Package ‘paleotree’

February 12, 2012

Type Package

Version 1.1

Title Paleontological and Phylogenetic Analyses of Evolution

Date 2012-02-12

Author David Bapst

Depends ape

Imports 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 these two data types.

License GPL version 2 or newer

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

R topics documented:

`paleotree-package` *paleotree: Paleontological and Phylogenetic Analyses of Evolution*

Description

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

Details

Package:	paleotree
Type:	Package
Version:	1.1
Date:	2012-02-01
License:	GPL version 2 or newer

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

Bapst, in prep. paleotree: an R package for paleontological and phylogenetic analyses of evolution. To be submitted to Methods in Ecology and Evolution.

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

See Also

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

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,maxExtant=0)
#let's see what the 'true' diversity curve looks like in this case
```

```
#plot the FADs and LADs with taxicDivCont()
taxicDivCont(taxa)

#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, default is 1 time unit
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 sequential non-overlappling time intervals, with 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,maxExtant=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)
```

`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`	A number between 0 to 1, which conditions the depth of nodes removed. If NA, no conditioning (this is the default).

Details

The nodes are removed at random using sample(). This function can be conditioned to remove nodes of a particular depth with greater probability/frequency by setting node.depth to some value between zero (deep nodes close to the root) or one (shallow nodes far from the root). Depth is evaluated based on the number of descendant tips. If node.depth is not NA, the relative proportion of descendants from each node is calculated, summed to 1 and the node.depth value subtracted from this proportion. These values are then squared, normalized again to equal to 1 and then used as the probabilities for sampling nodes for removal.

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)
#let's compare the input and output
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 Branch 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

The only purpose of this function is to make an aesthetically-pleasing graphic of one's tree, where branches are color-coded, using a rainbow palette, relative to their depth. 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(500)
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, plotLogRich = F)
taxicDivDisc(timeList, int.times = NULL, plot = T, plotLogRich = F)
phyloDiv(tree, int.length = 1, int.times = NULL, plot = T, plotLogRich = F, drop.ZLB=T)
```

Arguments

`timeData` Two-column matrix giving the per-taxon first and last appearances in absolute time. The matrix resulting from a single simFossilTaxon run can also be supplied to taxicDivCont, in which case all columns except the third and the fourth are ignored.

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

`tree` 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.

`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 plots of species/taxon/lineage richness over time for a particular group of organisms. For paleontological studies, these are generally based on per-taxon range data while more recently in evolutionary biology, molecular phylogenies have been used to calculate 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 (and plot them as rectangular bins). For continuous-time data or phylogenies, one could decrease the int.length used to get what is essentially 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 probably is not finely-resolved enough 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

These functions will invisibly return 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,timeSliceTree,

There are several different functions for traditional LTT plots (phylogenetic diversity curves), such as the function ,ltt.plot in the package ape, the function ltt() in the package phytools, the function plotLtt() in the package laser and the function LTT.average.root() in the package TreeSim.

Examples

```
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,maxExtant=0)
#let's see what the 'true' diversity curve looks like in this case
#plot the FADs and LADs with taxicDivCont()
taxicDivCont(taxa)
#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
  #using a tree from rtree in ape
set.seed(444)
tree<-rtree(100)
phyloDiv(tree)

#a neat example of using phyDiv with timeSliceTree
 #to simulate doing molecular-phylogeny studies
 #of diverification...in the past
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,maxExtant=0)
taxicDivCont(taxa)
#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 in time
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)
```

---

droppingBranches *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)
dropExtant(tree, tol = 0.01)
dropExtinct(tree, tol = 0.01, ignore.root.time = F)
```

Arguments

`tree` A phylogeny as a phylo object

`tol` Tolerance of determining modern age for distinguishing extinct from extant taxa

`ignore.root.time` Ignore root.time in calculating which tips are extinct? root.time will still be adjusted

Details

DropZLB 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`

DropExtinct drops all terminal branches which end before the modern (i.e. extinct taxa). DropExtant drops all terminal branches which end at the modern (i.e. extant/still-living taxa). In both cases, the modern is defined based on tree$root.time if available, or the modern is inferred to when the tip furthest from the root terminates in tree.

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 terminal branches are dropped. Adjusted root.times are only given if the input tree has root.times.

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,maxExtant=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])
#compare the two trees
layout(matrix(1:2,,2))
plot(ladderize(tree))
plot(ladderize(dropZLB(tree)))


#example using dropExtinct and dropExtant
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=40,maxtime=1000,maxExtant=20)
tree<-taxa2phylo(taxa)
```

```
phyloDiv(tree)
tree1<-dropExtinct(tree)
phyloDiv(tree1)
tree2<-dropExtant(tree)
phyloDiv(tree2)
```

