Package ‘BAMMtools’

September 13, 2024

Type Package

Title Analysis and Visualization of Macroevolutionary Dynamics on Phylogenetic Trees

Version 2.1.12

Imports Rcpp (>= 0.11.0), gplots, methods

Depends ape, R (>= 2.10)

Suggests parallel

Description 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://bamm-project.org/

NeedsCompilation yes

Encoding UTF-8

RoxygenNote 7.3.2

Author Dan Rabosky [aut],
Michael Grundler [aut],
Pascal Title [aut, cre] (<https://orcid.org/0000-0002-6316-0736>),
Carlos Anderson [aut],
Jeff Shi [aut],
Joseph Brown [aut],
Huateng Huang [aut],
Jon Mitchell [aut]

Maintainer Pascal Title <ptitle@umich.edu>

Repository CRAN

Date/Publication 2024-09-12 22:10:47 UTC

Contents

addBAMMlegend . . . 3
addBAMMshifts . . . 5
assignColorBreaks . . . 6
BAMMlikelihood . . . 8
BAMMtools . . . 10
BAMMtools-data . . . 11
cohorts . . . 12
computeBayesFactors . . . 14
credibleShiftSet . . . 15
cumulativeShiftProbsTree . . . 19
distinctShiftConfigurations . . . 20
dtRates . . . 22
generateControlFile . . . 23
getBestShiftConfiguration . . . 25
getBranchShiftPriors . . . 26
getCladeRates . . . 27
getCohortMatrix . . . 29
getEventData . . . 30
getJenksBreaks . . . 33
getMarginalBranchRateMatrix . . . 34
getMeanBranchLengthTree . . . 35
getmrca . . . 37
getRateThroughTimeMatrix . . . 37
getShiftNodesFromIndex . . . 40
getTipRates . . . 41
marginalOddsRatioBranches . . . 42
maximumShiftCredibility . . . 43
plot.bammdata . . . 45
plot.bammshifts . . . 50
plot.credibleshiftset . . . 52
plotPrior . . . 54
plotRateThroughTime . . . 55
ratesHistogram . . . 58
richColors . . . 60
samplingProbs . . . 60
setBAMMpriors . . . 62
speciesByRatesMatrix . . . 63
stepBF . . . 65
subsetEventData . . . 66
subtreeBAMM . . . 67
summary.bammdata . . . 68
summary.credibleshiftset . . . 69
testTimeVariableBranches . . . 70
traitDependentBAMM . . . 72
transparentColor . . . 75
writeEventData . . . 76

Index 77

`addBAMMlegend` *Add a color legend to a phylo-rate plot*

Description

Add a legend to a phylorate plot, with greater manual control.

Usage

```
addBAMMlegend(
  x,
  direction,
  side,
  location = "topleft",
  nTicks = 2,
  stretchInterval = FALSE,
  shortFrac = 0.02,
  longFrac = 0.3,
  axisOffset = 0.002,
  cex.axis = 0.8,
  labelDist = 0.7,
  ...
)
```

Arguments

`x`	A `plot.bammdata` object.
`direction`	Direction of color ramp. If omitted, then direction is automatically inferred, otherwise can be specified as horizontal or vertical.
`side`	Side for tick marks, see `axis` documentation. Automatically inferred if omitted.
`location`	Either a location name (see Details), or coordinates for the corners of the bar legend c(xmin, xmax, ymin, ymax).
`nTicks`	Number of tick marks, besides min and max.
`stretchInterval`	If color.interval was defined, should the legend be stretched to the color.interval, or should the full range of rates be presented.
`shortFrac`	Percent of the plot width range that will be used as the short dimention of the legend. Only applies to preset location options.
`longFrac`	Percent of the plot width range that will be used as the long dimension of the legend. Only applies to preset location options.
`axisOffset`	Distance from color bar for labels, as a percent of the plot width.
`cex.axis`	Size of axis labels.
`labelDist`	Distance from axis to axis labels (passed to mgp).
`...`	Additional parameters to be passed to axis.

Details

A number of predefined locations exist in this function to make it easy to add a legend to a phylorate plot. Preset `locations` are: `topleft`, `topright`, `bottomleft`, `bottomright`, `left`, `right`, `top` and `bottom`. If more fine-tuned control is desired, then a numeric vector of length 4 can be supplied to `location`, specifying the min x, max x, min y and max y values for the legend. See `Examples`.

Value

Invisibly returns a list with the following components:

- coords: A 2-column matrix of xy coordinates for each color bin in the legend.
- width: Coordinates for the short dimension of the legend.
- pal: The color ramp.
- tickLocs: The tick mark locations in plotting units.
- labels: The rate values associated with those tick locations.

Author(s)

Pascal Title

See Also

Requires an object created with `plot.bammdata`.

Examples

```
data(whales, events.whales)
ephy <- getEventData(whales, events.whales, burnin = 0.25, nsamples = 300)

# plot phylorate with extra margin space
x <- plot(ephy, lwd = 2, mar = c(5,4,4,4))
# presets
addBAMMlegend(x, location = 'topleft')
addBAMMlegend(x, location = 'bottom')
addBAMMlegend(x, location = 'right')

# fine-tune placement
x <- plot(ephy, lwd = 2, mar = c(5,4,4,4))
axis(1); axis(2)
addBAMMlegend(x, location = c(-1, -0.5, 40, 80), nTicks = 4)
addBAMMlegend(x, location = c(5, 20, 60, 61), nTicks = 4, side = 3,
              cex.axis = 0.7)

# addBAMMlegend also automatically detects the use of color.interval
data(primates, events.primates)
ephy <- getEventData(primates, events.primates, burnin=0.25,
                     nsamples = 300, type = 'trait')

x <- plot(ephy, breaksmethod = 'linear',
          color.interval = c(NA, 0.12), lwd = 2)
addBAMMlegend(x, location = c(0, 30, 200, 205), nTicks = 1, side = 3)
```

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.
index	An integer indicating which posterior sample to use for adding shifts to the plotted tree.
method	A character string indicating the method used in plotting. Must be "polar" or "phylogram".
cex	A numeric indicating the character expansion ("size") of the plotted points.
pch	An integer indicating the choice of plotting symbol.
col	An integer or character string indicating the border color of the plotting symbol.
bg	An integer or character string indicating the background color of the plotting symbol.
msp	If not NULL, an object of class phylo 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.
shiftnodes	An optional vector of node numbers indicating the locations of shifts to plot.
par.reset	A logical indicating whether to reset the graphical parameters before exiting.
...	additional arguments to be passed to points.

Details

Any given sample from the posterior distribution sampled using BAMM 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 BAMM for a given sample from the posterior.

