Package ‘paleotree’

January 26, 2012

Type Package

Version 1.0

Title Analyses for Paleontological Phylogenies

Date 2012-01-17

Author David Bapst

Depends ape, phangorn, geiger

Suggests paleoTS

Maintainer David Bapst <dwbapst@gmail.com>

Description Analyzes, time-scales and simulates phylogenies of extinct/fossil lineages. Also plots diversity curves for stratigraphic range data and phylogenies, including combinations of the those two data types.

License GPL version 2 or newer

URL http://home.uchicago.edu/~dwbapst/

R topics documented:

`paleotree-package` *paleotree: Analyses for Paleontological Phylogenies*

Description

Analyzes, time-scales and simulates phylogenies of extinct/fossil lineages. Also plots diversity curves for stratigraphic range data and phylogenies, including combinations of the those two data types.

Details

Package:	paleotree
Type:	Package
Version:	1.0
Date:	2012-01-18
License:	What license is it under?

This package contains functions for analyzing sampling rates given ranges of fossil taxa, in both continuous and discrete time, functions for time-scaling phylogenies of fossil taxa and functions for simulating the fossil record in both a taxic and phylogenetic varieties and converting between these.

Author(s)

David W. Bapst

Maintainer: David W. Bapst <dwbapst@gmail.com>

References

The SRC method of time-scaling will be fully described in an upcoming paper (Bapst, in prep).

See Also

This package relies extensively on the phylogenetic toolkit offered by the 'ape' package. ape

Examples

```
##Simulate some fossil ranges with simFossilTaxa()
##Simulate some fossil ranges with simFossilTaxa()
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#let's see what the 'true' diversity curve looks like in this case
#plot the FADs and LADs with taxicDivCont()
taxicDivCont(taxa[,3:4])
```

```
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont<-sampleRanges(taxa,r=0.5)
#plot the diversity curve based on the sampled ranges
layout(matrix(1:2,,2))
taxicDivCont(rangesCont)

#Now let’s use binTimeData() to bin in intervals of 1 time unit
rangesDisc<-binTimeData(rangesCont,int.length=1)
#plot with taxicDivDisc()
taxicDivDisc(rangesDisc)
#compare to the continuous time diversity curve

#taxa2phylo assumes we know speciation events perfectly... what if we don’t?
#first, let’s use taxa2cladogram() to get the ’ideal’ cladogram of the taxa
cladogram<-taxa2cladogram(taxa,plot=TRUE)
#Now let’s try timePaleoPhy() using the continuous range data
ttree<-timePaleoPhy(cladogram,rangesCont,type="basic",plot=TRUE)
#plot diversity curve
phyloDiv(ttree,drop.ZLB=TRUE)

#that tree lacked the terminal parts of ranges (tips stops at the taxon FADs)
#let’s add those terminal ranges back on with add.term
ttree<-timePaleoPhy(cladogram,rangesCont,type="basic",add.term=TRUE,plot=TRUE)
#plot diversity curve
phyloDiv(ttree)
```

binTimeData *Bin Temporal Ranges in Discrete Intervals*

Description

Converts a matrix of continuous-time first occurances and last occurances for fossil taxa into first and last occurances given in discrete-time intervals, along with information the dates of the given intervals.

Usage

```
binTimeData(timeData, int.length = 1, start = NA)
```

Arguments

`timeData`	Two-column matrix of first and last occurrances in absolute continous time
`int.length`	Time Interval Length
`start`	The starting time for calculating the intervals.

Details

This function takes a matrix of per-taxon first and last occurrances and divides those occurances into time intervals, each interval as long as int.length.

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

The numbering of intervals given in the output increase with time, however, as these are in relative time.

Value

a list containing:

`int.times`	A 2 column matrix with the start and end times of the intervals used.
`taxon.times`	A 2 column matrix with the first and last occurances of taxa in the intervals used, with numbers referring to the row of int.times.

Author(s)

David W. Bapst

See Also

simFossilTaxa,sampleRanges,taxicDivCont

Examples

```
#Simulate some fossil ranges with simFossilTaxa()
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont<-sampleRanges(taxa,r=0.5)
#Now let's use binTimeData() to bin in intervals of 1 time unit
rangesDisc<-binTimeData(rangesCont,int.length=1)
#plot with taxicDivDisc()
taxicDivDisc(rangesDisc)
```

ConvSampEst *Converting Sampling Estimates*

Description

Various functions for converting between various sampling estimates

Usage

```
sProb2sRate(R, int.length = 1)
sRate2sProb(r, int.length = 1)
pqsRate2sProb(r, p, q, int.length = 1)
qsProb2Comp(R, q)
qsRate2Comp(r, q)
probAnc(p, q, R)
```

Arguments

`R`	Per-interval probability of sampling
`r`	Instantaneous rate of sampling
`p`	Instantaneous rate of speciation (lambda)
`q`	Instantaneous rate of extinction (mu)
`int.length`	Length of Time Intervals

Details

This is a family of functions which all convert from some estimate of sampling to another estimate of sampling. Some of these also require estimates of the diversification rate. Diversification rates should always be the instanteous rate (often called the per-capita rates by paleontologists).

As with many models used in the paleotree library, it is generally assumed that the fossil record of interest is composed of discrete relatively-static taxonomic units which diversify mainly by budding cladogenesis, and that sampling is a very rare event well approximated by a model of exponential waiting times. The veracity of those assumptions is difficult to test and the senstivity of these analyses to relaxing those assumptions probably varies.

sProb2sRate() and sRate2sProb() give very rough conversions for converting between the probability of sampling once per time interval (R or "sProb" in this package as used in the references below) and the instantaneous rate of sampling per lineage/time unit ("sRate" or r). If you have estimates of the speciation and extinction rate, use pqsRate2sProb() instead for a more accurate estimate of R.

qsProb2Comp() and qsRate2Comp() are different calculations for "Pp" or the probability/proportion of taxa sampled in a clade. Theoretically, one could use it to extrapolate out the 'true' diversity, assuming the sampling rate model was correct. See Foote and Raup, 1996.

probAnc() obtains the probability of sampling indirect ancestors for a group as given by Foote (1996), assuming budding cladogenesis, given the sampling probability and estimates of the diversification rate.

See the references below for a more detailed explanation of the methods and formulae used. The relevant equations are generally in the appendices.

Value

The converted sampling estimate, depending on the function used. See details above.

Author(s)

David W. Bapst, with assistance from Michael Foote

References

Foote, M. 1996. On the Probability of Ancestors in the Fossil Record. Paleobiology 22(2):141-151.

Foote, M. 1997. Estimating Taxonomic Durations and Preservation Probability. Paleobiology 23(3):278-300.

Foote, M. 2000. Origination and extinction components of taxonomic diversity: general problems. Pp. 74-102. In D. H. Erwin, and S. L. Wing, eds. Deep Time: Paleobiology's Perspective. The Paleontological Society, Lawrence, Kansas.

Foote, M., and D. M. Raup. 1996. Fossil preservation and the stratigraphic ranges of taxa. Paleobiology 22(2):121-140.

Solow, A. R., and W. Smith. 1997. On Fossil Preservation and the Stratigraphic Ranges of Taxa. Paleobiology 23(3):271-277.

See Also

`sampleRanges`,`getSampRateCont`,`getSampProbDisc`

Examples

```
sRate2sProb(r=0.5)
sProb2sRate(R=0.1)
pqsRate2sProb(r=0.5,p=0.1,q=0.1)
qsProb2Comp(R=0.1,q=0.1)
qsRate2Comp(r=0.1,q=0.1)
probAnc(p=0.1,q=0.1,R=0.5)
```

---

degradeTree *Randomly collapse nodes on a phylogeny*

Description

This function removes a proportion of the total nodes in a tree, chosen randomly, collapsing the nodes such that it causes the tree to be more unresolved.

Usage

```
degradeTree(tree, prop_collapse, node.depth = NA)
```

Arguments

tree	A tree of class phylo
prop_collapse	Proportion of nodes to collapse
node.depth	Conditions depth of nodes removed. Default is NA (no conditioning).

Details

This function can be conditioned to remove nodes of a particular depth with greater probability by setting node.depth to some value between zero (deep nodes close to the root) or one (shallow nodes far from the root).