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 and Donoghue (2005) and Friedman (2009). Obviously, any analyses one wishes to do should be done by resolving this tree with multi2di in the ape package or the various time-scaling functions found in this package (paleotree).

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. By default, all higher taxa are treated as monophyletic if not otherwise specified.

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)
```

Arguments

`timeData` A 2 column matrix with the first and last occurances of taxa given in relative time intervals. If a list of length two is given for timeData, such as would be expected if the output of binTimeData was directly input, the second element is used.

`n_tbins` Number of time bins with different sampling/extinction parameters

`grp1` A vector of integers or characters, the same length as the number of taxa in timeData, where each taxon-wise element gives the group ID of the taxon for the respective row of timeData

`grp2` A vector of integers or characters, the same length as the number of taxa in timeData, where each taxon-wise element gives the group ID of the taxon for the respective row of timeData

`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, although the behavior of those datasets is less well understood.

getSampProbDisc 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 per-taxon vectors (of either character or integers) 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 allows 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. For now, data input into this function should be for taxa that have already gone extinct by the modern 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, a dataset where all taxa are listed from a set of sequential intervals of similar length, such as North American Mammal assemblage zones, microfossil faunal zones or graptolite biozones can be given as long as they are correctly numbered in sequential order in the input. As a counter example, a dataset which includes taxa resolved only to intervals as wide as the whole Jurassic and taxa resolved to biozones within the Jurassic should not be included in the same input. Drop the taxa from less poorly resolved intervals from such datasets if you want to apply this function, as long as this retains a large enough sample of taxa from the sequential intervals. Note that taxicDivDisc and the "bin_" timescaling methods do not require that intervals be truly sequential (they can be overlapping; see their helpfiles). The output from binTimeData is always sequential, at least by default.

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, per class of taxa fit

`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 (Pp) will be output with these parameters as long as classes are not overlapping, as those estimates would not otherwise refer to meaningful groups of taxa.

Author(s)

David W. Bapst, with considerable advice from Michael Foote.

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`,`qsProb2Comp`,

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,maxExtant=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)
#for discrete data we can estimate the sampling probability per interval (R)
#i.e. this is not the same thing as the instantaneous sampling rate (r)
#can use sRate2sProb to see what we would expect
sRate2sProb(r=0.5)
#expect R = ~0.39
#now we can use maximum likelihood to taxon ranges to get sampling probability
SPres1<-getSampProbDisc(rangesDisc)
sProb<-SPres1[[2]][2]
print(sProb)
#est. R = ~0.42; not too off what we would expect!
#for the src based timescaling methods, we want an estimate of the instantaneous samp rate
#we can use sProb2sRate() to get the rate. We will also need to also tell it the int.length
sRate<-sProb2sRate(sProb,int.length=1)
print(sRate)
#estimates that r=0.54... Not bad!
```

```
#Note: for real data, you may need to use an average int.length (no constant length)

## Not run:
#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 informative models)
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)

#testing temporal change in sampling rate
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=100,maxtaxa=125,maxtime=1000,maxExtant=0,plot=T)
#let's see what the 'true' diversity curve looks like in this case
#simulate two sets of ranges at r=0.7 and r=0.1
rangesCont<-sampleRanges(taxa,r=1.1)
rangesCont2<-sampleRanges(taxa,r=0.2)
#now make it so that taxa which originated after 850 have r=0.1
rangesCont[taxa[,3]<850,]<-rangesCont2[taxa[,3]<850,]
rangesDisc<-binTimeData(rangesCont)
#lets plot the diversity curve
taxicDivDisc(rangesDisc)
SPres1<-getSampProbDisc(rangesDisc)
SPres2<-getSampProbDisc(rangesDisc,n_tbins=2)
#compare the AICc of the models
print(c(SPres1$AICc,SPres2$AICc)) #model 2 looks pretty good
#when does it find the break in time intervals?
print(rangesDisc[[1]][SPres2$t_ends[2],1])
#not so great: estimates 940, not 850
#but look at the diversity curve: most richness in bin 1 is before 940
#might have found the right break time otherwise...
#the parameter values it found are less great. Finds variation in q

## 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 of integers or characters, the same length as the number of taxa in timeData, where each taxon-wise element gives the group ID of the taxon for the respective row of timeData

`grp2` A vector of integers or characters, the same length as the number of taxa in timeData, where each taxon-wise element gives the group ID of the taxon for the respective row of timeData

`threshold` The smallest allowable range (measured difference in the FAD and LAD of a taxon). Ranges below this size will be treated as "one-hit" sampling events.

`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, although the behavior of those datasets is less well understood.

getSampRateCont 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 per-taxon vectors (of either character or integers) 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 allows 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. For now, data input into this function should be for taxa that have already gone extinct by the modern 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 (Pp) will be output with these parameters as long as classes are not overlapping, as those estimates would not otherwise refer to meaningful groups of taxa.

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,qsRate2Comp,

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,maxExtant=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 informative)
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, drop.ZLB=T, plotLogRich = F)
plotMultiDiv(results, plotLogRich = F)
```