If the bammdata 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.bammdata

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25, nsamples = 500)

# 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 = 64,
  spex = "s",
  logcolor = FALSE,
  method = c("linear", "quantile", "jenks"),
  JenksSubset = NULL
)
```

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. "netdiv" means that diversification rates are used. Ignored for `BAMM` trait data.
`logcolor`	Logical. Should the natural logarithm of rates be used for the color map.
`method`	Determines how the color breaks are created. See Details.
`JenksSubset`	Number of regularly spaced samples to subset from `rates`. Only relevant when `method = "jenks"`. See Details.

Details

If `method = "quantile"` 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.bammdata`. Alternatively, if `method = "linear"` macroevolutionary rates are binned into `NCOLORS+1` equal length intervals between the minimum and maximum.

If `method = "jenks"`, macroevolutionary rates are binned into `NCOLORS+1` categories, according to the Jenks natural breaks classification method. This method is borrowed from the field of cartography, and seeks to minimize the variance within categories, while maximizing the variance between categories.

The Jenks natural breaks method was ported to C from code found in the classInt R package.

Value

A numeric vector of rate percentiles/intervals.

Author(s)

Mike Grundler, Pascal Title

See Also

`plot.bammdata`

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin = 0.2, nsamples = 500)

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="netdiv")
#diversification rates

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

plot(ed, colorbreaks = colors, spex="netdiv")
```

BAMMlikelihood *Calculate* BAMM *likelihood*

Description

Calculates the likelihood of a phylogeny exactly as is done by BAMM, given a set of events.

Usage

```
BAMMlikelihood(
  phy,
  eventdata,
  gen = "last",
  segLength = 0.02,
  sf = 1,
  return.intermediates = FALSE,
  e_prob_condition = "if_different",
  ...
)
```

Arguments

phy	Either an object of class phylo or the path to a tree file in newick format.
eventdata	A table of event data, as returned by BAMM, either as an object of class dataframe or as the path to an event_data file. Alternatively, a named numeric vector of length two holding speciation ("lambda") and extinction ("mu") rates for the constant-rate birth-death process.
gen	The BAMM generation for which the likelihood should be calculated. Can be an integer specifying a specific generation, or last, specifying the last generation, or all, in which case the likelihood will be calculated for all generations.

segLength — The relative segment length, exactly as defined for BAMM.

sf — The sampling fraction.

return.intermediates — Debugging option, returns augmented phylo objects for each generation, see Details.

e_prob_condition — Approach for how extinction probabilities are handled at nodes.

... — Additional arguments that will be passed to an internal function computeBAMMlikelihood.

Details

This function allows the user to check the likelihoods computed by BAMM using an independent R-based implementation. This is designed to provide a check on potential software bugs that might be introduced during future BAMM development and which might compromise the likelihood calculation. If you observe measurable discrepancies between the likelihood computed by this function and the corresponding likelihood returned by BAMM, please inform the BAMM development team.

Value

If return.intermediates == TRUE, then phylo objects are returned with the following components:

event_times — A list of length (number of nodes), where event_times[[k]] is the vector of absolute times, in order, of events that happened on a focal branch. If no event, it is NULL.

event_id — A list of length equal to number of nodes, as event_times, but holding the corresponding event id.

events — A dataframe giving parameters and associated nodes (and unique index values) of the event data.

node_event — The event governing the process realized at the node. This will be the first event encountered as one moves rootwards towards the tips from the focal node.

Author(s)

Dan Rabosky, Pascal Title

Examples

```
# a global sampling fraction of 0.98 was used in generating the whales
# dataset.
data(whales, events.whales, mcmc.whales)

x <- BAMMlikelihood(whales, events.whales, gen = 'last', sf = 0.98)

# Does the likelihood generated by BAMM match the R implementation?
identical(round(x, 3), mcmc.whales[nrow(mcmc.whales), 'logLik'])

# an example with a constant-rate birth-death process:
pars <- c(0.5, 0.45)
```

```
names(pars) <- c("lambda", "mu")
BAMMlikelihood(whales, pars, sf = 0.98)
```

BAMMtools *BAMMtools*

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 (http://bamm-project.org/).

Author(s)

Dan Rabosky, Mike Grundler, Pascal Title, Jonathan Mitchell, Carlos Anderson, Jeff Shi, Joseph Brown, Huateng Huang

References

http://bamm-project.org/

Rabosky, D., M. Grundler, C. Anderson, P. Title, J. Shi, J. Brown, H. Huang and J. Larson. 2014. BAMMtools: an R package for the analysis of evolutionary dynamics on phylogenetic trees. Methods in Ecology and Evolution 5: 701-707.

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

Shi, J. J., and D. L. Rabosky. 2015. Speciation dynamics during the global radiation of extant bats. Evolution 69: 1528-1545.

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

Useful links:

- http://bamm-project.org/

BAMMtools-data *BAMMtools datasets*

Description

Example datasets and sample BAMM output for the package BAMMtools.

Details

This includes both the raw data and the BAMM output for three 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).The third is a BAMM analysis of speciation and extinction rates for a 300-species subset of ray-finned fishes, along with body size data for these species from Rabosky et al. (2013).

Dataset whales is the raw time-calibrated tree that was analyzed with BAMM, primates is the corresponding time-calibrated phylogeny of 233 primate species, and fishes is the time-calibrated phylogeny of 300 fish species. Log-transformed body masses for primates are in dataset mass.primates, and fish body sizes are in dataset traits.fishes.

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 "core" BAMM output is included in the *event data* files (events.whales, events.primates and events.fishes). 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).

Source

Vos R.A., A.O. Mooers. 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., C. DeWolff, A.O. Mooers. 2010. Evolutionary distinctiveness, threat status and ecological oddity in primates. Conservation Biology 24: 1052-1058. DOI: 10.1111/j.1523-1739.2010.01532.x

Steeman, 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. Radiation of Extant Cetaceans Driven by Restructuring of the Oceans. Systematic Biology. 58: 573-585. DOI: 10.1093/sysbio/syp060

Rabosky, D. L., F. Santini, J.T. Eastman, S. . 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.

`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.bammdata = FALSE,
  useraster = FALSE,
  LARGE = 500,
  ...
)
```

Arguments

`x`	A matrix of pairwise correlations generated by `getCohortMatrix`.
`ephy`	An object of class `bammdata`.
`col`	A vector of colors passed to the function `image`. These will be used to color the values in `x`. See documentation for `image`. If `col = 'temperature'`, the color palette from `rich.colors` from the gplots package will be used.
`pal`	The palette to use if `use.plot.bammdata=TRUE`. See options documented in the help file for `plot.bammdata`.
`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`.

`...` Further arguments passed to `plot.bammdata` if `use.plot.bammdata=TRUE` or `plot.phylo` if `use.plot.bammdata=FALSE`.

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.

IMPORTANT: the legend DOES NOT apply to the phylorate plots shown in the margin if `use.plot.bammdata=TRUE`.

Author(s)

Mike Grundler

See Also

`plot.bammdata`, `getCohortMatrix`, `image`

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.1, nsamples=500)
x <- getCohortMatrix(ed)
cohorts(x, ed)
cohorts(x, ed, col='temperature')
cohorts(x, ed, ofs=0.05, col='temperature')
cohorts(x, ed, pal="temperature", col='temperature', use.plot.bammdata=TRUE)
# gray scale
cohorts(x, ed, col=gray(seq(0.2,0.9,length.out=128)),
        use.plot.bammdata=FALSE)
```

computeBayesFactors *Compute Bayes Factors*

Description

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

Usage

```
computeBayesFactors(postdata, expectedNumberOfShifts, burnin = 0.1, ...)
```

Arguments

postdata	Filename for the MCMC output file from a BAMM run. Alternatively, a dataframe containing this information.
expectedNumberOfShifts	Expected number of shifts under the prior.
burnin	What fraction of samples to discard from postdata as burnin?
...	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).

