", pal="RdYlBu", spex="s", add.freq.text=TRUE,
use.plot.bammdata=TRUE, border=TRUE, legend=FALSE, send2pdf=FALSE, ...)
```

Arguments

x	an object of class credibleShiftset.
plotmax	an integer number of plots to display.
method	a coordinate method to use for plotting. Options are "phylogram" or "polar".
pal	a color palette to use with plot.bammdata.
spex	a character string indicating what type of macroevolutionary rates should be plotted. "s" (default) indicates speciation rates, "e" indicates extinction rates, and any other character, e.g. "se", indicates net diversification rates. Ignored if ephy\$type = "trait".
add.freq.text	a logical indicating whether to add the posterior frequency of each shift configuration to the plotting region.
use.plot.bammdata	a logical indicating whether to use plot.bammdata (TRUE) or plot.phylo (FALSE).
border	a logical indicating whether to frame the plotting region.
legend	a logical indicating whether to plot a legend.
send2pdf	a logical indicating whether to print the figure to a PDF file.
...	further arguments to pass to plot.bammdata.

Details

This produces phylorate plots for the plotmax most-probable shift configurations sampled with BAMM. Shift configurations are plotted in a single graphics window. The posterior probability (frequency) of each rate shift configuration in the posterior is shown (omitted with argument add.freq.text = FALSE).

Points are added to the branches subtending the nodes of each rate configuration. The size of the point is proportional to the marginal probability that a shift occurs on a specific branch. If the instantaneous rate at a shift's origin represents an initial increase above the ancestral instantaneous rate the point is red. If the instantaneous rate at a shift's origin represents an initial decrease below the ancestral instantaneous rate the point is blue.

Author(s)

Mike Grundler

Referencesbamm-project.org**See Also**[credibleShiftSet](#), [distinctShiftConfigurations](#), [plot.bammdata](#), [plot.bammshifts](#)**Examples**

```

data(events.whales)
data(whales)
ed <- getEventData(whales, events.whales)

# Get branch prior on number of shifts:
data(prior.whales)
bp <- getBranchShiftPriors(whales, prior.whales)
cset <- credibleShiftSet(ed, bp)
plot(cset)

```

plotRateThroughTime *Plot rates through time*

Description

Generates a plot of diversification or phenotypic rate through time with confidence intervals.

Usage

```

plotRateThroughTime(ephy, useMedian = TRUE,
  intervals=seq(from = 0,to = 1,by = 0.01),
  ratetype = 'auto', nBins = 100, smooth = FALSE, smoothParam = 0.20,
  opacity = 0.01, intervalCol='blue', avgCol='red', start.time = NULL,
  end.time = NULL, node = NULL, nodetype='include', plot = TRUE, cex.axis=1,
  cex.lab=1.3, lwd=3, xline=3.5, yline=3.5, mar=c(6,6,1,1), xticks=NULL,
  yticks=NULL, xlim='auto', ylim='auto',add=FALSE, axis.labels=TRUE)

```

Arguments

ephy	object of class bammdata or bamm-ratematrix
useMedian	a logical: will plot median if TRUE, mean if FALSE
intervals	if NULL, no intervals will be plotted, otherwise a vector of quantiles must be supplied (these will define shaded polygons)

ratetype	if 'auto', defaults to speciation (for diversification) or beta (for traits). Can alternatively specify 'netdiv' or 'extinction'.
nBins	number of time slices used to generate rates through time.
smooth	a logical: whether or not to apply loess smoothing.
smoothParam	loess smoothing parameter, ignored if smooth = FALSE.
opacity	opacity of color for interval polygons.
intervalCol	color for interval polygons.
avgCol	color for mean/median line (line will not be plotted if avgCol = NULL).
start.time	start time (in units before present). If NULL, starts at root.
end.time	end time (in units before present). If NULL, ends at present.
node	if supplied, the clade descended from this node will be used or ignored, depending on nodetype.
nodetype	if 'include', rates will be plotted only for the clade descended from node. If 'exclude', the clade descended from node will be left out of the calculation of rates.
plot	a logical: if TRUE, a plot will be returned, if FALSE, the data for the plot will be returned.
cex.axis	size of axis tick labels.
cex.lab	size of axis labels.
lwd	line width of the average rate.
xline	margin line for placing x-axis label.
yline	margin line for placing y-axis label.
mar	passed to par() to set plot margins.
xticks	number of ticks on the x-axis, automatically inferred if NULL.
yticks	number of ticks on the y-axis, automatically inferred if NULL.
xlim	vector of length 2 with min and max times for x axis. X axis is time since present, so if plotting till the present, xlim[2]==0. Can also be 'auto'.
ylim	vector of length 2 with min and max rates for y axis. Can also be 'auto'.
add	a logical: should rates be added to an existing plot.
axis.labels	a logical: if TRUE, axis labels will be plotted.

Details

If the input ephy object has been generated by [getEventData](#) and is of class `bammdata`, then `start.time`, `end.time`, `node` and `nodetype` can be specified. If the input ephy object has been generated by [getRateThroughTimeMatrix](#) and is of class `bamm-ratematrix`, then those arguments cannot be specified because they are needed to generate the rate matrix, which in this case has already happened.

The user has complete control of the plotting of the confidence intervals. Confidence intervals will not be plotted at all if `intervals=NULL`. If a single confidence interval polygon is desired, rather than overlapping polygons, then `intervals` can specify the confidence interval bounds, and `opacity` should be set to 1 (see examples).

If working with a large dataset, we recommend first creating a `bamm-ratematrix` object with [getRateThroughTimeMatrix](#) and then using that object as input for `plotRateThroughTime`. This way, the computation of rates has already happened and will not slow the plotting function down, making it easier to adjust plotting parameters.

Value

If `plot = FALSE`, then a list is returned with the following components:

<code>poly</code>	a list of matrices, where each matrix contains the coordinates that define each overlapping confidence interval polygon.
<code>avg</code>	a vector of y-coordinates for mean or median rates used to plot the average rates line.
<code>times</code>	a vector of time values, used as x-coordinates in this plot function.

Author(s)

Pascal Title

See Also

See [getEventData](#) and [getRateThroughTimeMatrix](#) to generate input data.

Examples