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
`drop.ZLB`	Should terminal zero-length branches be dropped on phylogenetic datasets?
`results`	The output of a previous run of multiDiv
`plotLogRich`	If true, taxic diversity plotted on log scale

Details

This function is essentially a wrapper for the individual diversity curve functions included in this package (paleotree). 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. A list of matrices output from simFossilTaxa is allowable, and considered as input for taxicDivCont. Data of an unknown type gives back an error.

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 will be invisibly returned:

`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,maxExtant=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)
input<-list(rangesCont,rangesDisc,ttree)
multiDiv(input,plot=TRUE)

#using multiDiv with samples of trees
ttrees<-timePaleoPhy(cladogram,rangesCont,type="basic",randres=TRUE,ntrees=10,add.term=TRUE)
multiDiv(ttrees)
#uncertainty in diversity history is solely due to
 #the random resolution of polytomies

#multiDiv can also take output from simFossilTaxa
#what do many simulations run under some conditions 'look' like on average?
set.seed(444)
taxa<-simFossilTaxa(p=0.3,q=0.1,nruns=20,maxtime=20,maxtaxa=100,plot=TRUE,min.cond=FALSE)
multiDiv(taxa)
#increasing cone of diversity! Even better on a log scale:
multiDiv(taxa,plotLogRich=TRUE)
```

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 these estimates 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`

Also see the function `traitgram` in the library picante.

Examples

```
require(geiger)
set.seed(444)
tree<-rtree(10)
trait<-rTraitCont(tree)
#first, traitgram without conf intervals
plotTraitgram(trait,tree,conf.int=FALSE)

#now, with
plotTraitgram(trait,tree)
#not much confidence, eh?
```

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 two-column matrix of per-taxon ranges. The five-column matrix output of simFossilTaxa can also be supplied.
`r`	Instantaneous sampling rate per time unit
`min.taxa`	Minimum number of taxa sampled. The default is 2.
`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 a simple sampling models, where sampling events occur under a rate which is homogenous through time and across lineages and the waiting times between these sampling events 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.

Conditioning on sampling some minimum number of taxa may create strange simulation results for some analyses, such as simulation analyses of birth-death processes. Set min.taxa=0 to remove this conditioning.

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,maxExtant=0)
layout(matrix(1:2,2,))
#let's see what the 'true' diversity curve looks like in this case
taxicDivCont(taxa)
#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)
#compare the true history to what we would observe!
```

SamplingConv *Converting Sampling Estimates*

Description

Various functions for converting between estimates of sampling in the fossil record

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 a taxon at least once
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 only 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 found in the appendices of those papers.