Value

Returns the modified tree as an object of class phylo

Author(s)

David W. Bapst

See Also

di2multi

Examples

```
set.seed(444)
tree<-rtree(100)
tree1<-degradeTree(tree,0.5)
layout(matrix(1:2,,2))
plot(tree,show.tip.label=FALSE,use.edge.length=FALSE)
plot(tree1,show.tip.label=FALSE,use.edge.length=FALSE)
```

depthRainbow *Paint Tree Depth by Color*

Description

Paints the edges of a phylogeny with colors relative to their depth.

Usage

```
depthRainbow(tree)
```

Arguments

tree	A phylo object

Details

Depth is defined relative to as number of branching nodes awar from the root, not absolute distance or distance from the tips.

Value

No value returned, just plots a colorful phylogeny.

Author(s)

David W. Bapst

Examples

```
set.seed(444)
tree<-rtree(1000)
depthRainbow(tree)
```

DiversityCurves *Diversity Curves*

Description

Plots diversity curves based on taxic range data (in both discrete and continuous time) and for phylogenies.

Usage

```
taxicDivCont(timeData, int.length = 1, int.times = NULL, plot = T, output = F)
taxicDivDisc(timeList, int.times = NULL, plot = T, output = F)
phyloDiv(ttree, int.length = 1, int.times = NULL, output = F, plot = T, plotLogRich = F, drop.ZL
```

Arguments

`timeData` Two-column matrix giving the per-taxon first and last appearances in absolute time

`timeList` A list giving interval data. See details below.

`ttree` a time-scaled phylogeny object of class phylo

`int.length` The length of intervals used to make the diversity curve. Ignored if int.times is given.

`int.times` A two-column matrix of the interval start and end times for calculating the diversity curve. If NULL, calculated internally.

`output` If true, per-interval diversity data matrix is output, see details below

`plot` If true, diversity curve is plotted

`plotLogRich` If true, taxic diversity plotted on log scale

`drop.ZLB` If true, zero-length terminal branches are dropped from the input tree

Details

Background: Diversity curves are just plots of how species richness changes over time for a particular group of organisms. In the fossil record, these are based on per-taxon range data while more recently molecular phylogenies have been used to plot lineage-through-time plots (LTTs). Neither of these are without their weaknesses; reconstructing the true history of biodiversity is exceedingly difficult for any group.

The diversity curves produced by these functions will always measure diversity within binned time intervals. For continuous-time data or phylogenies, one could decrease the int.length used to get what is essentially almost an 'instantaneous' estimate of diversity. This is warned against, however, as most historical diversity data will have some time-averaging or uncertain temporal resolution and thus should not be used to calculate instantaneous estimates of diversity.

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

As diversity is counted within binned intervals, standing diversity may be somewhat lower than the measured quantity, particularly if intervals longer than the mean duration of taxa is used. This will be an issue with all diversity curve functions, but particularly the discrete-time variant.

The timeList object should be a list composed of two matrices, one giving by-interval start and end times (in absolute time), the other giving by-taxon first and last appearances in those intervals, numbered as the rows. Absolute time should be decreasing, while the intervals should be numbered so that the number increases with time. Unlike getSampProbDisc, the intervals can be overlapping. See the function of binTimeData for more information.

phyloDiv will resolve polytomies to be dichotomous nodes seperated by zero-length branches prior to calculating the diversity curve. There is not option to change this, but it should not affect the use of the function. phyloDiv will also drop zero-length terminal branches, as with the function dropZLB. This is done by default but can be turned off by setting the argument drop.zlb to false.

Value

If output=TRUE, then these functions will output a three-column matrix, where the first two columns are interval start and end times and the third column is the number of taxa/lineages counted in that interval.

Author(s)

David W. Bapst

See Also

multiDiv,ltt.plot,

Also see the function ltt() in the package phytools. The packages TreeSim and LASER also have similar functions.

Examples

```
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#let's see what the 'true' diversity curve looks like in this case
#plot the FADs and LADs with taxicDivCont()
taxicDivCont(taxa[,3:4])
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont<-sampleRanges(taxa,r=0.5)
#plot the diversity curve based on the sampled ranges
layout(matrix(1:2,,2))
taxicDivCont(rangesCont)
#Now let's use binTimeData() to bin in intervals of 1 time unit
rangesDisc<-binTimeData(rangesCont,int.length=1)
#plot with taxicDivDisc()
taxicDivDisc(rangesDisc)
#compare to the continuous time diversity curve
#Now let's make a tree using taxa2phylo()
tree<-taxa2phylo(taxa,obs_time=rangesCont[,2])
phyloDiv(tree)

#a simple example with phyloDiv
set.seed(444)
tree<-rtree(100)
phyloDiv(tree)

#a neat example of using phyDiv with timeSliceTree to simulate doing molecular-phylogeny studies of diverif
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
taxicDivCont(taxa[,3:4])
#that's the whole diversity curve
#with timeSliceTree we could look at the lineage accumulation curve we'd get of species sampled at a point
tree<-taxa2phylo(taxa)
#use timeSliceTree to make tree of relationships up until time=950
tree950<-timeSliceTree(tree,sliceTime=950,plot=TRUE,drop.extinct=FALSE)
#use drop.extinct=T to only get the tree of lineages extant at time=950
tree950<-timeSliceTree(tree,sliceTime=950,plot=TRUE,drop.extinct=TRUE)
#now its an ultrametric tree with many fewer tips...
#lets plot the lineage accumulation plot on a log scale
phyloDiv(tree950,plotLogRich=TRUE)
```

dropZLB *Drop Terminal Zero-Length Branches*

Description

Drops all terminal branches and tip nodes of a tree that is zero-length. Useful for paleo-trees, as ancestral taxa will often come out as these "ZLBS".

Usage

```
dropZLB(tree)
```

Arguments

`tree`	A phylogeny as a phylo object

Details

This function drops tip-taxa that are attached to the tree as zero-length branches ("ZLBs"). This is advised for analyses of diversification/diversity, as these will appear as instanteous speciation+extinction events, when in reality they probably represent 'pseudo-speciation' (also referred to as anagenesis).

Note this function only drops terminal tips that are on a ZLB. If you want to drop internal zero-length branches, see `di2multi`

If the tree given to the function as a $root.time element, as expected for most paleo-tree objects handled by this library, that root.time is adjusted if the relative time of the root divergence changes when ZLBs are dropped.

Value

Gives back a modified phylogeny as a phylo object

Author(s)

David W. Bapst

See Also

`phyloDiv`,`drop.tip`

Examples

```
##Simulate some fossil ranges with simFossilTaxa()
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont<-sampleRanges(taxa,r=0.5)
#Now let's make a tree using taxa2phylo()
tree<-taxa2phylo(taxa,obs_time=rangesCont[,2])
layout(matrix(1:2,,2))
plot(ladderize(tree))
plot(ladderize(dropZLB(tree)))
#compare the two trees
```

`expandTaxonTree` *Extrapolating Lower Taxon Phylogenies from Higher Taxon Trees*

Description

This tree takes a tree composed of higher taxa and a vector of lower taxa belonging to the first and produces a tree composed of the lower taxa, with the higher taxa treated as unresolved polytomies. A user can also mark higher taxa as paraphyletic and collapse them so they are not monophyetic in the outcome.

Usage

```
expandTaxonTree(taxonTree, taxaData, collapse = NULL, plot = F)
```

Arguments

`taxonTree`	A phylo object where tips represent higher taxa
`taxaData`	A character vector of higher taxa, with elements names equal to the lower taxa.
`collapse`	A character vector of non-monophyletic higher taxa to be collapsed
`plot`	If true, plots a comparison between input and output trees

Details

The result tree will probably be a rough unresolved view of the relationships among the taxa, full of soft polytomies. This is somewhat similar to the methods used in Webb et al. (2006) and Friedman (2009). Obviously, buyer beware: any analyses one wishes to do should be done by resolving this tree with `multi2di` in the ape package.

The taxaData vector should have one element per lower taxon that is to be added to the tree. The names of each elements should be the desired names for the tips of the resulting lower-taxon tree that is output by this function.

The collapse vector should be used if there is doubt about the monophyly of higher taxon on the input taxontree.

The output tree will always lack branch lengths and thus will not be time-scaled.