This function will only compute Bayes factors between models which were actually sampled during simulation of the posterior. Hence, if a model has such low probability that it is never visited by BAMM during the simulation of the posterior, it will be impossible to estimate its posterior probability (and thus, you will get no Bayes factors involving this particular model). This is likely to change in the future with more robust methods for estimating posterior probabilities in the tails of the distribution.

Value

A matrix of pairwise Bayes factors between models.

Author(s)

Dan Rabosky

Examples

```
data(mcmc.whales)
computeBayesFactors(mcmc.whales, expectedNumberOfShifts = 1, burnin = 0.1)
```

`credibleShiftSet` *Credible set of macroevolutionary rate shift configurations from* `BAMM` *results*

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,
  expectedNumberOfShifts,
  threshold = 5,
  set.limit = 0.95,
  ...
)
```

Arguments

`ephy`	An object of class `bammdata`.
`expectedNumberOfShifts`	The expected number of rate shifts under the prior.
`threshold`	The marginal posterior-to-prior odds ratio for a rate shift on a specific branch, used to distinguish core and non-core shifts.
`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 "poissonRatePrior" 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 marginal odds ratio: it is the marginal posterior frequency of one or more rate shifts normalized by the corresponding prior probability. Hence, any branch with a marginal odds ratio 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 `threshold` argument. This is simply a cutoff value for identifying "important" shifts for the purposes of identifying distinct shift configurations. This does not imply that it is identifying "significant" shifts. See the online documentation on this topic available at http://bamm-project.org for more information. If you specify `threshold = 5` as an argument to `credibleShiftSet`, the function will ignore all branches with marginal odds ratios less than 5 during the enumeration of topologically distinct shift configurations. Only shifts with marginal odds ratios greater than or equal to `threshold` 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.

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.

`frequency` A vector of frequencies of shift configurations, including those that account for `set.limit` (typically, 0.95 or 0.99) of the probability of the data. The index of the i'th element of this vector is the i'th most probable shift configuration (excepting ties).

`shiftnodes` A list of the "core" rate shifts (marginal probability > threshold) that occurred in each distinct shift configuration in the credible set. The i'th vector from this list gives the core shift nodes for the i'th shift configuration. They are sorted by frequency, so `x$shiftnodes[[1]]` gives the shift nodes that occurred together in the shift configuration with the highest posterior probability.

`indices` A list of vectors containing the indices of samples in the `bammdata` object that are assigned to a given shift configuration. All are sorted by frequency.

`cumulative` Like `frequency`, but contains the cumulative frequencies.

threshold
: The marginal posterior-to-prior odds for rate shifts on branches used during enumeration of distinct shift configurations.

number.distinct
: Number of distinct shift configurations in the credible set.

set.limit
: which credible set is this (0.9, 0.95, etc)?

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.

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

edge
: See documentation for class phylo in package ape.

Nnode
: See documentation for class phylo in package ape.

tip.label
: See documentation for class phylo in package ape.

edge.length
: See documentation for class phylo in package ape.

begin
: The beginning time of each branch in absolute time (the root is set to time zero)

end
: The ending time of each branch in absolute time.

numberEvents
: An integer vector with the number of core events contained in the bammdata 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.

eventData
: 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: node - a node number. This identifies the branch where the event originates. time - this is the absolute time on that branch where the event originates (with the root at time 0). lam1 - an initial rate of speciation or trait evolution.lam2 - a decay/growth parameter. mu1 - an initial rate of extinction. mu2 - a decay/growth parameter. index - a unique integer associated with the event. See 'Details' in the documentation for getEventData for more information.

eventVectors
: A list of integer vectors. Each element is for a single shift configuration in the posterior. For each branch in the bammdata object, gives the index of the event governing the (tipwards) end of the branch. Branches are ordered increasing here and elsewhere.

eventBranchSegs
: 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 phy. 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 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.

tipLambda	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 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 downseq[which(downseq==89):which(downseq==lastvisit[89]).

Author(s)

Dan Rabosky

References

http://bamm-project.org/

See Also

distinctShiftConfigurations, plot.bammshifts, summary.credibleshiftset, plot.credibleshiftset, getBranchShiftPriors

Examples

```
data(events.whales, whales)
ed <- getEventData(whales, events.whales, burnin=0.1, nsamples=500)

cset <- credibleShiftSet(ed, expectedNumberOfShifts = 1, threshold = 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(cset, plotmax=4)
```

---

cumulativeShiftProbsTree

Branch-specific rate shift probabilities

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

http://bamm-project.org/

See Also

maximumShiftCredibility

Examples

```
data(whales)
data(events.whales)
ed <- getEventData(whales, events.whales, nsamples = 500)

# 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)
```

---

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, expectedNumberOfShifts, threshold, ...)
```

Arguments

`ephy`	An object of class `bammdata`.
`expectedNumberOfShifts`	The expected number of rate shifts under the prior.
`threshold`	Threshold value for marginal posterior-to-prior odds ratios, used to identify branches with elevated shift probabilities relative to prior (core vs non-core shifts).
`...`	Other arguments to distinctShiftConfigurations (possibly deprecated args).

Details

See Rabosky et al (2014) and the `BAMM` project website for details on the nature of distinct shift configurations and especially the distinction between "core" and "non-core" rate shifts. Note that branches with elevated marginal posterior probabilities relative to the prior (marginal odds ratios) cannot be claimed to have "significant" evidence for a rate shift on the basis of this evidence alone.