Value

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

Author(s)

David W. Bapst, with advice 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)
```

---

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, mintime = 1,maxtime = 1000, minExtant = 0,
maxExtant = NULL, min.cond = T, print.runs = F, plot = F)

simFossilTaxa_SRCond(r, avgtaxa, p, q, w = 0, u = 0, nruns = 1,
maxtime = 100, maxExtant = NULL, plot = F)
```

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
mintime	Minimum time units to run any given simulation before stopping it
maxtime	Maximum time units to run any given simulation before stopping it
minExtant	Minimum number of living taxa allowed at end of simulations
maxExtant	Maximum number of living taxa allowed at end of simulations
min.cond	Stop simulations when they hit minimum conditions or go until they hit maximum conditions?
print.runs	Print the proportion of simulations accepted for output?
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

simFossilTaxa simulates a birth-death process (Kendall, 1948; Nee, 2006), but unlike most functions for this implemented in R, this function simulates the diversification of clades where taxa are relatively morphologically static over long time intervals. The output is a description of the temporal and phylogenetic relationships of those morphotaxa. This is meant to emulate the sort of data that paleobiologists often work with, especially in well-sampled groups.

If min.cond=T (the default), simulations will stop when clades satisfy mintime, mintaxa, minExtant and maxExtant (if the later is set). To reduce the effect of one condition, simply set the limit to an arbitrarily low number. If min.cond=F, then the simulations are not stopped until they (a) go extinct or (b) hit either maxtaxa or maxtime. Whether they are accepted or not for output is still dependent on mintaxa, mintime, minExtant and maxExtant. Note that some combinations of conditions, such as setting minExtant=maxExtant>0

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 mintaxa or minExtant is hit, until the next taxon (limit +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 evolutionary history of the simulated clade, not the extant richness at the end of the simulation. Use minExtant and maxExtant if you want to condition on the 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 maxExtant 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 (a value of 1 indicates a taxon is still alive, a value of 0 indicates the taxon is extinct).

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. Kendall, D. G. 1948. On the Generalized "Birth-and-Death" Process. The Annals of Mathematical Statistics 19(1):1-15. Nee, S. 2006. Birth-Death Models in Macroevolution. Annual Review of Ecology, Evolution, and Systematics 37(1):1-17.

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,maxExtant=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])
#can also see this by setting plot=TRUE in simFossilTaxa

#make datasets with multiple speciation modes
#following has anagenesis, budding cladogenesis and bifurcating cladogenesis
#all set to 1/2 extinction rate
set.seed(444)
res<-simFossilTaxa(p=0.1,q=0.1,w=0.05,u=0.5,mintaxa=30,maxtaxa=60,maxExtant=0,nruns=1,plot=TRUE)
#what does this mix of speciation modes look like as a phylogeny?
tree<-taxa2phylo(res,plot=TRUE)

#can generate datasets that meet multiple conditions: time, # total taxa, # extant taxa
set.seed(444)
res<-simFossilTaxa(p=0.1,q=0.1,mintime=10,mintaxa=30,maxtaxa=40,minExtant=10,maxExtant=20,nruns=20,plot=FAL
#use print.run to know how many simulations were accepted of the total generated
layout(matrix(1:2,2,))
#histogram of # taxa over evolutionary history
hist(sapply(res,nrow),main="#taxa")
#histogram of # extant taxa at end of simulation
hist(sapply(res,function(x) sum(x[,5])),main="#extant")
```

```
#can generate datasets where simulations go until extinction or max limits
#and THEN are evaluated whether they meet min limits
#good for producing unconditioned birth-death trees
set.seed(444)
res<-simFossilTaxa(p=0.1,q=0.1,maxtaxa=100,maxtime=100,nruns=10,plot=TRUE,print.runs=TRUE,min.cond=FALSE)
#hey, look, we accepted everything! (That's what we want.)
layout(matrix(1:2,2,))
#histogram of # taxa over evolutionary history
hist(sapply(res,nrow),main="#taxa")
#histogram of # extant taxa at end of simulation
hist(sapply(res,function(x) sum(x[,5])),main="#extant")