```
##NOT RUN##
# data(events.whales)
# data(whales)
# ephy <- getEventData(whales,events.whales)

## Simple plot of rates through time with default settings
# plotRateThroughTime(ephy)

## Plot two processes separately with 90% CI and loess smoothing
# plotRateThroughTime(ephy, intervals=seq(from=0,0.9,by=0.01), smooth=TRUE,
# node=141, nodetype='exclude', ylim=c(0,1.2))
#
# plotRateThroughTime(ephy, intervals=seq(from=0,0.9,by=0.01), smooth=TRUE,
# node=141, nodetype='include', add=TRUE, intervalCol='orange')
#
# legend('topleft',legend=c('Dolphins','Whales'),col='red',
# fill=c('orange','blue'),border=FALSE,lty=1,lwd=2,merge=TRUE,seg.len=0.6)
#
## Same plot, but from bamm-ratematrix objects
# rmat1 <- getRateThroughTimeMatrix(ephy, node=141, nodetype='exclude')
# rmat2 <- getRateThroughTimeMatrix(ephy, node=141, nodetype='include')
#
# plotRateThroughTime(rmat1, intervals=seq(from=0,0.9,by=0.01),
# smooth=TRUE, ylim=c(0,1.2))
#
# plotRateThroughTime(rmat2, intervals=seq(from=0,0.9,by=0.01),
```

```

# smooth=TRUE, add=TRUE, intervalCol='orange')
#
## To plot the mean rate without the confidence envelope
# plotRateThroughTime(ephy,useMedian=FALSE, intervals=NULL)
#
## To plot the mean rate, with a single 95% confidence envelope, grayscale
# plotRateThroughTime(ephy,useMedian=FALSE, intervals=c(0.05,0.95),
# intervalCol='gray70', avgCol='black', opacity=1)
#
## To not plot, but instead return the plotting data generated in this function,
## we can make plot=FALSE
# plotRateThroughTime(ephy, plot=FALSE)
##END##

```

samplingProbs

Creates clade-specific sampling fractions

Description

If the user would like to specify species sampling on a clade-by-clade basis, a sampling probability table can be provided to BAMM.

Usage