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 shift configuration.
- marginal_odd_ratio: Marginal posterior-to-prior odds ratios for one or more rate shifts an a given 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 marginal odds ratio of at least `threshold` supporting a rate shift.
- threshold: A single numeric value giving the marginal posterior:prior odds ratio 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)
ed <- getEventData(whales, events.whales, burnin=0.25, nsamples=500)

sc <- distinctShiftConfigurations(ed, expectedNumberOfShifts = 1,
                                  threshold = 5)

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

---

dtRates — *Calculate macroevolutionary rate changes on a phylogeny from* BAMM *output*

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 bammdata.
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 Grundler

References

http://bamm-project.org/

See Also

plot.bammdata

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25, nsamples=500)

# 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)
```

---

generateControlFile *Generate control file for* BAMM

Description

Generates a template diversification or trait control file for BAMM, while allowing the user to specify parameter values.

Usage

```
generateControlFile(
  file = "controlfile.txt",
  type = "diversification",
  params = NULL
)
```

Arguments

file	Destination file name with or without path.
type	Character, either "diversification" or "trait", depending on the desired BAMM analysis.
params	List of parameters, see Details.

Details

The user can supply parameters as a list, where the name of the list item is the name of the parameter as it appears in the control file, and the value of the list item is what will be placed in the contol file.

If a parameter is specified by the user, it will automatically be uncommented if it was commented in the template.

Author(s)

Pascal Title

References

http://bamm-project.org/

Examples

```
## Not run:
#Produce a blank template control file
generateControlFile(file = 'traitcontrol.txt', type='trait')

#Produce a customized control file
data(whales)

#get bamm priors to supply to control file
priors <- setBAMMpriors(whales, outfile = NULL)

generateControlFile(file = 'divcontrol.txt', params = list(
    treefile = 'whales.tre',
    globalSamplingFraction = '1',
    numberOfGenerations = '100000',
    overwrite = '1',
    lambdaInitPrior = as.numeric(priors['lambdaInitPrior']),
    lambdaShiftPrior = as.numeric(priors['lambdaShiftPrior']),
    muInitPrior = as.numeric(priors['muInitPrior']),
    expectedNumberOfShifts = '1'))
```

```
## End(Not run)
```

`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, expectedNumberOfShifts, threshold = 5)
```

Arguments

`x`	Either a `bammdata` object or a `credibleshiftset` object.
`expectedNumberOfShifts`	The expected number of shifts under the prior.
`threshold`	The marginal posterior-to-prior odds ratio used as a cutoff for distinguishing between "core" and "non-core" rate shifts.

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 marginal posterior-to-prior odds ratio `threshold`.

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, nsamples=500)

# Get prior distribution on shifts-per-branch:
bp <- getBranchShiftPriors(whales, expectedNumberOfShifts = 1)

# Pass the event data object in to the function:
best <- getBestShiftConfiguration(ed, expectedNumberOfShifts = 1,
                                  threshold = 5)
plot(best, lwd=2)
addBAMMshifts(best, cex=2)

# Now we can also work with the credible shift set:
css <- credibleShiftSet(ed, expectedNumberOfShifts = 1, threshold = 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, expectedNumberOfShifts = 1,
                                  threshold = 5)

plot(best)
addBAMMshifts(best)
```

`getBranchShiftPriors` *Compute prior odds of a rate shift on each branch of a phylogeny from BAMM 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, expectedNumberOfShifts)
```

Arguments

`phy`	An object of class `phylo`.
`expectedNumberOfShifts`	The expected number of shifts under the prior.

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 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.

Value

A class `phylo` 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

http://bamm-project.org/

See Also

`distinctShiftConfigurations`, `plot.bammshifts`, `summary.credibleshiftset`, `plot.credibleshiftset`, `credibleShiftSet`

Examples

```
data(whales)
prior_tree1 <- getBranchShiftPriors(whales, expectedNumberOfShifts = 1)
prior_tree10 <- getBranchShiftPriors(whales, expectedNumberOfShifts = 10)
# plot prior expectations for branches based on these two counts:
plot(prior_tree1$edge.length ~ prior_tree10$edge.length, xlim=c(0,0.05),
     ylim=c(0,0.05), asp=1)
lines(x=c(0,1), y=c(0,1))
```

---

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. If multiple nodes are supplied, then the equivalent number of nodetype must be supplied.
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. If multiple nodes are specified, there must be a nodetype for each node.
verbose	Logical. 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'th rate corresponding to the mean rate from the i'th sample in the posterior.
- mu: A vector of extinction rates (if applicable), with the i'th rate corresponding to the mean rate from the i'th sample in the posterior.
- beta: A vector of phenotypic rates (if applicable), with the i'th rate corresponding to the mean rate from the i'th sample in the posterior.

Author(s)

Dan Rabosky

References

http://bamm-project.org/

Examples

```
data(events.whales, whales)
ed <- getEventData(whales, events.whales, nsamples=500)
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)

# defining multiple nodes
mean(getCladeRates(ed, node=c(132, 140),
     nodetype=c('include','exclude'))$lambda)
```

---

getCohortMatrix *Compute the pairwise correlation in rate regimes between all tips in a* bammdata *object*

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 x 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

http://bamm-project.org/

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, nsamples=500)

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 BAMM 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 BAMM. Alternatively, a character string specifying the path to a newick-formatted tree.
eventdata	A character string specifying the path to a BAMM event-data file. Alternatively, an object of class data.frame that includes the event data from a BAMM 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 progess is outputted to the console. Defaults to FALSE.
type	A character string. Either "diversification" or "trait" depending on your BAMM analysis.

Details

In the BAMM 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, BAMM 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 BAMM 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 BAMMtools functions.

The parameter nsamples can be used to reduce the total amount of data included in the raw eventdata output from a BAMM 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:

- edge: See documentation for class phylo in package ape.
- Nnode: See documentation for class phylo in package ape.

- tip.label: See documentation for class `phylo` in package ape.
- edge.length: See documentation for class `phylo` in package ape.
- begin: The beginning time of each branch in absolute time (the root is set to time zero)
- end: The ending time of each branch in absolute time.
- numberEvents: An integer vector with the number of events contained in `phy` for each posterior sample. The length of this vector is equal to the number of posterior samples in the `bammdata` object.
- eventData: 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: `node` - a node number. This identifies the branch where the event originates. `time` - this is the absolute time on that branch where the event originates (with the root at time 0). `lam1` - an initial rate of speciation or trait evolution. `lam2` - a decay/growth parameter. `mu1` - an initial rate of extinction. `mu2` - a decay/growth parameter. `index` - a unique integer associated with the event. See 'Details'.
- eventVectors: A list of integer vectors. Each element is a single posterior sample. For each branch in `phy` the index of the event that occurs along that branch. Branches are ordered increasing here and elsewhere.
- eventBranchSegs: A list of matrices. Each element is a single posterior sample. Each matrix has four columns: `Column 1` identifies a node in `phy`. `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 `downseq[which(downseq==89):which(downse`

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 Grundler

References

http://bamm-project.org/

See Also

summary.bammdata, plot.bammdata, dtRates.

Examples

```
data(primates, events.primates)
xx <- getEventData(primates, events.primates, burnin=0.25, nsamples=500,
                   type = 'trait')

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

# Plot rates:
plot(xx)
```

---

getJenksBreaks *Jenks natural breaks classification*

Description

Given a vector of numeric values and the number of desired breaks, calculate the optimum breakpoints using Jenks natural breaks optimization.

