

Package ‘paleotree’

April 19, 2012

Type Package

Version 1.3

Title Paleontological and Phylogenetic Analyses of Evolution

Date 2012-04-17

Author David Bapst

Depends R (>= 2.15.0), ape

Imports phangorn, geiger, Matrix

Suggests paleoTS

Maintainer David Bapst <dwbapst@gmail.com>

Description Analyzes, time-scales and simulates phylogenies of extinct/fossil lineages. Also calculates diversity curves for stratigraphic range data and phylogenies.

License GPL (>= 2)

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

Repository CRAN

Date/Publication 2012-04-19 05:39:57

R topics documented:

paleotree-package	2
binTimeData	3
cladogeneticTraitCont	5
compareTimescaling	7
degradeTree	8
depthRainbow	9
DiversityCurves	10
expandTaxonTree	13

getSampProbDisc	15
getSampRateCont	18
modifyTerminalBranches	21
multiDiv	23
plotTraitgram	25
sampleRanges	27
SamplingConv	30
simFossilTaxa	32
simPaleoTrees	37
srcTimescaling	39
taxa2cladogram	43
taxa2phylo	45
timePaleoPhy	47
timeSliceTree	51
unitLengthTree	52

Index	54
--------------	-----------

paleotree-package	<i>paleotree: Paleontological and Phylogenetic Analyses of Evolution</i>
-------------------	--

Description

Analyzes, time-scales and simulates phylogenies of extinct/fossil lineages. Also calculates diversity curves for stratigraphic range data and phylogenies.

Details

Package: paleotree
 Type: Package
 Version: 1.3
 Date: 2012-04-17
 License: GPL (>= 2)

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 taxic and phylogenetic varieties.

Author(s)

David W. Bapst

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

References

Bapst, in revision. paleotree: an R package for paleontological and phylogenetic analyses of evolution. *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 and standards offered by the [ape](#) package.

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(1:2)
taxicDivCont(rangesCont)
#Now let's use binTimeData to bin in intervals of 10 time units
rangesDisc <- binTimeData(rangesCont,int.length=10)
#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)
```

Description

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

Usage

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

Arguments

<code>timeData</code>	Two-column matrix of first and last occurrences in absolute continuous time
<code>int.length</code>	Time interval length, default is 1 time unit
<code>start</code>	Starting time for calculating the intervals.
<code>int.times</code>	A two column matrix with the start and end times of the intervals to be used.

Details

This function takes a matrix of per-taxon first and last occurrences and divides those occurrences into sequential non-overlapping time intervals, with each interval of uniform duration equal to `int.length`.

Alternatively, a two column matrix of the interval start and end times to be used. None of these intervals can have a duration (temporal length) equal to or less than zero. If a first or last appearance in the input range data could fit into multiple intervals (i.e. the input discrete time intervals are overlapping), then the appearance data is placed in the interval of the shortest duration. When output, the interval times matrix (see below) will be sorted from first to last.

As with many functions in the paleotree library, absolute time is always decreasing, i.e. the present day is zero. However, the numbering of intervals given in the output increases with time, as these are numbered relative to each other, from first to last.

Value

A list containing:

<code>int.times</code>	A 2 column matrix with the start and end times of the intervals used; time decreases relative to the present.
<code>taxon.times</code>	A 2 column matrix with the first and last occurrences of taxa in the intervals listed in <code>int.times</code> , with numbers referring to the row of <code>int.times</code> .

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()
equalDiscInt <- taxicDivDisc(rangesDisc)

#example with pre-set intervals input (including overlapping)
presetIntervals <- cbind(c(1000,990,970,940),c(980,970,950,930))
rangesDisc1 <- binTimeData(rangesCont,int.times=presetIntervals)
taxicDivDisc(rangesDisc1)
#now let's plot diversity with (different) equal length intervals used above
taxicDivDisc(rangesDisc1,int.times=equalDiscInt[,1:2])

```

cladogeneticTraitCont *Simulate Cladogenetic Trait Evolution*

Description

This function simulates trait evolution at each speciation/branching event in a matrix output from simFossilTaxa.

Usage

```
cladogeneticTraitCont(taxa, rate = 1, meanChange = 0, rootTrait = 0)
```

Arguments

taxa	A five-column matrix of taxonomic data, as output by simFossilTaxa
rate	rate of trait change; variance of evolutionary change distribution per speciation event
meanChange	Mean change per speciation event. Default is 0; change to simulate 'speciational trends'.
rootTrait	The trait value of the first taxon in the dataset; set to 0 by default.

Details

This function simulates continuous trait evolution where change occurs under a Brownian model, but only at events that create new 'species', either branching events or anagenesis (pseudospeciation); i.e. the types of 'morphological differentiation' which can be simulated in the function simFossilTaxa. This is sometimes referred to as cladogenetic or speciation trait evolution and is heavily related to Punctuated Equilibrium theory. Anagenetic shifts aren't cladogenetic events per

se (no branching!), so perhaps the best way to this of this function is it allows traits to change anytime `simFossilTaxa` created a new 'morphotaxon' in a simulation.