Value

Outputs the modified tree as an object of class phylo, with the higher taxa expanded into polytomies.

Author(s)

David W. Bapst

References

Friedman, M. 2009. Ecomorphological selectivity among marine teleost fishes during the end-Cretaceous extinction. Proceedings of the National Academy of Sciences 106(13):5218-5223.

Webb, C. O., and M. J. Donoghue. 2005. Phylomatic: tree assembly for applied phylogenetics. Molecular Ecology Notes 5(1):181-183.

See Also

multi2di,bind.tree,

Examples

```
set.seed(444)
#lets make our hypothetical simulated tree of higher taxa
taxtr<-rtree(10)
taxd<-sample(taxtr$tip.label,30,replace=TRUE) #taxa to place within higher taxa
names(taxd)<-paste(taxd,"_x",1:30,sep="")
coll<-sample(taxtr$tip.label,3) #what to collapse?
expandTaxonTree(taxonTree=taxtr,taxaData=taxd,collapse=coll,plot=TRUE)
```

getSampProbDisc *Fit Models of Sampling Probability to Discrete-Interval Taxon Ranges*

Description

Uses ML to find the best-fit parameters for models of sampling probability and extinction rate, given a set of discrete-interval taxon ranges from the fossil record. This function can fit models where there are different groupings of taxa with different parameters and different free-floating time intervals with different parameters.

Usage

```
getSampProbDisc(timeData, n_tbins = 1, grp1 = NA, grp2 = NA, est_only = F, throttle = 1)
```

Arguments

timeData	A 2 column matrix with the first and last occurances of taxa given in relative time intervals OR the output of binTimeData
n_tbins	Number of time bins with different sampling/extinction parameters
grp1	A vector, the same length as the number of taxa in timeData, each element is a different identified for the group ID of included taxa
grp2	A vector, the same length as the number of taxa in timeData, each element is a different identified for the group ID of included taxa
est_only	If true, function will give back a matrix of ML extinction rates and sampling probabilities per species rather than usual output (see below)
throttle	Controls the tradeoff between run-time and accuracy

Details

This function uses maximum-likelihood solutions found by Foote (1997). These analyses are ideally applied to data from single stratigraphic section but can potentially be applicable to regional or global datasets (Foote and Raup, 1996, tested the method using Alroy's North American mammal data), although their behavior for those datasets is less well understood.

This function allows for a considerable level of versatility in terms of the degree of variation allowed among taxa in sampling rates. Essentially, this function allows taxa to be broken down into different possibly overlapping classes which have 'average' parameter values that are then combined to calcualte per-taxon parameters. For example, perhaps I think that taxa that live in a particular environment have a different characteristic sampling rate/probability, taxa of several different major clades have different characteristic sampling parameters and that there may be several temporal shifts in the characteristic extinction rate or sampling parameters. The classification IDs for the first two can be included as grp1 and grp2 and the hypothesized number of temporal breaks can be included as the n_tbins argument. A model where taxa differ in parameters across time, clades and environments will then be fit and the AIC calculated, so that it can be compared to other models.

By default, the simple model where all taxa belong to a single class, with a single characteristic extinction rate and a single characteristic sampling parameter, is fit to the range data.

The timebins option will always allow for timebins with free-floating boundaries that are not defined a priori. The boundaries between time bins with different characteristic parameters will thus be additional parameters included in the AIC calculation. If you have the prior inclination that sampling/extinction changed at a particular point in time, then seperate the taxa that originated before and after that point as two different groups and include those classifications as a grp in the arguments.

This function does not implement the finite window of observation modification for dealing with data that leads up to the recent. This is planned for a future update, however. Thus, data input into this should be for taxa that have already gone extinct by the present and are not presently extant.

The timeData should be non-overlapping sequential intervals of roughly equal length. They should be in relative time, so the earliest interval should be 1 and it should increase as the intervals go up with age. This is so differences in interval numbers represents the same rough difference in interval timing. For example, taxa with listed as from North American Mammal assemblage zones, microfossil faunal zones or graptolite biozones can be given as long as they are correctly numbered in sequential. As a counter example, taxa resolved only to the Mesozoic and taxa resolved to the Turinean should not be included in the same input. I reccommend droping the taxa from less poorly resolved intervals.

The optimization of the maximum-likelihood solution for sampling probability is computationally slow because completeness (Pp in Foote, 1997) must be calculated for every parameter combination to obtain the likelihood, and that operation requires summation over a discrete probability distribution. To deal with this, by default this program will ignore the tails of the distributions, which decreases run time but also decreases accuracy of the likelihood calculation. For very flat likelihood surfaces, the accuracy of the likelihood calculation can be increased by increasing the threshold (I recommend a maximum threshold of 4). The default threshold is 1.

Please check the $message element of the output to make sure that convergence occurred.

Value

If est_only=T, a matrix of per-taxon sampling and extinction parameters is output.

If est_only=F (default), then the output is a list:

`Title` Gives details of the analysis, such as the number and type of parameters included and the number of taxa analyzed

`pars` Maximum-likelihood parameters of the sampling model

`SMax` The maximum support (log-likelihood) value

`AICc` The second-order Akaike's Information Criterion, corrected for small sample sizes

`message` Messages output by optim(); check to make sure that model convergence occurred

If the multi-class models are using, the element $pars will not be present, but there will be several different elements that sum the characteristic parameter components for each class. As noted in the $title, these should not be interpretated as the actual rates/probabilities of any real taxa but rather as components which must be assessed in combination with other classes to be meaningful. For example, for taxa of a given group in a given time bin, their extinction rate is the extinction rate component of that time bin times the extinction rate component of their group.

Completeness estimates are only output when model classes are not overlapping (and thus 'meaningful').

Author(s)

David W. Bapst

References

Foote, M. 1997. Estimating Taxonomic Durations and Preservation Probability. Paleobiology 23(3):278-300.

Foote, M., and D. M. Raup. 1996. Fossil preservation and the stratigraphic ranges of taxa. Paleobiology 22(2):121-140.

See Also

getSampRateCont,sProb2sRate,

Examples

```
#Simulate some fossil ranges with simFossilTaxa()
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont<-sampleRanges(taxa,r=0.5)
#Now let's use binTimeData() to bin in intervals of 1 time unit
rangesDisc<-binTimeData(rangesCont,int.length=1)
#now, get an estimate of the sampling rate (we set it to 0.5 above)
#we can do something very similar for the discrete time data (can be a bit slow)
SPres1<-getSampProbDisc(rangesDisc)
sProb<-SPres1[[2]][2]
print(sProb)
#note that that's the sampling PROBABILITY per bin, not the instantaneous rate of change
#we want the rate for srcTimePaleoPhy()
#we can use sProb2sRate() to get the rate. We'll need to also tell it the int.length
sRate<-sProb2sRate(sProb,int.length=1)
print(sRate)
#estimates that r=0.3... kind of low... (simulated sampling rate is 0.5)
## Not run:
#Note: for real data, you may need to use an average int.length (no constant length)
#this data was simulated under homogenous sampling rates, extinction rates
#if we fit a model with random groups and allow for multiple timebins, AIC should be higher (less informat
randomgroup<-sample(1:2,nrow(rangesDisc[[2]]),replace=TRUE)
SPres2<-getSampProbDisc(rangesDisc,grp1=randomgroup)
SPres3<-getSampProbDisc(rangesDisc,n_tbins=2)
print(c(SPres1$AICc,SPres2$AICc,SPres3$AICc))
#and we can see the most simple model has the lowest AICc (most informative model)

## End(Not run)
```

getSampRateCont *Fit Models of Sampling Rates to Continuous-Time Taxon Ranges*

Description

Uses ML to find the best-fit parameters for models of sampling and extinction rates, given a set of continuous-time taxon ranges from the fossil record

Usage

```
getSampRateCont(timeData, n_tbins = 1, grp1 = NA, grp2 = NA, threshold = 0.1, est_only = F)
```

Arguments

`timeData`	Two-column matrix of per-taxon first and last occurrances in absolute continous time
`n_tbins`	Number of time bins with different sampling/extinction parameters
`grp1`	A vector, the same length as the number of taxa in timeData, each element is a different identified for the group ID of included taxa
`grp2`	A vector, the same length as the number of taxa in timeData, each element is a different identified for the group ID of included taxa
`threshold`	The smallest allowable range. See below.
`est_only`	If true, function will give back a matrix of ML extinction rates and sampling probabilities per species rather than usual output (see below)

Details

This function uses maximum-likelihood solutions found by Foote (1997). These analyses are ideally applied to data from single stratigraphic section but can potentially be applicable to regional or global datasets (Foote and Raup, 1996, tested the method using Alroy's North American mammal data), although their behavior for those datasets is less well understood.