```
samplingProbs(tree, cladeTable, cladeRichness = NULL,
globalSampling, output, writeToDisk = TRUE)
```

Arguments

tree	an object of class phylo.
cladeTable	a dataframe with one column of species names and a second column of group assignment.
cladeRichness	either NULL or a vector of species counts, named by clade names.
globalSampling	percent sampling for the backbone of the phylogeny
output	path + output file name
writeToDisk	a logical, should the table be written to disk, defaults to TRUE.

Details

This function handles two types of input: The cladeTable can either contain the species found in the phylogeny, along with clade assignment of those species, or it can contain more species than found in the phylogeny. If the table only contains those species in the phylogeny, then a vector cladeRichness must be provided that contains known clade richness. If the cladeTable contains more than the species in the phylogeny, then cladeRichness should be set to NULL. The globalSampling value should represent the overall completeness of species sampling in terms of major clades. See bamm-project.org for additional details.

Value

If `writeToDisk = TRUE`, then no object is returned. If `writeToDisk = FALSE`, then a dataframe is returned. The resultant table must contain one row for each species in the phylogeny, along with clade assignment, and sampling fraction. The first line must contain the overall sampling fraction for the phylogeny and must be written as tab-delimited, with no headers.

Author(s)

Pascal Title

Examples

```
#Generate dummy data
tree <- read.tree(text="(((t1:2,(t2:1,t3:1):1):1,((t4:1,t5:1):1,t6:2):1):1,
(t7:3,(t8:2,t9:2):1):1);")
tree$tip.label <- paste(rep('Species',9),1:9,sep='')

spTable <- as.data.frame(matrix(nrow=9,ncol=2))
spTable[,1] <- tree$tip.label
spTable[1:3,2] <- 'cladeA'
spTable[4:6,2] <- 'cladeB'
spTable[7:9,2] <- 'cladeC'
richnessVec <- c(cladeA=5, cladeB=4, cladeC=12)

# Option 1: We have a table of clade assignment for the species in the tree,
# along with a vector of known clade richness
spTable
richnessVec
samplingProbs(tree, cladeTable = spTable, cladeRichness = richnessVec,
globalSampling = 1, writeToDisk = FALSE)

# Option 2: We have a table of known species, beyond the sampling in the phylogeny
spTable <- rbind(spTable, c('Species10','cladeA'),c('Species11','cladeA'),
c('Species12','cladeC'), c('Species13','cladeC'),c('Species14','cladeC'))

spTable

samplingProbs(tree, cladeTable = spTable, cladeRichness = NULL,
globalSampling = 0.9, writeToDisk = FALSE)
```

setBAMMpriors

setBAMMpriors

Description

Set priors for BAMM analysis

Usage

```
setBAMMpriors(phy, total.taxa = NULL, traits=NULL,
outfile = 'myPriors.txt', Nmax = 1000)
```

Arguments

phy	an object of class phylo, e.g., the phylogenetic tree you will analyze with BAMM
total.taxa	If doing speciation-extinction analysis, the total number of taxa in your clade. If your tree contains all taxa in the clade (100% sampling), then leave this as NULL
traits	A filename where the trait data (BAMM format) are stored. Leave NULL if doing a speciation-extinction analysis.
outfile	Filename for outputting the sample priors block.
Nmax	If analyzing a very large tree for phenotypic evolution, uses only this many taxa to estimate priors for your dataset. Avoid matrix inversion issues with large numbers of tips.

Details

This is a "quick and dirty" tool for identifying approximately acceptable priors for a BAMM analysis. We have found that choice of prior can have a substantial impact on BAMM analyses. It is difficult to simply set a default prior that applies across datasets, because users often have trees with branch lengths in very different units (e.g., numbers of substitutions versus millions of years). Hence, without some careful attention, you can inadvertently specify some very bad prior distributions. This function is designed to at least put you in the right ballpark for decent prior distributions, but there are no guarantees that these are most appropriate for your data.

The general rules applied here are:

For the `lambdaInitPrior`, we estimate the speciation rate of the data under a pure birth model. We then set this prior to give an exponential distribution with a mean five times greater than this computed pure birth speciation estimate.

The `lambdaShiftPrior` is the standard deviation of the normal prior on the exponential change parameter k . We set the prior distribution based on the age of the root of the tree. We set the standard deviation of this distribution such that 2 standard deviations give a parameter that will yield a 90% decline in the initial speciation rate between the root of the tree and the tips. The basic model is $\lambda(t) = \lambda_0 * \exp(k * t)$. This is a straightforward calculation: let $x = -\log(0.1) / TMAX$, where $TMAX$ is the age of the tree. Then set the standard deviation equal to $(x / 2)$.

We set `muInitPrior` equal to `lambdaInitPrior`.

For trait evolution, we first compute the maximum likelihood estimate of the variance of a Brownian motion process of trait evolution. The prior `betaInitPrior` is then set to an exponential distribution with a mean 5 times greater than this value (similar to what is done for `lambda` and `mu`, above).

This function generates an output file containing a prior parameters block that can be pasted directly into the priors section of your BAMM control file.

Value

The function does not return anything. It simply performs some calculations and writes formatted output to a file.

Author(s)

Dan Rabosky

ShiftProbsTree	<i>Branch-specific rate shift probabilities</i>
----------------	---

Description

`marginalShiftProbsTree` computes a version of a phylogenetic tree where each branch length is equal to the marginal probability that a shift occurred on a particular branch. The `cumulativeShiftProbsTree` includes the cumulative probability that a shift occurred on a given branch. See details.

Usage