Usage

```
getJenksBreaks(var, k, subset = NULL)
```

Arguments

var	Numeric vector.
k	Number of breaks.
subset	Number of regularly spaced samples to subset from var. Intended to improve runtime for large datasets. If NULL, all values are used.

Details

getJenksBreaks is called by assignColorBreaks.

The values in var are binned into k+1 categories, according to the Jenks natural breaks classification method. This method is borrowed from the field of cartography, and seeks to minimize the variance within categories, while maximizing the variance between categories. If subset = NULL, all values of var are used for the optimization, however this can be a slow process with very large datasets. If subset is set to some number, then subset regularly spaced values of var will be sampled. This is slightly less accurate than when using the entirety of var but is unlikely to make much of a difference. If subset is defined but length(var) < subset, then subset has no effect.

The Jenks natural breaks method was ported to C from code found in the classInt R package.

Value

A numeric vector of intervals.

Author(s)

Pascal Title

See Also

See assignColorBreaks and plot.bammdata.

Examples

```
# load whales dataset
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25, nsamples=500)

# for demonstration purposes, extract the vector of speciation rates
ed <- dtRates(ed, tau=0.01)
vec <- ed$dtrates$rates[[1]]

# Return breaks for the binning of speciation rates into 65 groups
# yielding 64 breaks
getJenksBreaks(vec, 64)
```

---

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

http://bamm-project.org/

See Also

plot.bammdata

Examples

```
data(whales)
data(events.whales)
ed <- getEventData(whales, events.whales, burnin=0.1, nsamples=500)
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 Grundler

See Also

`subtreeBAMM`

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 BAMM 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:

lambda	A nsamples x nslices matrix of speciation rates, where nsamples is the number of posterior samples in the bammdata object.
mu	A nsamples x nslices matrix of extinction rates.
beta	A nsamples x nslices matrix of phenotypic rates (if applicable).
times	A vector of timepoints where rates were computed.
times	A vector of timepoints where rates were computed (see Examples).
type	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(Not run)
```

---

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 *not* 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, burnin=0.1, nsamples=500)

# 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	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, nsamples = 500)

# 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)
```

`marginalOddsRatioBranches` *Ratio of (marginal) posterior-to-prior probabilities on individual branches*

Description

Compute marginal posterior-to-prior odds ratio associated with observing one or more rate shift on a given branch.

Usage

```
marginalOddsRatioBranches(ephy, expectedNumberOfShifts)
```

Arguments

ephy	An object of class bammdata.
expectedNumberOfShifts	Expected number of shifts under the prior alone.

Details

This function returns a copy of a phylogenetic tree where each branch length is equal to the marginal odds ratio in favor of a rate shift on a particular branch. These cannot be interpreted as evidence for a rate shift in an absolute sense. As explained on the website, they are a marginal odds ratio. This function is provided primarily for the purpose of distinguishing core and non-core shifts.

Value

A object of class phylo but where each branch length is equal to the marginal shift odds on each branch.

Author(s)

Dan Rabosky

See Also

getBranchShiftPriors, distinctShiftConfigurations, credibleShiftSet

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.1, nsamples=500)
marginalOddsRatioBranches(ed, expectedNumberOfShifts = 1)
```

---

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 marginalShiftProbsTree. 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, getBestShiftConfiguration

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25, nsamples=500)
best_config <- maximumShiftCredibility(ed)
plot(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",
  xlim = NULL,
  ylim = NULL,
  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,
  breaksmethod = "linear",
  color.interval = NULL,
  JenksSubset = 20000,
  par.reset = FALSE,
  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.
xlim	A numeric vector of coordinates for the x-axis endpoints. Defaults to NULL, in which case they are calculated automatically. The x-axis corresponds to time when the phylogeny is plotted facing to the left or to the right. The time at the root equals zero.
ylim	A numeric vector of coordinates for the y-axis endpoints. Defaults to NULL, in which case they are calculated automatically. Tips are plotted at integer values beginning from zero and stepping by one when the phylogeny is plotted facing to the left or to the right.
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 "netdiv" indicates net diversification rates. Ignored if ephy$type = "trait".
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 mask.color 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 bammdata object.
logcolor	Logical. Should colors be plotted on a log scale.
breaksmethod	Method used for determining color breaks. See help file for assignColorBreaks.
color.interval	Min and max value for the mapping of rates. If NULL, then min and max are inferred from the data. NA can also be supplied for one of the two values. See details.

JenksSubset	If breaksmethod = "jenks", the number of regularly spaced samples to subset from the full rates vector. Only relevant for large datasets. See help file for assignColorBreaks.
par.reset	A logical indicating whether or not to reset the graphical parameters when the function exits. Defaults 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 par.

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 = "netdiv" 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 that contains the mapping of colors to numerical values.

A number of color palettes come built in with BAMMtools. 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

Some grayscale options include:

- grayscale

- revgray

For more information about these color palettes visit https://colorbrewer2.org/ and https://pjbartlein.github.io/datagraphics/color_scales.html 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 color.interval is defined, then those min and max values override the automatic detection of min and max. This might be useful if some small number of lineages have very high or very low rates, such that the map of colors is being skewed towards these extremes, resulting in other rate variation being drowned out. If specified, the color ramp will be built between these two color.interval values, and the rates outside of the color interval range will be set to the highest and lowest color. The total number of colors will also be increased such that 64 color bins are found within the color.interval.

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.

- coords: 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 method = "polar" a fifth column gives the angle(in radians) of each branch.
- colorbreaks: 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.

Author(s)

Mike Grundler, Pascal Title

Source

https://colorbrewer2.org/, https://pjbartlein.github.io/datagraphics/color_scales.html

See Also

dtRates, addBAMMshifts, assignColorBreaks, subtreeBAMM, colorRampPalette

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25, nsamples=500)

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

# Compare the different color breaks methods
par(mfrow=c(1,3))
plot(ed, lwd = 3, spex = "s", breaksmethod = "linear")
title(main="linear")
plot(ed, lwd = 3, spex = "s", breaksmethod = "quantile")
title(main="quantile")
plot(ed, lwd = 3, spex = "s", breaksmethod = "jenks")
title(main="jenks")

## 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))
samples <- sample(1:length(ed$eventData), 6)
```

```
ed <- dtRates(ed, 0.005)
# individual plots will have a color map relative to the mean
xx <- plot(ed, show=FALSE)
for (i in 1:6) {
    ed <- dtRates(ed, 0.005, samples[i])
    plot(ed, 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(Not run)
```

`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,
  logcolor = FALSE,
  breaksmethod = "linear",
  color.interval = NULL,
  JenksSubset = 20000,
  ...
)
```

Arguments

x	An object of class bammshifts.
ephy	An object of class bammdata.
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 plot.bammdata.
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.
spex	A character string indicating what type of macroevolutionary rates should be plotted. "s" (default) indicates speciation rates, "e" indicates extinction rates, and 'netdiv' indicates net diversification rates. Ignored if ephy$type = "trait".
legend	Logical indicating whether to plot a legend.
add.freq.text	A logical indicating whether the frequency of each sampled shift configuration should be added to each plot.
logcolor	Logical. Should colors be plotted on a log scale.
breaksmethod	Method used for determining color breaks. See help file for assignColorBreaks.
color.interval	Min and max value for the mapping of rates. One of the two values can be NA. See details in plot.bammdata for further details.
JenksSubset	If breaksmethod = "jenks", the number of regularly-spaced samples to subset from the full rates vector. Only relevant for large datasets. See help file for assignColorBreaks.
...	Other arguments to plot.bammdata.