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

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=1, all.extinct=F, modern.samp=F, mintime=1,
maxtime=100, mintaxa=2, maxtaxa=500, drop.zlb = T, plot = F)
```

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
`all.extinct`	Condition on all taxa being extinct by modern? Default is false
`modern.samp`	Should taxa extant at present be sampled perfectly at the present-day (time=0)?
`mintime`	Minimum time to run simulation
`maxtime`	Maximum time to run simulation
`mintaxa`	Minimum number of taxa allowed over entire simulation
`maxtaxa`	Maximum number of taxa allowed over entire simulation
`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).

simPaleoTrees essentially uses simFossilTaxa with no.cond=T and other minimal conditioning, to get as unbiased a sample of simulations as possible (without exceeding the maximum constraints). This is useful for birth-death analyses, although a number of options available in simFossilTaxa are thus unavailable in simPaleoTrees. By default, there is no conditioning on the number of extant taxa, living taxa are sampled perfectly at time 0 and zero-length branches are dropped. Unlike simPaleoTrees, you cannot condition on a certain number of extant taxa, only whether they are allowed or not (via all.extinct). There are also no options relating to speciation modes: taxa are only simulated under budding cladogenesis.

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. Additionally, each of these simulated phylogenies will have the original simulated taxa data (from simFossilTaxa) and sampled ranges (from sampleRanges) attached as the elements $taxa and $ranges.

Author(s)

David W. Bapst

See Also

simFossilTaxa,taxa2phylo,sampleRanges,

Examples

```
set.seed(444)
#simulate trees conditioned to have no living descendants
trees<-simPaleoTrees(p=0.1,q=0.1,r=0.5,ntrees=10,all.extinct=TRUE,maxtime=100,plot=TRUE)
#number of tips
sapply(trees,Ntip)

#simulate trees conditioned to (possibly) have living descendants and perfect sampling at modern
trees<-simPaleoTrees(p=0.1,q=0.1,r=0.5,ntrees=10,all.extinct=FALSE,maxtime=100,modern.samp=TRUE,plot=TRUE)
#number of tips
sapply(trees,Ntip)
```

---

srcTimescaling *SampRate-Calibrated Timescaling of Paleo-Phylogenies*

Description

This function takes an input an unscaled cladogram of fossil taxa, information on their ranges and an estimate of the instantaneous rate of sampling. The output is a sample of timescaled trees, as resulting from a stochastic algorithm that samples observed gaps in the fossil record with weights calculated from the sampling rate. This function also uses the sampling-rate calibrated time-scaling algorithim to resolve polytomies randomly and infer potential ancestor-descendant relationships, simultaneous with the time-scaling.

Usage

```
srcTimePaleoPhy(tree, timeData, sampRate, ntrees = 1, anc.wt = 1, rand.obs = F, node.mins = NULL
root.max = 200, plot = F)

bin_srcTimePaleoPhy(tree, timeList, sampRate, ntrees = 1, sites = NULL, anc.wt = 1, node.mins =
 rand.obs = F, root.max = 200, plot = F)
```

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. 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.

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

`node.mins` Minimum ages of nodes on the tree. 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.

`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` Optional two column matrix, composed of site IDs for taxon 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.

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 algorithm is also extended to apply to resolving polytomies and designating possible ancestor-descendant relationships. The full details of this method and the algorithm use will be given in Bapst (in prep). Its performance with other time-scaling methods will also be compared via simulation.

srcTimePaleoPhy is only applicable to datasets with taxon occurances in continuous time. 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 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. Most datasets will probably use getSampProbDisc and sProb2sRate prior to using this function, as shown in an example below.

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). This can be turned off by setting anc.wt=0. As this function may infer anagenetic relationships during time-scaling, this can create zero-length terminal branches in the output. 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.

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 that are missing from timeData.

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. 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.

Note

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.

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,maxExtant=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)
#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)

#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)

#to get a fair sample of trees, let's increse ntrees
#can do an example of such an analysis via multDiv
ttrees<-srcTimePaleoPhy(cladogram,rangesCont,sampRate=sRate,ntrees=10,plot=FALSE)
multiDiv(ttrees)

#example with time in discrete intervals
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,maxExtant=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 use binTimeData() to bin in intervals of 1 time unit
rangesDisc<-binTimeData(rangesCont,int.length=1)
#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 simulates an ideal cladistic process, where the relationships of a set of morphologically static taxa is resolved into a set of nested hierarchial relationships (a standard cladogram), as much as would be expected given the input relationships among those taxa. taxa2cladogram uses information on the ancestor-descendant relationships of a bunch of taxa and constructs an unscaled cladogram of the hierarcially-nesting relationships among those taxa. There's no actual cladistics going on, this is just a simulation of that process. If there is any chance that a set of taxa could be resolved into a set of nested relationships given their ancestor-descendant relationships, they will be resolved so in the output of taxa2cladogram. No morphological characters are considered, we just assume that if there is a nesting relationship, then it could be resolved as such. This makes it the "ideal" cladogram of a simulated clade.

The result will probably not be fully resolved, as including both ancestor and descendant taxa will generally make it 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 both the second and third. If we imagine an ideal cladistic analysis of the morphological characters of those three taxa, this set of taxa will be unable to be broken up into bifurcating-nested relationships and thus result in a polytomy. 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.

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,maxExtant=0)
#let's use taxa2cladogram() to get the 'ideal' cladogram of the taxa
layout(matrix(1:2,,2))
cladogram<-taxa2cladogram(taxa,plot=TRUE)
#compare the real relationships (taxa2phylo) to the ideal cladogram
tree<-taxa2phylo(taxa,plot=TRUE)
```