```
cumulativeShiftProbsTree(ephy)
marginalShiftProbsTree(ephy)
```

Arguments

`ephy` An object of class `bammdata`

Details

The *marginal shift probability tree* is a copy of the target phylogeny, but where each branch length is equal to the branch-specific marginal probability that a rate-shift occurred on the focal branch. For example, a branch length of 0.333 implies that 1/3 of all samples from the posterior had a rate shift on the focal branch.

Note: It is highly inaccurate to use marginal shift probabilities as a measure of whether diversification rate heterogeneity occurs within a given dataset. Consider the following example. Suppose you have a tree with topology (A, (B, C)). You find a marginal shift probability of 0.5 on the branch leading to clade C, and also a marginal shift probability of 0.5 on the branch leading to clade BC. Even though the marginal shift probabilities appear low, it may be the case that the joint probability of a shift occurring on *either* the branch leading to C or BC is 1.0. Hence, you could be extremely confident (posterior probabilities approaching 1.0) in rate heterogeneity, yet find that no single branch has a particularly high marginal shift probability. In fact, this is exactly what we expect in most real datasets, because there is rarely enough signal to strongly support the occurrence of a shift on any particular branch.

The *cumulative shift probability tree* is a copy of the target phylogeny but where branch lengths are equal to the cumulative probability that a rate shift occurred somewhere on the path between the root and the focal branch. A branch length equal to 0.0 implies that the branch in question has evolutionary rate dynamics that are shared with the evolutionary process starting at the root of the tree. A branch length of 1.0 implies that, with posterior probability 1.0, the rate dynamics on a branch are decoupled from the "root process".

Value

An object of class `phylo`, but with branch lengths equal to the marginal or cumulative shift probabilities.

Author(s)

Dan Rabosky

References

bamm-project.org

See Also

[maximumShiftCredibility](#)

Examples

```
data(whales)
data(events.whales)
ed <- getEventData(whales, events.whales)

# computing the marginal shift probs tree:
mst <- marginalShiftProbsTree(ed)

# The cumulative shift probs tree:
cst <- cumulativeShiftProbsTree(ed)

#compare the two types of shift trees side-by-side:
plot.new()
par(mfrow=c(1,2))
plot.phylo(mst, no.margin=TRUE, show.tip.label=FALSE)
plot.phylo(cst, no.margin=TRUE, show.tip.label=FALSE)
```

`speciesByRatesMatrix` *Compute species-specific rate through time trajectories*

Description

Computes the mean of the marginal posterior density of speciation/extinction or phenotypic rates for equally spaced points along the root to tip path for each species

Usage

```
speciesByRatesMatrix(ephy, nslices, index, spex)
```

Arguments

ephy	an object of class bammdata
nslices	an integer number of time slices. This determines the number of equally spaced points in time at which rates are computed for each species
index	an integer or vector of mode integer indicating which posterior samples to use in the calculation. If NULL (default) all samples are used
spex	a character string. "s" (default) calculates speciation rates; "e" calculates extinction rates; "se" calculates diversification rates. Ignored if ephy\$type = "trait"

Value

A list with two components:

times	A vector of time points where rates were calculated
rates	A species X times matrix of rate through time trajectories

Author(s)

Mike Grundler

References

bamm-project.org

See Also

[getRateThroughTimeMatrix](#)

Examples