This function allows for a considerable level of versatility in terms of the degree of variation allowed among taxa in sampling rates. Essentially, this function allows taxa to be broken down into different possibly overlapping classes which have 'average' parameter values that are then combined to calcualte per-taxon parameters. For example, perhaps I think that taxa that live in a particular environment have a different characteristic sampling rate/probability, taxa of several different major clades have different characteristic sampling parameters and that there may be several temporal shifts in the characteristic extinction rate or sampling parameters. The classification IDs for the first two can be included as grp1 and grp2 and the hypothesized number of temporal breaks can be included as the n_tbins argument. A model where taxa differ in parameters across time, clades and environments will then be fit and the AIC calculated, so that it can be compared to other models.

By default, the simple model where all taxa belong to a single class, with a single characteristic extinction rate and a single characteristic sampling parameter, is fit to the range data.

The timebins option will always allow for timebins with free-floating boundaries that are not defined a priori. The boundaries between time bins with different characteristic parameters will thus be additional parameters included in the AIC calculation. If you have the prior inclination that sampling/extinction changed at a particular point in time, then seperate the taxa that originated before and after that point as two different groups and include those classifications as a grp in the arguments.

This function does not implement the finite window of observation modification for dealing with data that leads up to the recent. This is planned for a future update, however. Thus, data input into this should be for taxa that have already gone extinct by the present and are not presently extant.

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

Please check the $message element of the output to make sure that convergence occurred.

Value

If est_only=T, a matrix of per-taxon sampling and extinction parameters is output.

If est_only=F (default), then the output is a list:

`Title`	Gives details of the analysis, such as the number and type of parameters included and the number of taxa analyzed
`pars`	Maximum-likelihood parameters of the sampling model
`SMax`	The maximum support (log-likelihood) value
`AICc`	The second-order Akaike's Information Criterion, corrected for small sample sizes
`message`	Messages output by optim(); check to make sure that model convergence occurred

If the multi-class models are using, the element $pars will not be present, but there will be several different elements that sum the characteristic parameter components for each class. As noted in the $title, these should not be interpretated as the actual rates/probabilities of any real taxa but rather as components which must be assessed in combination with other classes to be meaningful. For example, for taxa of a given group in a given time bin, their extinction rate is the extinction rate component of that time bin times the extinction rate component of their group.

Completeness estimates are only output when model classes are not overlapping (and thus 'meaningful').

Author(s)

David W. Bapst

References

Foote, M. 1997. Estimating Taxonomic Durations and Preservation Probability. Paleobiology 23(3):278-300.

Foote, M., and D. M. Raup. 1996. Fossil preservation and the stratigraphic ranges of taxa. Paleobiology 22(2):121-140.

See Also

getSampProbDisc,sRate2sProb,

Examples

```
##Simulate some fossil ranges with simFossilTaxa()
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont<-sampleRanges(taxa,r=0.5)
#now, get an estimate of the sampling rate (we set it to 0.5 above)
(SRres1<-getSampRateCont(rangesCont))
#that's all the results...
sRate<-SRres1$pars[2]
print(sRate) #estimates that sRate=~0.4 (not too bad...)
#this data was simulated under homogenous sampling rates, extinction rates
#if we fit a model with random groups and allow for multiple timebins, AIC should be higher (less informat
randomgroup<-sample(1:2,nrow(rangesCont),replace=TRUE)
SRres2<-getSampRateCont(rangesCont,grp1=randomgroup)
SRres3<-getSampRateCont(rangesCont,n_tbins=2)
SRres4<-getSampRateCont(rangesCont,n_tbins=3,grp1=randomgroup)
print(c(SRres1$AICc,SRres2$AICc,SRres3$AICc,SRres4$AICc))
#and we can see the most simple model has the lowest AICc (most informative model)
```

multiDiv *Calculating Diversity Curves Across Multiple Datasets*

Description

Given a list of data, such as taxic range and phylogenetic trees, diversity curves are calculated for the same intervals across all the individual datasets. A median curve with 95 percent quantile bounds can be calculated and plotted.

Usage

```
multiDiv(data, int.length = 1, plot = T, output=F, drop.ZLB=T)
plotMultiDiv(results)
```

Arguments

data	A list where each element is a dataset formatted to be input in one of the diversity curve functions in this package
int.length	Time interval length
plot	Should the median diversity curve be plotted
output	Should the median diversity curve be output?
drop.ZLB	Should terminal zero-length branches be dropped on phylogenetic datasets?
results	The output of a previous run of multiDiv

Details

This function is essentially a wrapper for the individual diversity curve functions in this package. It will intuitively decide whether datasets it is given are continuous-time taxic ranges, discrete-time (binned interval) taxic ranges or phylogenetic trees, as long as they are formatted as the individual functions request. A list that contains a mix of data types is entirely acceptable.

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

The 'averaged' curve is actually the median rather than the mean as diversity counts are often highly skewed (in my experience).

The shaded certainty region around the median curve is the two-tailed 95 percent lower and upper quantiles, calculated from the observed data. It is not a true confidence interval, as it has no relaitonship to the standard error.

Value

A list composed of three elements:

`int.times`	A two column matrix giving interval start and end times
`div`	A matrix of measured diversities in particular intervals by rows, with each column representing a different dataset included in the input
`median.curve`	A three column matrix, where the first column is the calculated median curve and the second and third columns are the 95 percent quantile upper and lower bounds

Author(s)

David W. Bapst

See Also

The diversity curve functions used: `phyloDiv`,`taxicDivCont`,`taxicDivDisc`,

Also see the function LTT.average.root() in the package TreeSim, which calculates an average LTT curve for multiple phylogenies.

Examples

```
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
rangesCont<-sampleRanges(taxa,r=0.5)
rangesDisc<-binTimeData(rangesCont,int.length=1)
cladogram<-taxa2cladogram(taxa,plot=TRUE)
#using multiDiv with very different data types
ttree<-timePaleoPhy(cladogram,rangesCont,type="basic",add.term=TRUE,plot=FALSE)
multiDiv(list(rangesCont,rangesDisc,ttree),plot=TRUE,output=FALSE)
#using multiDiv with samples of trees
ttrees<-timePaleoPhy(cladogram,rangesCont,type="basic",ntrees=10,add.term=TRUE,plot=FALSE)
multiDiv(ttrees)
```

`plotTraitgram` *Plot a Traitgram for Continuous Traits*

Description

This function uses maximum-likelihood ancestral trait estimation to plot a 'traitgram' (Ackerly, 2009) given a tree and a set of continuous trait values.

Usage

```
plotTraitgram(trait, tree, trait.name = "’trait’", conf.int = T, lwd = 1.5)
```

Arguments

`trait`	a vector of continuous trait values
`tree`	a phylo object
`trait.name`	The name of the trait plotted, used for the figure’s main title
`conf.int`	if true, confidence intervals are plotted
`lwd`	The line width used for the figure

Details

By default, this function uses `ace` from the library ape to reconstruct ancestral traits and confidence intervals with maximum likelihood. If confidence intervals are turned off, a somewhat faster function `getAncStates` from geiger is used.

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

Value

Returns no value, just plots the traitgram.

Note

One should probably never do ancestral trait estimation without looking at the confidence intervals, as estimate tend to be very uncertain.

Author(s)

David W. Bapst

References

Ackerly, D. 2009. Conservatism and diversification of plant functional traits: Evolutionary rates versus phylogenetic signal. Proceedings of the National Academy of Sciences 106(Supplement 2):19699-19706.

See Also

`ace`,`getAncStates`,

Also see the function traitgram in the library picante.

Examples

```
require(geiger)
set.seed(444)
tree<-rtree(10)
trait<-rTraitCont(tree)
plotTraitgram(trait,tree)
```

sampleRanges *Sampling Taxon Ranges*

Description

A function for simulating the effect of incomplete sampling of the fossil record.