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

http://bamm-project.org/

See Also

distinctShiftConfigurations, plot.bammdata

Examples

```
data(whales, events.whales)

ed <- getEventData(whales, events.whales, burnin=0.25, nsamples=500)

sc <- distinctShiftConfigurations(ed, expectedNumberOfShifts = 1,
                                  threshold = 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",
  shiftColor = "black",
  spex = "s",
  add.freq.text = TRUE,
  use.plot.bammdata = TRUE,
  border = TRUE,
  legend = FALSE,
  send2pdf = FALSE,
  logcolor = FALSE,
  breaksmethod = "linear",
  color.interval = NULL,
  JenksSubset = 20000,
```

```
  ...
)
```

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.
shiftColor	Color to use for shift points.
spex	A character string indicating what type of macroevolutionary rates should be plotted. "s" (default) indicates speciation rates, "e" indicates extinction rates, and "netdiv" 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.
logcolor	A logical indicating whether the rates should be log-transformed.
breaksmethod	Method used for determining color breaks. See help file for assignColorBreaks.
color.interval	Min and max value for the mapping of rates. One of the two values can be NA. See details in plot.bammdata for further details.
JenksSubset	If breaksmethod = "jenks", the number of regularly spaced samples to subset from the full rates vector. Only relevant for large datasets. See help file for assignColorBreaks.
...	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.

Author(s)

Mike Grundler

References

http://bamm-project.org/

See Also

credibleShiftSet, distinctShiftConfigurations, plot.bammdata, plot.bammshifts

Examples

```
data(events.whales)
data(whales)
ed <- getEventData(whales, events.whales, nsamples=500)
cset <- credibleShiftSet(ed, expectedNumberOfShifts = 1, threshold = 5)
plot(cset)
```

---

plotPrior *Plot the prior and posterior distribution of shifts*

Description

Generates a barplot of the prior and posterior distributions of the number of shifts.

Usage

```
plotPrior(
  mcmc,
  expectedNumberOfShifts = 1,
  burnin = 0.15,
  priorCol = "light blue",
  postCol = "red",
  legendPos = "topright",
  ...
)
```

Arguments

mcmc	A dataframe of the mcmc_out file from a BAMM run, or the filename.
expectedNumberOfShifts	Expected number of shifts under the prior.
burnin	The fraction of samples to discard as burn-in.
priorCol	Color for the prior distribution.
postCol	Color for the posterior distribution.
legendPos	Placement of the legend, see legend.
...	Additional parameters that are passed to barplot.

Value

Invisibly returns a matrix with the probability of each shift number under the prior and the posterior.

Author(s)

Pascal Title

Examples

```
data(mcmc.whales)
plotPrior(mcmc.whales, expectedNumberOfShifts = 1, burnin = 0.15)
```

---

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 = 0.2,
  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:

- poly: A list of matrices, where each matrix contains the coordinates that define each overlapping confidence interval polygon.
- avg: A vector of y-coordinates for mean or median rates used to plot the average rates line.
- times: 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(Not run)
```

---

ratesHistogram *Histogram of* BAMM *rate frequencies*

Description

Plots a histogram of the frequency of rate values across the phylogeny.

Usage

```
ratesHistogram(
  phylorates,
  plotBrks = TRUE,
  xlab = "speciation rate",
  ylab = "density",
  lwd = 0.2,
  lty = 1,
  brksCol = "black",
  ...
)
```

Arguments

phylorates	A saved plot.bammdata object.
plotBrks	Boolean, should breaks be plotted over the histogram.
xlab	x-axis label.
ylab	y-axis label.
lwd	Line width for breaks.
lty	Line style for breaks.
brksCol	Color of breaks lines.
...	Additional arguments passed on to mtext for axis labels.

Details

With this function, a histogram is plotted that shows the frequency of rates present in the dataset. The color scheme plotted is taken from the saved plot.bammdata object that is the main input. Therefore, the mapping of colors to rates in the histogram corresponds exactly to what is plotted in the phylorate plot. If plotBrks = TRUE, then the color breaks used for the phylorates plot are shown.

This function can be a useful tool for exploring different plot.bammdata options. Please see http://bamm-project.org/colorbreaks.html on the bamm-project website for more information on the utility of this function.

Author(s)

Pascal Title

References

http://bamm-project.org/

See Also

plot.bammdata

Examples

```
data(primates, events.primates)
ed <- getEventData(primates, events.primates, burnin=0.25, nsamples=500,
                   type = 'trait')

# create phylorate plot with the jenks breaks method to generate output
phylorates <- plot(ed, breaksmethod='jenks', show = FALSE)

ratesHistogram(phylorates, plotBrks = TRUE, xlab = 'trait rates')
```

richColors — *Rich color palette*

Description

Deprecated function. Please use `rich.colors` instead.

Usage

```
richColors(n)
```

Arguments

`n`	The number of desired colors.

See Also

`rich.colors`

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 http://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 *Set BAMM Priors*

Description

Set priors for BAMM analysis.

Usage

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

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, or a numeric vector named according to the tips in phy. Leave NULL if doing a speciation-extinction analysis.
outfile	Filename for outputting the sample priors block. If NULL, then a vector is returned instead.
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.
suppressWarning	Logical. If TRUE, then the warning about setting the Poisson rate prior is suppressed. Only applies if outfile = NULL.

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. However, if `outfile = NULL`, then a named vector is returned.

Author(s)

Dan Rabosky

Examples

```
# for diversification analyses
data(whales)
setBAMMpriors(phy = whales, total.taxa = 89, outfile = NULL)

# for trait analyses
data("primates")
data("mass.primates")

## create a named vector of trait values
mass <- setNames(mass.primates[,2], mass.primates[,1])

setBAMMpriors(phy = primates, traits = mass, outfile = NULL)
```

`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 = NULL, spex = "s")
```