```
data(whales, events.whales)
ed <- getEventData(whales,events.whales,burnin=0.25)
ratemat <- speciesByRatesMatrix(ed, nslices = 100)

dolphins <- extract.clade(whales,140)$tip.label
plot.new()
plot.window(xlim=c(0,35),ylim=c(0,0.8))
for (i in 1:nrow(ratemat$rates)) {
  if (whales$tip.label[i] %in% dolphins) {
    lines(ratemat$times, ratemat$rates[i,], lwd=2, col=4)
  }
  else {
    lines(ratemat$times, ratemat$rates[i,], lwd=2, col=8)
  }
}
axis(1,seq(-5,35,5))
axis(2,seq(-0.2,0.8,0.2),las=1)
mtext("Time since root",1,line=2.5)
mtext("Speciation rate",2,line=2.5)
```

subsetEventData	<i>Subset a bammdata object</i>
-----------------	---------------------------------

Description

Subsets a bammdata object. Returns a bammdata object after extracting a specified set of samples from the posterior.

Usage

```
subsetEventData(ephy, index)
```

Arguments

ephy	an object of class bammdata
index	A vector of integers corresponding to samples to be extracted from the posterior distribution of shift configurations included in the bammdata object.

Details

This will result in an error if you attempt to access samples that do not exist in the ephy data object. For example, if your bammdata object includes 100 samples from a posterior distribution sampled with BAMM, you can only attempt to subset with index values 1:100.

Value

A bammdata object, but only containing index samples from the posterior in the original object.

Author(s)

Dan Rabosky

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales)
ed2 <- subsetEventData(ed, index=1)
plot.bammdata(ed2)
addBAMMshifts(ed2, cex=2)
```

subtreeBAMM *Pulls out a subtree from bammdata object*

Description

Given a set of tips or a node, this function extracts the corresponding subtree from the bammdata object. User should specify either a set of tips or a node, and the node will overwrite the tips if both are given.

Usage

```
subtreeBAMM(ephy, tips=NULL, node=NULL)
```

Arguments

ephy	an object of class bammdata.
tips	an integer or character vector indicating which tips (more than one) to be included in the subtree.
node	an integer indicating the root of the subtree to be extracted, and it must correspond to an innernode on the tree.

Details

This function allows users to extract a subtree from a big bammdata object, and examine the subset using [plot.bammdata](#)

Author(s)

Huateng Huang

See Also

[getmrca](#), [plot.bammdata](#)

Examples

```
data(whales, events.whales)
ephy <- getEventData(whales, events.whales, burnin=0.25)

#specify a set of tips for the subtree
tips <- sample(ephy$tip.label, size=20, replace=FALSE)
subphy <- subtreeBAMM(ephy, tips=tips)

#specify a innernode for subsetting
subphy <- subtreeBAMM(ephy, node=103)

#plot the subtree
plot(subphy)
```

summary.bammdata	<i>Summary of rate shift results from BMM analysis.</i>
------------------	---

Description

Summarizes the posterior distribution on the number of shifts.

Usage

```
## S3 method for class 'bammdata'  
summary(object, display = 10, print=T, ...)
```

Arguments

object	an object of class bammdata.
display	an integer for the number of rows of the posterior to display.
print	Print summary of shift distribution in console window?
...	additional arguments (unused).

Details

Prints to console the number of posterior samples and the posterior distribution on the number of shifts, which is just the fraction of samples in the posterior having 0, 1, 2,...n shifts.

Value

Returns (invisibly) a dataframe with 2 components:

shifts	the number of shifts
prob	the corresponding posterior probability of a model with a given number of rate shifts.

Author(s)

Mike Grundler, Dan Rabosky

References

bamm-project.org

Examples

```
data(whales, events.whales)  
ephy <- getEventData(whales,events.whales,nsamples=100)  
summary(ephy)
```

summary.branchprior *Summary of class "branchprior" object.*

Description

Prints summary attributes of the branchprior object

Usage

```
## S3 method for class 'branchprior'  
summary(object, ...)  
## S3 method for class 'branchprior'  
print(x, ...)
```

Arguments

object, x an object of class branchprior.
... additional arguments (unused).

Author(s)

Dan Rabosky

References

bamm-project.org

See Also

[distinctShiftConfigurations](#), [plot.bammshifts](#), [credibleShiftSetgetBranchShiftPriors](#)

summary.credibleshiftset

Summary of credible set of shift configurations from a BMM analysis.