`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

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

All data relating to when static morpho-taxa appear or disappear in the record is lost; branching points will be the actual time of speciation, which (under budding) will often be in the middle of the temporal range of a taxon.

Value

The resulting phylogeny with branch lengths is output as an object of class phylo. This function will output trees with the element $root.time, which is the time of the root divergence in absolute time.

Note

Please, DO NOT use this function to time-scale a real tree for a real dataset. It assumes you know the divergence/speciation times of the branching nodes perfectly, which is almost impossible given the nature of the fossil record. Use timePaleoPhy or srcTimePaleoPhy instead.

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.

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,maxExtant=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,drop.ZLB=FALSE)
```

`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 = F, rand.obs = F, node.mins = NULL, plot = F)

bin_timePaleoPhy(tree, timeList, type = "basic", vartime = NULL, ntrees = 1,
randres = F, sites = NULL,  add.term = F, rand.obs = F, node.mins = NULL, plot = F)
```

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". Type="basic" by default. See the note below for more details

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

`ntrees` Number of time-scaled trees to output. If ntrees is greater than one and both randres and rand.obs are false, the function will fail and a warning is issued, as these arguments would simply produce multiple identical time-scaled trees.

`randres` Should polytomies be randomly resolved? 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, a warning is printed that users should analyze multiple randomly-resolved trees, rather than a single such tree, although a tree is still output.

`add.term` If true, adds terminal ranges.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.

`rand.obs` Should the tips represent observation times uniform distributed within taxon ranges? If rand.obs=T but add.term=F, the function fails and a warning is issued. If rand.obs=TRUE, tips are placed randomly within taxon ranges, as if uniformly distributed. This serves those users that wish for 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. As with randres, multiple trees should be created and then analyzed.

`node.mins` Minimum ages of nodes on the tree. 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.

`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` Optional two column matrix, composed of site IDs for taxon 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.

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 time-scaling methods to specific references in the literature. A review of these time-scaling methods is forthcoming (Bapst, in prep).

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

"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. Discussed 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 Lloyd et al. (2012).

timePaleoPhy can only be applied to datasets where taxon appearances are in continuous time. 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.

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.

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. 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.

Author(s)

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

References

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

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

```
##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,maxExtant=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 well 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=FALSE)
#notice 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 after that moment had been sliced away.

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 function assumes that ttree will generally have an element called $root.time, which is the time before present that the root divergence occurred. 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).

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

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

Value

Returns the modified phylogeny as an object of class phylo. Tip labels for cut branches which held multiple descendant tips will be the label for the earliest appearing tip descendant of that branch.

Author(s)

David W. Bapst

See Also

`phyloDiv`,`dropExtinct`,`dropExtant`

Also see the function treeSlice in the library phytools, which will slice a tree at some point in and return all the subtrees which remain after the slicing time. (Effectively the reverse of timeSliceTree.)

Examples

```
#a neat example of using phyloDiv with timeSliceTree
 #to simulate doing molecular-phylogeny studies
 #of diverification...in the past
set.seed(444)
taxa<-simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,maxExtant=0)
taxicDivCont(taxa)
#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 in time
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. Any $root.time element is removed.

Author(s)

David W. Bapst

See Also

timePaleoPhy,

See also speciationalTree in the package geiger, which does essentially the same thing.

Examples

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

Index