Usage

```
sampleRanges(taxad, r, min.taxa = 2, ranges.only = T, modern.samp = T)
```

Arguments

`taxad`	A five column matrix, as output by simFossilTaxa, or a two-column matrix of per-taxon ranges
`r`	Instantaneous sampling rate per time unit
`min.taxa`	Minimum number of taxa sampled
`ranges.only`	Give time of all sampling events or ranges only?
`modern.samp`	Always perfectly sample living taxa at the present day (time=0)?

Details

Currently, this function only implements the simplest of sampling models, where sampling events occur under a homogenous rate through time, a homogenous rate across lineages and the waiting times between them are exponentially distributed across lineage*time-units (Foote, 1997). Future modifications of this function will include more sophisticated options, such as the hat model (Liow et al., 2010).

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

If min.taxa=0, the simulation may output ranges for which no taxa were ever sampled.

If modern.samp=T, then living taxa will always be sampled at least at the present day (if there are any living taxa). If not true, then it is assumed that almost nothing is known about the living taxa. This is to simulate the fact that we generally assume that is the modern day is much better sampled than the fossil record, although probably not perfectly. Presently, there is no option for increased sampling at the present-day.

Value

If ranges.only=T, then the output is a two-column per-taxon matrix of first and last appearances in absolute time. NAs mean the taxon was never sampled in the simulation.

If ranges.only=F, then the output is a list, where each element is a vector of sampling events the timing of sampling events, each corresponding to a different taxon in the input. Elements that are NA are unsampled taxa.

Author(s)

David W. Bapst

References

Foote, M. 1997. Estimating Taxonomic Durations and Preservation Probability. Paleobiology 23(3):278-300.

Liow, L. H., T. B. Quental, and C. R. Marshall. 2010. When Can Decreasing Diversification Rates Be Detected with Molecular Phylogenies and the Fossil Record? Systematic Biology 59(6):646-659.

See Also

`simFossilTaxa`,`binTimeData`

Examples

```
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#let's see what the 'true' diversity curve looks like in this case
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont<-sampleRanges(taxa,r=0.5)
#plot the diversity curve based on the sampled ranges
taxicDivCont(rangesCont)
```

`simFossilTaxa` *Simulating Taxa in the Fossil Record*

Description

Functions for simulating taxon ranges and relationships under various models of evolution

Usage

```
simFossilTaxa(p, q, w = 0, u = 0, nruns = 1, mintaxa = 1, maxtaxa = 1000, maxtime = 100, nExtant
simFossilTaxa_SRCond(r, avgtaxa, p, q, w = 0, u = 0, nruns = 1, maxtime = 100, nExtant = NULL, p
```

Arguments

`p`	Instantaneous rate of speciation/branching
`q`	Instantaneous rate of extinction
`w`	Instantaneous rate of pseudoextinction/anagensis
`u`	Proportion of branching by bifurcating cladogenesis relative to budding cladogenesis
`nruns`	Number of datasets to be output
`mintaxa`	Minimum number of total taxa over the entire history of a clade necessary for a dataset to be accepted
`maxtaxa`	Maximum number of total taxa over the entire history of a clade necessary for a dataset to be accepted
`maxtime`	Maximum time units to run any given simulation before stopping it
`nExtant`	Maximum number of living taxa allowed in simulations
`plot`	Plot the diversity curves of the accepted datasets as they are simulated?
`r`	Instantaneous sampling rate per time unit
`avgtaxa`	Desired average number of taxa

Details

This function will simulate the diversification of clades where taxa are relatively morphologically static over long time intervals.

Simulations will stop when various conditions are met (generally maxtaxa or maxtime). To reduce the effect of one condition, simply set the limit to an arbitrarily high number. Warning, this may cause the function to enter an unending loop. Hartmann et al. (2011) recently discovered a potential statistical artifact when branching simulations are conditioned on some maximum number of taxa. Thus, this function continues the simulation once maxtaxa is hit, until the next taxon (maxtaxa+1) originates. Once the simulation terminates, it is judged whether it is acceptable for all conditions given and if so, it is accepted as a dataset to be output.

Please note that mintaxa and maxtaxa refer to the number of static morphotaxa that were birthed over the entire history of the simulated clade. Use nExtant if you want to condition on the maximum number of taxa living at some time.

The simFossilTaxa function can effectively simulate clades evolving any combination of the three "modes" of speciation generally referred to by paleontologists: budding cladogenesis, branching cladogenesis and anagenesis (Foote, 1996). The first two are "speciation" in the typical sense used by biologists, with the major distinction between these two modes being whether the ancestral taxon shifts morphologically at the time of speciation. The third is where a morphotaxon changes into another morphotaxon with no branching, hence the use of the terms "pseudoextinction" and "pseudospeciation". As bifurcation and budding are both branching events, both are controlled by the p, the instantaneous rate, while the probability of a branching event being either is set by u. By default, only budding cladogenesis occurs To have these three modes occur in equal proportions, set p to be twice the value of w and set u to 0.5. There is no option for cryptic speciation in this function.

If nExtant is 0, then the function will be limited to only accepting simulations that end in total clade extinction before maxtime.

If conditions are such that a clade survives to maxtime, then maxtime will become the time of first appearance for the first taxa. Unless maxtime is very low, however, it is more likely the maxtaxa limit will be reached first, in which case the point in time at which maxtaxa is reached will become the present data and the entire length of the simulation will be the time of the first appearance of the first taxon.

simFossilTaxa_SRCond is a wrapper for simFossilTaxa for when you want clades of a particular size, post-sampling. This function accomplishes this task by calculating the probability of sampling per-taxon and calculating the average clade size needed to produce the number of sampled taxa given by avgtaxa. We will call that quantity N. Then, it uses simFossilTaxa, with mintaxa set to N and maxtaxa set to 2*N. It will generally produce simulated datasets that are generally of that size or larger post-sampling (although there can be some variance). Some combinations of p, q, r and avgtaxa may take an extremely long time to find large enough datasets. Some combinations may produce very strange datasets that may have weird structure that is only a result of the conditioning (for example, the only clades that have many taxa when net diversification is low or negative will have lots of very early divergences, which could impact analyses). Needless to say, conditioning can be very difficult.

Value

Both of these functions give back a list containing nruns number of taxa datasets. Sampling has not been simulated in the output for either function; the output represents the 'true' history of the simualted clade.

For each dataset, the output is a five column per-taxon matrix where all entries are numbers, with the first column being the taxon ID, the second being the ancestral taxon ID (the first taxon is NA for ancestor), the third column is the first appearance date of a species in absolute time, the fourth column is the last appearance data and the fifth column records whether a species is still extant at the time the simulation terminated (1 means a taxon is still alive).

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

Author(s)

David W. Bapst

References

Foote, M. 1996. On the Probability of Ancestors in the Fossil Record. Paleobiology 22(2):141-151.

See Also

sampleRanges,simPaleoTrees,taxa2phylo,taxa2cladogram,

Examples

```
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#let's see what the 'true' diversity curve looks like in this case
#plot the FADs and LADs with taxicDivCont
taxicDivCont(taxa[,3:4])

set.seed(444)
avgtaxa=50
r<-0.5
taxa<-simFossilTaxa_SRCond(r=r,p=0.1,q=0.1,nruns=20,avgtaxa=avgtaxa)
#sample and count number of taxa
ranges<-lapply(taxa,sampleRanges,r=r)
ntaxa<-sapply(ranges,function(x) sum(!is.na(x[,1])))
#works okay... some parameter combinations are difficult to get right number of taxa
hist(ntaxa);mean(ntaxa)
```

simPaleoTrees *Simulating Un-Conditioned Trees of Fossil Taxa*

Description

Creates sets of paleo-trees with minimal conditioning and with sampling, except node-times are known perfectly

Usage

```
simPaleoTrees(p, q, r, ntrees, nExtant = NULL, maxtime=1000, maxtaxa=3000, modern.samp = T, drop
```

Arguments

`p` Instantaneous rate of speciation per lineage-time units

`q` Instantaneous rate of extinction per lineage-time units

`r` Instantaneous rate of sampling per lineage-time units

`ntrees` Number of trees to simulate

`nExtant` Maximum number of extant lineages

`maxtime` Maximum time to run simulation

`maxtaxa` Maximum number of taxa allowed over entire simulation

`modern.samp` Should taxa extant at present be sampled perfectly at the present-day (time=0)?