Description

Prints summary attributes of the BMM credible set of shift configurations.

Usage

```
## S3 method for class 'credibleshiftset'  
summary(object, ...)  
## S3 method for class 'credibleshiftset'  
print(x, ...)
```


Arguments

object, x an object of class credibleShiftset.
... additional arguments (unused).

Details

Prints to console summary attributes of the XX% credible set of shift configurations sampled using BMM. Attributes printed include: the number of distinct configurations in the XX% credible set and the posterior probability, cumulative probability, and number of rate shifts in the 9 most-probable shift configurations.

Value

summary.credibleShiftset returns (invisibly) a dataframe with a number of rows equal to the number of shift configurations in the credible set and four columns:

rank	The ranked index of each shift configuration (ranked by posterior probability)
probability	the posterior probability of each shift configuration
cumulative	the cumulative probability of each shift configuration
N_shifts	the number of rate shifts in each shift configuration (can be zero).

Author(s)

Dan Rabosky

References

bamm-project.org

See Also

[distinctShiftConfigurations](#), [plot.bammshifts](#), [credibleShiftSet](#)

Examples

```
data(events.whales)
data(whales)
ed <- getEventData(whales, events.whales)

data(prior.whales)
bp <- getBranchShiftPriors(whales, prior.whales)
cset <- credibleShiftSet(ed, bp)
summary(cset)
```

 testTimeVariableBranches

Evaluate evidence for temporal rate variation across tree

Description

For each branch in a phylogenetic tree, evaluates the evidence (posterior probability or Bayes factor) that macroevolutionary rates have varied through time

Usage

```
testTimeVariableBranches(ephy, prior_tv = 0.5, return.type = "posterior")
```

Arguments

ephy	an object of class bammdata.
prior_tv	The prior probability that rate shifts lead to a new time-varying rate process (versus a time-constant process)
return.type	either "posterior" or "bayesfactor", depending on which form of evidence you would like.

Details

In BMM 2.0, rate shifts on trees can lead to time-varying or constant-rate diversification processes. In other words, the model will incorporate temporal variation in rates only if there is sufficient evidence in the data to favor it. The function `testTimeVariableBranches` enables the user to extract the evidence in favor of time-varying rates on any branch of a phylogenetic tree from a `bammdata` object.

The function returns a copy of the original phylogenetic tree, but where branch lengths have been replaced by either the posterior probability (`return.type = "posterior"`) or the Bayes factor evidence (`return.type = "bayesfactor"`) that the macroevolutionary rate regime governing each branch is time-variable. Consider a particular branch X on a phylogenetic tree. If the length of this branch is 0.97 and `return.type = "posterior"`, this implies that branch X was governed by a time-varying rate dynamic in 97% of all samples in the posterior. Alternatively, only 3% of samples specified a constant rate dynamic on this branch.

The function also provides an alternative measure of support if `return.type = "bayesfactor"`. In this case, the Bayes factor evidence for temporal rate variation is computed for each branch. We simply imagine that diversification rates on each branch can be explained by one of two models: either rates vary through time, or they do not. In the above example (branch X), the Bayes factor would be computed as follows, letting *Prob_timevar* and *Prior_timevar* be the posterior and prior probabilities that a particular branch is governed by a time-varying rate process:

$$(Prob_timevar) / (1 - Prob_timevar) * (1 - prior_timevar) / (prior_timevar)$$

The Bayes factor is not particularly useful under uniform prior odds (e.g., `prior_tv = 0.5`), since this simply reduces to the ratio of posterior probabilities. Note that the prior must correspond

to whatever you used to analyze your data in BMM. By default, time-variable and time-constant processes are assumed to have equal prior odds.

This function can be used several ways, but this function allows the user to quickly evaluate which portions of a phylogenetic tree have "significant" evidence for rate variation through time (see examples below).

Value

An object of class `phylo`, but where branch lengths are replaced with the desired evidence (posterior probability or Bayes factor) that each branch is governed by a time-varying rate dynamic.