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; "netdiv" 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

http://bamm-project.org/

See Also

getRateThroughTimeMatrix

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, burnin=0.25, nsamples=500)
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)
```

---

stepBF *Identify the optimal number of shifts using Bayes factors*

Description

stepBF is a function to determine the overall best fitting number of shifts using Bayes factor evidence.

Usage

```
stepBF(BFmat, step.size = 20, expectedNumberOfShifts = 1, inputType = "matrix")
```

Arguments

`BFmat`	square Bayes factor matrix or a named vector of posterior probabilities
`step.size`	how much Bayes factor support is required to accept a more complex model, see Details
`expectedNumberOfShifts`	expected number of shifts under the prior (only needed for `inputType = 'postProb'`)
`inputType`	describes the input: `'matrix'` or `'postProb'`

Details

stepBF takes either a square Bayes factor matrix (such as output by `computeBayesFactors`) or a named vector of posterior probabilities. If posterior probabilities are supplied, the model prior (`expectedNumberOfShifts`) must also be provided. If the input is a Bayes factor matrix, specify `inputType = 'matrix'`, otherwise if the input is a named vector of posterior probabilities, specify `inputType = 'postProb'`.

The `step.size` argument is how much Bayes factor support is needed to accept a more complex model. By default, this value is 1, so any more complex model that has a better Bayes factor than the previous model will be accepted. Increasing the step size greatly reduces the Type I error at the cost of inflating Type II error. So, with increasing step.size, you will infer fewer shifts.

Value

a list of 3 items: the number of shifts for the best model, the number of shifts for the second best model, and the Bayes factor support for the best model over the second best.

Author(s)

Jonathan Mitchell

References

http://bamm-project.org/

See Also

computeBayesFactors

Examples

```
data(mcmc.whales)
# remove 10% burnin
mcmc.whales <- mcmc.whales[floor(0.1 * nrow(mcmc.whales)):nrow(mcmc.whales), ]
# from a square matrix of Bayes factor values (inputType = 'matrix')
bfmat <- computeBayesFactors(mcmc.whales, expectedNumberOfShifts = 1, burnin = 0)
stepBF(bfmat, step.size = 1, inputType = 'matrix')
# or from a vector of posterior probabilities (inputType = 'postProb')
postProb <- table(mcmc.whales$N_shifts) / nrow(mcmc.whales)
stepBF(postProb, step.size = 1, inputType = 'postProb')
```

---

subsetEventData *Subset a* bammdata *object*

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.

Author(s)

Dan Rabosky

See Also

plot.bammdata, getCohortMatrix, image

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, nsamples=500)
ed2 <- subsetEventData(ed, index=1)
plot(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

Value

An object of class bammdata.

Author(s)

Huateng Huang

See Also

getmrca, plot.bammdata

Examples

```
data(whales, events.whales)
ephy <- getEventData(whales, events.whales, burnin=0.25, nsamples=500)

# 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` *Summary of rate shift results from* `BAMM` *analysis*

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 (currently 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

http://bamm-project.org/

Examples

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

---

summary.credibleshiftset

Summary of credible set of shift configurations from a BAMM *analysis*

Description

Prints summary attributes of the BAMM 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 BAMM. 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

http://bamm-project.org/

See Also

distinctShiftConfigurations, plot.bammshifts, credibleShiftSet

Examples

```
data(whales, events.whales)
ed <- getEventData(whales, events.whales, nsamples = 500)
cset <- credibleShiftSet(ed, expectedNumberOfShifts = 1, threshold = 5)
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 BAMM 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 BAMM. 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

References

http://bamm-project.org/

See Also

getRateThroughTimeMatrix

Examples

```
# Load whale data:
data(whales, 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.
```

---

traitDependentBAMM *STRAPP: STructured Rate Permutations on Phylogenies*

Description

Given a bammdata object and a vector of (continuous) trait data, assess whether the correlation between the trait and bamm estimated speciation, extinction or net diversification rate is significant using permutation. A set of posterior samples is randomly drawn from the bammdata object. If the trait is continuous, this function calculates the correlation coefficients between the trait and tip rates (observed correlation), as well as that with permuted rates for each posterior sample. In a one-tailed test for positive correlations, the reported p-value is the proportion of the posterior samples in which the observed correlation is larger than the correlations calculated with permuted rates. In a two-tailed test, the reported p-value is the proportion of the posterior samples in which the null correlation is as extreme as the correlation observed. If the trait is binary, the U statistic of the Mann-Whitney test is calculated instead of correlation coefficients to assess whether there is a significant difference in rate between the two trait states. For categorical traits with more than two states, the Kruskal-Wallis rank sum statistic is used.

Usage

```
traitDependentBAMM(
  ephy,
  traits,
  reps,
  rate = "speciation",
  return.full = FALSE,
  method = "spearman",
  lograte s = TRUE,
  two.tailed = TRUE,
  traitorder = NA,
  nthreads = 1
)
```

Arguments

ephy	An object of class bammdata.
traits	A vector of trait data, with names corresponding to tips in the bammdata object. It can be numeric or categorical.
reps	An integer specifying the number of permutations (i.e., the number of posterior samples to randomly draw with replacement from the bammdata object).
rate	A character string specifying which estimated rate from the bammdata object to use for testing correlation, must be one of 'speciation', 'extinction', 'net diversification' or 'trait'. Defaults to 'speciation'. You can specify just the initial letter. Ignored for trait event data.
return.full	A logical. If TRUE, the list of posterior samples, the observed correlation for each posterior sample, and the null distribution will be included in the returned object. Defaults to FALSE.
method	A character string, must be one of 'spearman', 'pearson', 'mann-whitney', or 'kruskal'. Defaults to 'spearman'. You can specify just the initial letter.
lograte s	A logical. If TRUE log-transform the rates before analysis. Defaults to TRUE. This can only matter for the pearson correlation.

`two.tailed` A logical, used for continuous trait data. If `TRUE`, perform a two-tailed statistical test (i.e., if the null distribution is symmetric, it is equivalent to doubling the p-value). Defaults to `TRUE`.

`traitorder` A character string specifying the direction of correlation for the alternative hypothesis. For continuous traits, it must be either "positive" or "negative"; only the initial letter is needed. For binary traits, it must be a string indicating states with increasing rate under the alternative hypothesis, separated by comma (e.g., 'A, B'). One-tailed test for categorical data with more than two states is not supported.

`nthreads` Number of threads to use for parallelization of the function. The R package `parallel` must be loaded for `nthreads > 1`.

Details

Tip rates –trait, speciation, extinction, or net diversification rates– are permuted in a way such that pairwise covariances in rates between species are maintained. That is, tips with the same `tipStates` still have the same rate after permutation. Posterior samples are randomly selected with replacement from the `bammdata` object, so `reps` could be smaller or larger than the total number of samples in the object.

This function expects that the bamm-data object and the trait data have the same taxon set. It may be necessary to subset the trait data and/or run `subtreeBAMM` on the `bamm-data` object in order to meet this requirement.

Value

A list with the following components:

- estimate: A numeric value for continous trait data: the average observed correlation between tip rates and the trait across the posterior samples. For categorical traits, it is a list showing the median species-specific rates for each trait state.
- p.value: A numeric value. The probability that the observed correlation is less than or equal to a sample from the null distribution.
- method: A character string, as input.
- rate: A character string, as input.
- two.tailed: A logical, as input.
- gen: An integer vector, recording which posterior samples were selected. Only present when `return.full` is `TRUE`.
- obs.corr: A numeric vector, the observed correlation coefficents for each posterior sample. Only present when `return.full` is `TRUE`. For binary traits, centered U statistics (U - n1* n2/2; where n1 and n2 are the number of species in each state of the binary trait) is reported.
- null: A numeric vector. The null distribution of correlation coefficients (or centered U statistics for binary traits) from permutation. Only present when `return.full` is `TRUE`.

Author(s)

Dan Rabosky, Huateng Huang

References

http://bamm-project.org/

Rabosky, D. L. and Huang, H., 2015. A Robust Semi-Parametric Test for Detecting Trait-Dependent Diversification. Systematic Biology 65: 181-193.

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

subtreeBAMM

Examples