`drop.zlb` Should zero-length branches be dropped?

`plot` Should data be plotted as it is simulated?

Details

Essentially a complex wrapper for simFossilTaxa, sampleRanges and taxa2phylo, this function will spit out simulated phylogenies of fossil taxa where the divergence times are perfectly known, only sampled taxa are included on the tree and tips are located at the last observed time for the species (the apparent time of extinction, except for living taxa).

The defaults are set such that there is as minimal conditioning as possible, to get as unbiased a sample of simulations as possible. There is no conditioning on the number of extant taxa, living taxa are sampled perfectly at time 0 and zero-length branches are dropped. maxtaxa and maxtime are as high as possible without decreasing performance.

Because the divergence times are known perfectly, yet tips are at the apparent time of extinction and unsampled taxa are dropped, one should not use the output of this analysis except for very specialized simulation analyses. The results are probably not anything like real data at least in most qualities.

Value

Output is an object of class multiphylo containing the simulated phylogenies.

Author(s)

David W. Bapst

See Also

`simFossilTaxa`,`taxa2phylo`,`sampleRanges`,

Examples

```
#simulate trees with no living descendants
trees<-simPaleoTrees(p=0.1,q=0.1,r=0.5,ntrees=2,nExtant=0,modern.samp=TRUE,drop.zlb=TRUE,plot=TRUE)
sapply(trees,Ntip)
#simulate trees with living descendants
trees<-simPaleoTrees(p=0.1,q=0.1,r=0.5,ntrees=2,nExtant=10,modern.samp=TRUE,drop.zlb=TRUE,plot=TRUE)
sapply(trees,Ntip)
```

srcTimescaling *SampRate-Calibrated Timescaling of Paleo-Phylogenies*

Description

Takes an unscaled cladogram of fossil taxa and information on their ranges and the instantaneous rate of sampling and outputs samples of timescaled trees, as a result of stochastic process that uses the sampling rate to weigh observed gaps in the fossil record. Also can uses the sampling-rate calibrated time-scaling algorithim to resolve polytomies randomly and infer potential ancestor-descendant relationships.

Usage

```
srcTimePaleoPhy(tree, timeData, sampRate, ntrees = 1, anc.wt = 1, rand.obs = F, node.mins = NULL
bin_srcTimePaleoPhy(tree, timeList, sampRate, ntrees = 1, sites = NULL, anc.wt = 1, node.mins =
```

Arguments

`tree` An unscaled cladogram of fossil taxa

`timeData` Two-column matrix of first and last occurrances in absolute continous time, with rownames as the taxon IDs used on the tree

`sampRate` Either a single estimate of the instanteous sampling rate or a vector of per-taxon estimates

`ntrees` Number of time-scaled trees to output

`anc.wt` Weighting against inferring ancestor-descendant relationships

`rand.obs` Should the tips represent observation times uniform distributed within taxon ranges?

`node.mins` Minimum ages of nodes on the tree, see below

`root.max` Maximum time before the first FAD that the root can be pushed back to

`plot` If true, plots the input, "basic" timescaled and output SRC-timescaled phylogenies

`timeList` A list composed of two matrices giving interval times and taxon appearance datums, as would be output by binTimeData. The rownames of the second matrix should be the taxon IDs

`sites` A two column matrix, composed of site IDs for taxon FADs and LADs. Does not have to be given by default; see explanation below.

Details

The sampling-rate calibrated (SRC) algorithim time-scales trees by stochastically picking node divergence times relative to a probability distribution of expected waiting times between speciation and first appearance in the fossil record. This simple idea can also be extend to apply to resolving polytomies and designating possible ancestor-descendant relationships. The full details of this method will be given in a paper currently in prep.

Most importantly, please note the stochastic element of the SRC method. It does not use traditional optimization methods, but instead pulls node times from a distribution. This means analyses MUST be done over many SRC-timescaled trees for analytical rigor! No one tree is correct.

The sampling rate used by SRC methods is the instantaneous sampling rate, as estimated by various other function in the paleotree package. See getSampRateCont for more details. If you have the per-time unit sampling probability ('R' as opposed to 'r') look at the sampling parameter conversion functions also included in this package.

By default, the SRC functions will consider that ancestor-descendant relationships may exist among the given taxa, under a budding cladogenetic or anagenetic modes. Which tips are designated as which is given by two additional elements added to the output tree, $budd.tips (taxa designated as ancestors via budding cladogenesis) and $anag.tips (taxa designated as ancestors via anagenesis).

The argument anc.wt allows users to change the default consideration of anc-desc relationships. This value is used as a multiplier applied to the probability of choosing any node position which would infer an ancestor-descendant relationship. By default, anc.wt=1, and thus these probabilities are unaltered. if anc.wt is less than 1, the probabilities decrease and at anc.wt=0, no ancestor-descendant relationships are inferred at all.

As this function can infer possible anagenetic relationships, this can create zero-length terminal branches. Use dropZLB() to get rid of these before doing analyses of lineage diversification.

Unlike timePaleoPhy, SRC methods will always resolve polytomies (using the sampling-rate calibrated algorithim) and will always add the terminal ranges of taxa. However, because of the ability to infer potential ancestor-descendant relationships, the length of terminal branches may be shorter than taxon ranges themselves, as budding may have occurred during the range of a morphologically static taxon. By resolving polytomies with the SRC method, this function allows for taxa to be ancestral to more than one descendant taxon.

If rand.obs=T, then it is assumed that users wish the tips to represent observations made with some temporal uncertainty, such that they might have come from any point within a taxon's range. This might be the case, for example, if a user is interested in applying phylogeny-based approaches to studying trait evolution, but have per-taxon measurements of traits that come from museum specimens with uncertain temporal placement. When rand.obs=T, the tips are placed randomly within taxon ranges, as if uniformly distributed.

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

These functions will intuitively drop taxa from the tree with NA for range or are missing from timeData.

The minimum dates of nodes can be set using node.mins; this argument takes a vector of the same length as the number of nodes, with dates given in the same order as nodes are number in the tree$edge matrix. Not all nodes need be set; those without minimum dates can be given as NA in node.mins. These nodes will be frozen and will not be shifted by the SRC algorithm. If the dates refer to a polytomy, then the first divergence will be frozen with additional divergence able to occur after the minimum date.

All trees are output with an element $root.time. This is the time of the root on the tree and is important for comparing patterns across trees.

bin_srcTimePaleoPhy is a wrapper of srcTimePaleoPhy which produces timescaled trees for datasets which only have interval data available. For each output tree, taxon FADs and LADs are placed within their listed intervals under a uniform distribution. Thus, a large sample of time-scaled trees will approximate the uncertainty in the actual timing of the FADs and LADs.

The sites argument allows users to constrain the placement of dates in bin_srcTimePaleoPhy by restricting multiple fossil taxa whose FADs or LADs are from the same very temporally restricted sites (such as fossil-rich Lagerstatten) to always have the same date, across many iterations of time-scaled trees from bin_timePaleoPhy. To do this, simply give a matrix where the "site" of each FAD and LAD for every taxon is listed, as corresponding to the second matrix in timeList. If no sites matrix is given (the default), then it is assumed all fossil come from different "sites" and there is no shared temporal structure among the events.

Value

The output of these functions is a time-scaled tree or set of time-scaled trees, of either class phylo or multiphylo, depending on the argument ntrees.

Author(s)

David W. Bapst

References

Bapst, in prep. Time-scaling Trees of Fossil Taxa. To be submitted to Paleobiology.