Author(s)

Dan Rabosky

Examples

```
# Load whale data:
data(whales)
data(events.whales)
ed <- getEventData(whales, events.whales, burnin=0.1, nsamples=200)

# compute the posterior probability of
# time-varying rates on each branch
tree.pp <- testTimeVariableBranches(ed)

# Plot tree, but color all branches where the posterior
# probability of time-varying rates exceeds 95%:

colvec <- rep("black", nrow(whales$edge))
colvec[tree.pp$edge.length >= 0.95] <- 'red'

plot.phylo(whales, edge.color=colvec, cex=0.5)

# now, compute Bayes factors for each branch:

tree.bf <- testTimeVariableBranches(ed, return.type = "bayesfactor")

# now, assume that our prior was heavily stacked in favor
# of a time-constant process:
tree.bf2 <- testTimeVariableBranches(ed, prior_tv = 0.1, return.type = "bayesfactor")

# Plotting the branch-specific Bayes factors against each other:

plot.new()
par(mar=c(5,5,1,1))
plot.window(xlim=c(0, 260), ylim=c(0, 260))
points(tree.bf2$edge.length, tree.bf$edge.length, pch=21, bg='red', cex=1.5)
axis(1)
axis(2, las=1)
mtext(side=1, text="Bayes factor: prior_tv = 0.1", line=3, cex=1.5)
mtext(side = 2, text = "Bayes factor: uniform prior odds", line=3, cex=1.5)
```

```
# and you can see that if your prior favors CONSTANT RATE dynamics
# you will obtain much stronger Bayes factor support for time varying rates
# IF the evidence is present in your data to support time variation.
# To be clear, the Bayes factors in this example were computed from the
# same posterior probabilities: it is only the prior odds that differed.
```

Index

- *Topic **\textasciitildekwd1**
 - plot.bammdata, [37](#)
- *Topic **\textasciitildekwd2**
 - plot.bammdata, [37](#)
- *Topic **datasets**
 - BAMMtools-data, [6](#)
- *Topic **graphics**
 - addBAMMshifts, [3](#)
 - assignColorBreaks, [5](#)
 - dtRates, [17](#)
 - getEventData, [25](#)
 - maximumShiftCredibility, [36](#)
 - ShiftProbsTree, [50](#)
 - subtreeBAMM, [54](#)
- *Topic **manip**
 - getmrca, [30](#)
 - maximumShiftCredibility, [36](#)
 - subsetEventData, [53](#)
- *Topic **models**
 - computeBayesFactors, [10](#)
 - credibleShiftSet, [12](#)
 - getBestShiftConfiguration, [19](#)
 - getBranchShiftPriors, [20](#)
 - getCladeRates, [22](#)
 - getCohortMatrix, [24](#)
 - getEventData, [25](#)
 - getMarginalBranchRateMatrix, [28](#)
 - getMeanBranchLengthTree, [29](#)
 - getRateThroughTimeMatrix, [31](#)
 - getShiftNodesFromIndex, [33](#)
 - getTipRates, [34](#)
 - setBAMMpriors, [48](#)
 - speciesByRatesMatrix, [51](#)
 - summary.bammdata, [55](#)
 - summary.branchprior, [56](#)
 - summary.credibleShiftSet, [56](#)
- *Topic **package**
 - BAMMtools-package, [2](#)
- addBAMMshifts, [3](#), [19](#), [34](#), [36](#), [37](#), [40](#)
- arc (BAMMtools-internal), [7](#)
- areEventConfigurationsIdentical (BAMMtools-internal), [7](#)
- areShiftSetsEqual (BAMMtools-internal), [7](#)
- as.bammdata (BAMMtools-internal), [7](#)
- as.phylo.bammdata (BAMMtools-internal), [7](#)
- as.phylo.branchprior (BAMMtools-internal), [7](#)
- assignColorBreaks, [5](#), [40](#)
- BAMMtools (BAMMtools-package), [2](#)
- BAMMtools-data, [6](#)
- BAMMtools-internal, [7](#)
- BAMMtools-package, [2](#)
- barLegend (BAMMtools-internal), [7](#)
- bayesFactorBranches, [8](#)
- branchMeanRateExponential (BAMMtools-internal), [7](#)
- cohorts, [9](#)
- colorMap (BAMMtools-internal), [7](#)
- colorRampPalette, [40](#)
- computeBayesFactors, [7](#), [10](#)
- credibleShiftSet, [9](#), [12](#), [17](#), [19](#), [21](#), [36](#), [37](#), [44](#), [56](#), [57](#)
- cumulativeShiftProbsTree, [37](#)
- cumulativeShiftProbsTree (ShiftProbsTree), [50](#)
- distinctShiftConfigurations, [9](#), [15](#), [16](#), [21](#), [42](#), [44](#), [56](#), [57](#)
- dtRates, [17](#), [27](#), [40](#)
- events.primates (BAMMtools-data), [6](#)
- events.whales (BAMMtools-data), [6](#)
- exponentialRate (BAMMtools-internal), [7](#)
- getBAMMCorrelationMatrix (getCohortMatrix), [24](#)