```
# using a small subset of the fish data set (300 taxa) in Rabosky et al.
# 2013. Nat. Com. paper
data(fishes, events.fishes)
xx <- getEventData(phy = fishes, eventdata = events.fishes,
      nsamples = 500, type = "diversification")
# traits.fishes is the trait -- body size
data(traits.fishes)
x <- traitDependentBAMM(ephy = xx, traits = traits.fishes, reps = 1000,
                        return.full = TRUE, method = 's', lograte s = TRUE,
                        two.tailed = TRUE)
```

---

transparentColor *Define colors with transparency*

Description

Converts a named color and opacity and returns the proper RGB code.

Usage

```
transparentColor(namedColor, alpha = 0.8)
```

Arguments

namedColor	A color name.
alpha	A transparency value between 0 and 1, where 0 is fully transparent.

Details

This function is used internally by plotRateThroughTime.

Value

Returns the transparent color in RGB format.

Author(s)

Pascal Title

---

writeEventData — *Write a* bammdata *object to disk*

Description

Takes a bammdata object and re-writes it back into a treefile and an event csv file.

Usage

```
writeEventData(ephy, outtreefile, outeventfile, ...)
```

Arguments

ephy	A bammdata object.
outtreefile	The file name for outputting the tree.
outeventfile	The file name for outputting the event csv file.
...	Additional arguments to pass to write.csv.

See Also

subtreeBAMM

Index

∗ **datasets**
BAMMtools-data, 11
∗ **graphics**
addBAMMshifts, 5
assignColorBreaks, 6
cumulativeShiftProbsTree, 19
dtRates, 22
getJenksBreaks, 33
maximumShiftCredibility, 43
plot.bammdata, 45
subtreeBAMM, 67
∗ **manip**
getmrca, 37
maximumShiftCredibility, 43
subsetEventData, 66
transparentColor, 75
∗ **models**
computeBayesFactors, 14
credibleShiftSet, 15
getBestShiftConfiguration, 25
getBranchShiftPriors, 26
getCladeRates, 27
getCohortMatrix, 29
getEventData, 30
getMarginalBranchRateMatrix, 34
getMeanBranchLengthTree, 35
getRateThroughTimeMatrix, 37
getShiftNodesFromIndex, 40
getTipRates, 41
plot.bammdata, 45
setBAMMpriors, 62
speciesByRatesMatrix, 63
summary.bammdata, 68
summary.credibleshiftset, 69
∗ **nonparametric**
traitDependentBAMM, 72
∗ **package**
BAMMtools, 10

addBAMMlegend, 3
addBAMMshifts, 5, *25*, *40*, *44*, *49*
assignColorBreaks, 6, *34*, *46, 47*, *49*, *51*, *53*
axis, *3*

BAMMlikelihood, 8
BAMMtools, 10
BAMMtools-data, 11
BAMMtools-package *(BAMMtools)*, 10
barplot, *54*

cohorts, 12
colorRampPalette, *49*
computeBayesFactors, 14, *65, 66*
credibleShiftSet, 15, *22*, *25*, *27*, *43, 44*, *54*, *70*
cumulativeShiftProbsTree, 19, *44*

distinctShiftConfigurations, *18*, 20, *27*, *43*, *51, 52*, *54*, *70*
dtRates, 22, *33*, *46*, *49*

events.fishes *(BAMMtools-data)*, 11
events.primates *(BAMMtools-data)*, 11
events.whales *(BAMMtools-data)*, 11

fishes *(BAMMtools-data)*, 11

generateControlFile, 23
getBestShiftConfiguration, 25, *44*
getBranchShiftPriors, *18*, 26, *43*
getCladeRates, 27
getCohortMatrix, *13*, 29, *67*
getEventData, *17*, *25*, 30, *37*, *42*, *57*
getJenksBreaks, 33
getMarginalBranchRateMatrix, 34
getMeanBranchLengthTree, 35
getmrca, 37, *67*
getRateThroughTimeMatrix, 37, *57*, *64*, *71*
getShiftNodesFromIndex, *6*, 40
getTipRates, 41

image, *13*, *67*

legend, *54*

marginalOddsRatioBranches, 42
marginalShiftProbsTree, *44*
marginalShiftProbsTree
(cumulativeShiftProbsTree), 19
mass.primates (BAMMtools-data), 11
maximumShiftCredibility, *20*, *40*, 43
mcmc.primates (BAMMtools-data), 11
mcmc.whales (BAMMtools-data), 11

plot.bammdata, *4*, *6*, *7*, *12*, *13*, *23*, *25*, *33*, *34*, *36*, *40*, 45, *51–54*, *59*, *67*
plot.bammshifts, *18*, *22*, *27*, 50, *54*, *70*
plot.credibleshiftset, *18*, *25*, *27*, 52
plotPrior, 54
plotRateThroughTime, *38*, *39*, 55, *75*
points, *5*
primates (BAMMtools-data), 11
print.credibleshiftset
(summary.credibleshiftset), 69

ratesHistogram, 58
rich.colors, *12*, *60*
richColors, 60

samplingProbs, 60
setBAMMpriors, 62
speciesByRatesMatrix, 63
stepBF, 65
strapp (traitDependentBAMM), 72
subsetEventData, 66
subtreeBAMM, *37*, *49*, 67, *74–76*
summary.bammdata, *33*, 68
summary.credibleshiftset, *18*, *27*, 69

testTimeVariableBranches, 70
traitDependentBAMM, 72
traits.fishes (BAMMtools-data), 11
transparentColor, 75

whales (BAMMtools-data), 11
writeEventData, 76