See Also

timePaleoPhy,binTimeData,getSampRateCont,multi2di

Examples

```
##Simulate some fossil ranges with simFossilTaxa()
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont<-sampleRanges(taxa,r=0.5)
#Now let's use binTimeData() to bin in intervals of 1 time unit
rangesDisc<-binTimeData(rangesCont,int.length=1)
#let's use taxa2cladogram() to get the 'ideal' cladogram of the taxa
cladogram<-taxa2cladogram(taxa,plot=TRUE)
#this library allows one to use SRC type time-scaling methods (Bapst, in prep.)
#to use these, we need an estimate of the sampling rate (we set it to 0.5 above)
SRres<-getSampRateCont(rangesCont)
sRate<-SRres$pars[2]
#now let's try srcTimePaleoPhy(), which timescales using a sampling rate to calibrate
#This can also resolve polytomies based on sampling rates, with some stochastic decisions
ttree<-srcTimePaleoPhy(cladogram,rangesCont,sampRate=sRate,ntrees=1,plot=TRUE)
#notice the warning it gives!
phyloDiv(ttree)

#Again, we would need to set ntrees to a large number to get a fair sample of trees
#can do an example of such an analysis via multDiv
ttrees<-srcTimePaleoPhy(cladogram,rangesCont,sampRate=sRate,ntrees=10,plot=FALSE)
multiDiv(ttrees)

#by default, srcTimePaleoPhy() is allowed to predict indirect ancestor-descendant relationships
#can turn this off by setting anc.wt=0
ttree<-srcTimePaleoPhy(cladogram,rangesCont,sampRate=sRate,ntrees=1,anc.wt=0,plot=TRUE)

#we can do something very similar for the discrete time data (can be a bit slow)
SPres<-getSampProbDisc(rangesDisc)
sProb<-SPres$pars[2]
#but that's the sampling PROBABILITY per bin, not the instantaneous rate of change
#we can use sProb2sRate() to get the rate. We'll need to also tell it the int.length
sRate1<-sProb2sRate(sProb,int.length=1)
#estimates that r=0.3... kind of low (simulated sampling rate is 0.5)
#Note: for real data, you may need to use an average int.length (no constant length)
ttree<-bin_srcTimePaleoPhy(cladogram,rangesDisc,sampRate=sRate1,ntrees=1,plot=TRUE)
phyloDiv(ttree)
```

taxa2cladogram *Converting taxon data into cladogram*

Description

Convert ancestor-descendant relationships of taxa into an ideal unscaled cladogram

Usage

```
taxa2cladogram(taxad, plot = F)
```

Arguments

taxad	A five-column matrix of taxonomic data, as output by simFossilTaxa
plot	Should the result be plotted?

Details

This function takes a matrix that has information on the ancestor-descendant relationships of a bunch of taxa and constructs an unscaled cladogram of the hierarcially-nesting relationships among those taxa.

The result will probably not be fully resolved, as including both ancestor and descendant taxa will make it nearly impossible to produce a fully nesting system of relationships. For example, consider a set of three morphologically-static taxa where the first is an ancestor (either direct or indirect, ala Foote, 1996) of the second and third. This set of taxa cannot be broken up into bifurcating nested relationships and will result in a polytomy for the output of this function. Any set of ancestor-descendant relationships will have many of these, as some ancestors must have more than one descendant for the clade to diversify at all.

Value

The resulting phylogeny without branch lengths is output as an object of class phylo

Author(s)

David W. Bapst

References

Foote, M. 1996. On the Probability of Ancestors in the Fossil Record. Paleobiology 22(2):141-151.

See Also

simFossilTaxa,taxa2phylo,

Examples

```
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#let’s use taxa2cladogram() to get the ’ideal’ cladogram of the taxa
cladogram<-taxa2cladogram(taxa,plot=TRUE)
plot(cladogram)
```

taxa2phylo *Convert taxon data into Phylogeny*

Description

Convert temporal and ancestor-descendant relationships of taxa into a time-scaled phylogeny

Usage

```
taxa2phylo(taxad, obs_time = NULL, plot = F)
```

Arguments

taxad	A five-column matrix of taxonomic data, as output by simFossilTaxa
obs_time	Per-taxon times of observation; if NULL, the LADs (column 4) in taxad2 are used
plot	Plot the resulting phylogeny?

Details

This function will output trees with the element $root.time, which is the time of the root divergence in absolute time.

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

Please, DON'T use this function to time-scale a real tree. It assumes you know the use timePaleoPhy or srcTimePaleoPhy instead.

Please, DO use this function when doing simulations and you want to make a tree of the 'true' history, such as for simulating trait evolution along phylogenetic branches.

Value

The resulting phylogeny with branch lengths is output as an object of class phylo

Author(s)

David W. Bapst

See Also

simFossilTaxa,taxa2cladogram,

Examples

```
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#let's use taxa2cladogram() to get the 'ideal' cladogram of the taxa
tree<-taxa2phylo(taxa)
phyloDiv(tree)
#now a phylogeny with tips placed at the apparent time of extinction
rangesCont<-sampleRanges(taxa,r=0.5)
tree<-taxa2phylo(taxa,obs_time=rangesCont[,2])
phyloDiv(tree)
```

timePaleoPhy *Timescaling of Paleo-Phylogenies*

Description

Takes an unscaled cladogram of fossil taxa and information on their ranges and outputs a timescaled tree using various methods. Also can resolve polytomies randomly and output samples or randomly-resolved trees.

Usage

```
timePaleoPhy(tree, timeData, type = "basic", vartime = NULL, ntrees = 1, randres = F, add.term =
bin_timePaleoPhy(tree, timeList, type = "basic", vartime = NULL, ntrees = 1, randres = F, sites
```

Arguments

`tree` An unscaled cladogram of fossil taxa

`timeData` Two-column matrix of first and last occurrances in absolute continous time, with rownames as the taxon IDs used on the tree

`type` Type of time-scaling method used. Can be "basic", "equal", "aba", "zbla" or "mbl". See below for details. Type "basic" by default.

`vartime` Time variable; usage depends on the method 'type' argument. Ignored if type = "basic"

`ntrees` Number of time-scaled trees to output

`randres` Should polytomies be randomly resolved?

`add.term` If true, adds terminal ranges (see below)

`rand.obs` Should the tips represent observation times uniform distributed within taxon ranges?

`node.mins` Minimum ages of nodes on the tree, see below

`plot` If true, plots the input and output phylogenies

`timeList` A list composed of two matrices giving interval times and taxon appearance datums, as would be output by binTimeData. The rownames of the second matrix should be the taxon IDs

`sites` A two column matrix, composed of site IDs for taxon FADs and LADs. Does not have to be given by default; see explanation below.

Details

This function is an attempt to unify and collect previously used and discussed methods for time-scaling phylogenies of fossil taxa. Unfortunately, it is difficult to attribute some method types to specific references.

There are five method types that can be used by timePaleoPhy. Four of these use some amount of absolute time, chosen a priori, to time-scale the tree. This is handled by the argument vartime, which is NULL by default.

"basic" : This most simple of methods ignores vartime and scales nodes so they are as old as the first appearance of their oldest descendant. This method produces many zero-length branches.

"equal" : The 'equal' method defined by G. Lloyd and used in Brusatte et al. (2008) and Lloyd et al. (2012). Originally usable in code supplied by G. Lloyd, it is recreated here. This method works by increasing the time of the root divergence by some amount (here set by vartime) and then adjusting zero-length branches so that time on early branches is re-apportioned out along those later branches equally.

"aba" : All branches additive. This method takes the "basic" tree and adds vartime to all branches.

"zlba" : Zero-length branches additive. This method adds vartime to all zero-length branches in the "basic" tree. Mentioned by Hunt and Carrano, 2010.

"mbl": Minimum branch length. Scales all branches so they are greater than or equal to vartime, and subtract time added to later branches from earlier branches in order to maintain the temporal structure of events.

This function cannot time-scale branches relative to reconstructed character changes along branches, as used by Brusatte et al. (2008).

By default, timePaleoPhy does not resolve polytomies, instead outputting a time-scaled tree that is only as resolved as the input tree. If randres=T, then polytomies will be randomly resolved using multi2di() from the package ape. If randres=T and ntrees=1, then a warning is printed that users should analyze multiple randomly-resolved trees, rather than a single such tree. If ntrees is greater than one and randres=F, the function will fail and a warning is issued, as these are by definition arguments in conflict with each other.

By default, this function will not add the ranges of taxa when time-scaling a tree, so that the tips correspond temporally to the first appearance datums of the given taxa. If add.term=T, then the 'terminal ranges' of the taxa are added to the tips after tree is time-scaled, such that the tips now correspond to the last appearance datums.