- getBestShiftConfiguration, [19](#), [36](#), [37](#)
- getBranchShiftPriors, [8](#), [9](#), [12](#), [15](#), [19](#), [20](#), [56](#)
- getCladeRates, [22](#)
- getCohortMatrix, [10](#), [24](#)
- getDesc (BAMMtools-internal), [7](#)
- getEventData, [19](#), [25](#), [30](#), [35](#), [45](#), [46](#)
- getMacroevolutionaryCohortMatrix (getCohortMatrix), [24](#)
- getMarginalBranchRateMatrix, [28](#)
- getMeanBranchLengthTree, [29](#)
- getmrca, [30](#), [54](#)
- getPathToRoot (BAMMtools-internal), [7](#)
- getRateThroughTimeMatrix, [31](#), [45](#), [46](#), [52](#)
- getRecursiveSequence (BAMMtools-internal), [7](#)
- getSampleCoMat (BAMMtools-internal), [7](#)
- getSequenceForwardTraversal (BAMMtools-internal), [7](#)
- getShiftNodesFromIndex, [4](#), [33](#)
- getSpanningTaxonPair (BAMMtools-internal), [7](#)
- getStartStopTimes (BAMMtools-internal), [7](#)
- getTipRates, [34](#)
- image, [10](#)
- inv.logit (BAMMtools-internal), [7](#)
- logit (BAMMtools-internal), [7](#)
- marginalShiftProbsTree, [8](#), [36](#), [37](#)
- marginalShiftProbsTree (ShiftProbsTree), [50](#)
- mass.primates (BAMMtools-data), [6](#)
- maximumShiftCredibility, [34](#), [36](#), [51](#)
- mcmc.primates (BAMMtools-data), [6](#)
- mcmc.whales (BAMMtools-data), [6](#)
- mkdtsegsPhylo (BAMMtools-internal), [7](#)
- mkdtsegsPolar (BAMMtools-internal), [7](#)
- palettes (BAMMtools-internal), [7](#)
- phylogeneticMean (BAMMtools-internal), [7](#)
- plot.bammdata, [4](#), [5](#), [10](#), [18](#), [19](#), [27](#), [30](#), [34](#), [37](#), [42](#), [44](#), [54](#)
- plot.bammshifts, [15](#), [17](#), [21](#), [41](#), [44](#), [56](#), [57](#)
- plot.credibleshiftset, [15](#), [19](#), [21](#), [43](#)
- plotRateThroughTime, [31](#), [32](#), [44](#)
- primates (BAMMtools-data), [6](#)
- print.bammdata (BAMMtools-internal), [7](#)
- print.branchprior (summary.branchprior), [56](#)
- print.credibleshiftset (summary.credibleshiftset), [56](#)
- prior.primates (BAMMtools-data), [6](#)
- prior.whales (BAMMtools-data), [6](#)
- rateLegend (BAMMtools-internal), [7](#)
- recursivelySetNodeStates (BAMMtools-internal), [7](#)
- redirect (BAMMtools-internal), [7](#)
- richColors (BAMMtools-internal), [7](#)
- samplingProbs, [47](#)
- segMap (BAMMtools-internal), [7](#)
- setBAMMpriors, [48](#)
- setPhyloTreeCoords (BAMMtools-internal), [7](#)
- setPolarTreeCoords (BAMMtools-internal), [7](#)
- ShiftProbsTree, [50](#)
- speciesByRatesMatrix, [51](#)
- subsetEventData, [53](#)
- subtreeBAMM, [40](#), [54](#)
- summary.bammdata, [27](#), [55](#)
- summary.branchprior, [56](#)
- summary.credibleshiftset, [15](#), [21](#), [56](#)
- testTimeVariableBranches, [58](#)
- timeIntegratedBranchRate (BAMMtools-internal), [7](#)
- transparentColor (BAMMtools-internal), [7](#)
- whales (BAMMtools-data), [6](#)