If add.term and rand.obs are both set to be true, then it is assumed that users wish the tips to represent observations made with some temporal uncertainty, such that they might have come from any point within a taxon's range. This might be the case, for example, if a user is interested in applying phylogeny-based approaches to studying trait evolution, but have per-taxon measurements of traits that come from museum specimens with uncertain temporal placement. When these arguments are set to be true, the tips are placed randomly within taxon ranges, as if uniformly distributed. Obviously, as with randres, multiple trees should be created and then analyzed. If add.term=F and rand.obs=T, the function fails and a warning is issued.

The minimum dates of nodes can be set using node.mins; this argument takes a vector of the same length as the number of nodes, with dates given in the same order as nodes are number in the tree$edge matrix. Not all nodes need be set; those without minimum dates can be given as NA in node.mins.

All trees are output with an element $root.time. This is the time of the root on the tree and is important for comparing patterns across trees.

These functions will intuitively drop taxa from the tree with NA for range or are missing from timeData or timeList.

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero.

bin_timePaleoPhy is a wrapper of timePaleoPhy which produces timescaled trees for datasets which only have interval data available. For each output tree, taxon FADs and LADs are placed within their listed intervals under a uniform distribution. Thus, a large sample of time-scaled trees will approximate the uncertainty in the actual timing of the FADs and LADs.

The sites argument allows users to constrain the placement of dates in bin_timePaleoPhy by restricting multiple fossil taxa whose FADs or LADs are from the same very temporally restricted sites (such as fossil-rich Lagerstatten) to always have the same date, across many iterations of time-scaled trees from bin_timePaleoPhy. To do this, simply give a matrix where the "site" of each FAD and LAD for every taxon is listed, as corresponding to the second matrix in timeList. If no sites matrix is given (the default), then it is assumed all fossil come from different "sites" and there is no shared temporal structure among the events.

Value

The output of these functions is a time-scaled tree or set of time-scaled trees, of either class phylo or multiphylo, depending on the argument ntrees.

Author(s)

David W. Bapst, heavily inspired by code supplied by Graeme Lloyd and Gene Hunt.

References

Brusatte, S. L., M. J. Benton, M. Ruta, and G. T. Lloyd. 2008. Superiority, Competition, and Opportunism in the Evolutionary Radiation of Dinosaurs. Science 321(5895):1485-1488.

Hunt, G., and M. T. Carrano. 2010. Models and methods for analyzing phenotypic evolution in lineages and clades. In J. Alroy, and G. Hunt, eds. Short Course on Quantitative Methods in Paleobiology. Paleontological Society.

Lloyd, G. T., S. C. Wang, and S. L. Brusatte. 2012. Identifying Heterogeneity in Rates of Morphological Evolutio: Discrete Character Change in the Evolution of Lungfish(Sarcopterygii, Dipnoi). Evolution 66(2):330-348.

See Also

srcTimePaleoPhy,binTimeData,multi2di

Examples

```
#timePaleoPhy
##Simulate some fossil ranges with simFossilTaxa()
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont<-sampleRanges(taxa,r=0.5)
#let's use taxa2cladogram() to get the 'ideal' cladogram of the taxa
cladogram<-taxa2cladogram(taxa,plot=TRUE)
#Now let's try timePaleoPhy() using the continuous range data
ttree<-timePaleoPhy(cladogram,rangesCont,type="basic",plot=TRUE)
#plot diversity curve
phyloDiv(ttree)

#that tree lacked the terminal parts of ranges (tips stops at the taxon FADs)
#let's add those terminal ranges back on with add.term
ttree<-timePaleoPhy(cladogram,rangesCont,type="basic",add.term=TRUE,plot=TRUE)
#plot diversity curve
phyloDiv(ttree)

#that tree didn't look very resolved, does it? (The curse of sampled ancestry!)
#if we set ntrees>1, timePaleoPhy() will make multiple time-trees
#each resulting tree will have polytomies randomly resolved in different ways using multi2di()
ttree<-timePaleoPhy(cladogram,rangesCont,type="basic",ntrees=1,randres=TRUE,add.term=TRUE,plot=TRUE)
#notice that the warning it prints!
#now let's plot the first tree (both trees will be identical because we used set.seed)
```

```
phyloDiv(ttree)
#we would need to set ntrees to a large number to get a fair sample of trees

#compare different methods of timePaleoPhy
layout(matrix(1:6,3,2));par(mar=c(3,2,1,2))
plot(ladderize(timePaleoPhy(cladogram,rangesCont,type="basic",vartime=NULL,add.term=TRUE)))
axisPhylo();text(x=50,y=23,"type=basic",adj=c(0,0.5),cex=1.2)
plot(ladderize(timePaleoPhy(cladogram,rangesCont,type="equal",vartime=10,add.term=TRUE)));axisPhylo()
axisPhylo();text(x=55,y=23,"type=equal",adj=c(0,0.5),cex=1.2)
plot(ladderize(timePaleoPhy(cladogram,rangesCont,type="aba",vartime=1,add.term=TRUE)));axisPhylo()
axisPhylo();text(x=55,y=23,"type=aba",adj=c(0,0.5),cex=1.2)
plot(ladderize(timePaleoPhy(cladogram,rangesCont,type="zlba",vartime=1,add.term=TRUE)));axisPhylo()
axisPhylo();text(x=55,y=23,"type=zlba",adj=c(0,0.5),cex=1.2)
plot(ladderize(timePaleoPhy(cladogram,rangesCont,type="mbl",vartime=1,add.term=TRUE)));axisPhylo()
axisPhylo();text(x=55,y=23,"type=mbl",adj=c(0,0.5),cex=1.2)

#Using bin_timePaleoPhy to timescale with discrete interval data
#first let's use binTimeData() to bin in intervals of 1 time unit
rangesDisc<-binTimeData(rangesCont,int.length=1)
ttree<-bin_timePaleoPhy(cladogram,rangesDisc,type="basic",ntrees=1,randres=TRUE,add.term=TRUE,plot=TRUE)
#notice that the warning it prints!
phyloDiv(ttree)
```

timeSliceTree *Taking a Timeslice of a Tree*

Description

Removes the portion of a tree after a set point in time, as if the tree had been sliced at that moment.

Usage

```
timeSliceTree(ttree, sliceTime, drop.extinct = F, plot = T)
```

Arguments

`ttree`	A time-scaled phylogeny of class phylo
`sliceTime`	Time to 'slice' the tree at. See details below.
`drop.extinct`	If true, drops tips that go extinct before timeSlice.
`plot`	If true, plots input and output trees for comparison.

Details

The sliceTime is always calculated as on the same as scale as ttree$root.time, which the time before present that the root divergence occurred. In other words, if root.time=100, then timeSlice=80 will slice the tree 20 time units after the root.

The function assumes that ttree will generally have an element called $root.time. If $root.time is not present as an element of ttree, then it is assumed the tip furthest from the root is at time 0 (present-day) and a new $root.time is calculated (a warning will be issued in this case).

If drop.extinct=T, then extinct tips are dropped and (if present) the $root.time of ttree is adjusted.

Value

Returns the modified phylogeny as an object of class phylo

Author(s)

David W. Bapst

See Also

phyloDiv,

Examples

```
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,nExtant=0)
taxicDivCont(taxa[,3:4])
#that's the whole diversity curve
#with timeSliceTree we could look at the lineage accumulation curve we'd get of species sampled at a point
tree<-taxa2phylo(taxa)
#use timeSliceTree to make tree of relationships up until time=950
tree950<-timeSliceTree(tree,sliceTime=950,plot=TRUE,drop.extinct=FALSE)
#use drop.extinct=T to only get the tree of lineages extant at time=950
tree950<-timeSliceTree(tree,sliceTime=950,plot=TRUE,drop.extinct=TRUE)
#now its an ultrametric tree with many fewer tips...
#lets plot the lineage accumulation plot on a log scale
phyloDiv(tree950,plotLogRich=TRUE)
```

unitLengthTree *Scale Tree to Unit-Length*

Description

Rescales all edges of a phylogeny to be equal to 1 ("unit-length").

Usage

```
unitLengthTree(tree)
```

Arguments

tree	an object of class phylo

Details

No additional details.

Value

Returns the modified phylogeny as an object of class phylo

Author(s)

David W. Bapst

See Also

speciationalTree,timePaleoPhy,

Examples

```
set.seed(444)
tree<-rtree(10)
layout(matrix(1:2,,2))
plot(tree);plot(unitLengthTree(tree))
```

Index