Importantly, the change only occurs at the base of 'new' species, thus allowing cladogenetic trait evolution to be asymmetrical at branching points: i.e. only one branch actually changes position in trait-space, as expected under a budding cladogenesis model. This distinction is important as converting the taxa matrix to a phylogeny and simulating the trait changes under a 'speciational' tree-transformation would assume that divergence occurred on both daughter lineages at each node. (This has been the standard approach for simulating cladogenetic trait change on trees).

Cryptic taxa generated with `prop.cryptic` in `simFossilTaxa` will not differ at all in trait values. These species will all be identical.

See this link for additional details: <http://nemagraptus.blogspot.com/2012/03/simulating-budding-cladogenetictrait.html>

Value

Retuns a vector of trait values for each taxon, with value names being the taxa IDs (column 1 of the input) with a 't' pasted (as with `rTree` in the `ape` library).

Author(s)

David W. Bapst

See Also

[simFossilTaxa](#),

This function is similar to Brownian motion simulation functions such as `rTraitCont` in `ape`, `sim.char` in `geiger` and `fastBM` in `phytools`.

See also [unitLengthTree](#) in this package and `speciationalTree` in the package `geiger`. These are tree transformation functions; together with BM simulation functions, they would be expected to have a similar effect as this function (when cladogenesis is 'bifurcating' and not 'budding'; see above).

Examples

```
set.seed(444)
taxa <- simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=30,maxtime=1000,plot=TRUE)
trait <- cladogeneticTraitCont(taxa)
tree <- taxa2phylo(taxa)
plotTraitgram(trait,tree,conf.int=FALSE)

#with cryptic speciation
taxa <- simFossilTaxa(p=0.1,q=0.1,prop.cryptic=0.5,nruns=1,mintaxa=30,maxtime=1000,plot=TRUE)
trait <- cladogeneticTraitCont(taxa)
tree <- taxa2phylo(taxa)
plotTraitgram(trait,tree,conf.int=FALSE)
```

compareTimescaling *Comparing the Time-Scaling of Trees*

Description

These functions take two trees and calculate the changes in node ages (for `compareNodeAges`) for shared clades or terminal branch lengths leading to shared tip taxa (for `compareTermBranches`).

Usage

```
compareNodeAges(tree1, tree2)
```

```
compareTermBranches(tree1, tree2)
```

Arguments

tree1	A time-scaled phylogeny of class 'phylo'
tree2	A time-scaled phylogeny of class 'phylo'

Details

These functions compare the time-scaling of two trees. Any taxa not-shared on both trees are dropped before analysis, based on tip labels.

As with many paleotree functions, calculations relating to time are done with respect to any included `$root.time` elements. If these are not present, the latest tip is assumed to be at the present day (`time=0`).

`compareNodeAges` calculates the changes in the clade ages among those clades shared by the two trees, relative to the first tree in absolute time. For example, a shift of +5 means the clade originates 5 time-units later in absolute time on the second tree, while a shift of -5 means the clade originated 5 time-units prior on the second tree.

`compareTermBranches` calculates the changes in the terminal branch lengths attached to tip taxa shared by the two trees, relative to the first tree. Thus, a shift of +5 means that terminal taxon has a terminal branch which is five time-units longer.

Value

Returns a vector of temporal shifts for nodes or terminal branches, dependent on the function applied. If `compareTermBranches` is used, the vector will have the tip names as labels.

Author(s)

David W. Bapst

See Also

[taxa2phylo](#), [phyloDiv](#)

Examples

```

set.seed(444)
taxa <- simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,maxExtant=0)
#get the true tree
tree1 <- taxa2phylo(taxa)
#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
tree2 <- timePaleoPhy(cladogram,rangesCont,type="basic")
#let's look at the distribution of node shifts
hist(compareNodeAges(tree1,tree2))
#let's look at the distribution of terminal branch lengths
hist(compareTermBranches(tree1,tree2))

set.seed(444)
tree1 <- rtree(10)
tree2 <- tree1
tree1$root.time <- 10
compareNodeAges(tree1,tree2)
compareTermBranches(tree1,tree2)

```

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 to produce a less-resolved tree.

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 the basic function `sample`. `degradeTree` can be conditioned to remove nodes of a particular depth with greater probability/frequency by setting `node.depth` to a value between zero (favoring the removal of 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.

Branch lengths are removed from the input tree prior to degradation and entirely absent from the output tree.

Value

Returns the modified tree as an object of class phylo, with no edge lengths.

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)

layout(1)
```

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 the number of branching nodes between the basal node of a branch and the root, not the absolute distance (i.e. branch length) to the root or the 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

Functions to plot 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 = TRUE, plotLogRich = FALSE, timelims = NU
```

```
taxicDivDisc(timeList, int.times = NULL, plot = TRUE, plotLogRich = FALSE, timelims = NULL, split.int =
```

```
phyloDiv(tree, int.length = 1, int.times = NULL, plot = TRUE, plotLogRich = FALSE, drop.ZLB = TRUE, time]
```

Arguments

<code>timeData</code>	Two-column matrix giving the per-taxon first and last appearances in absolute time. The matrices output by <code>simFossilTaxa</code> can also be supplied to <code>taxicDivCont</code> .
<code>timeList</code>	A list giving interval data. See details below.
<code>tree</code>	a time-scaled phylogeny of class <code>phylo</code>
<code>int.length</code>	The length of intervals used to make the diversity curve. Ignored if <code>int.times</code> is given.
<code>int.times</code>	A two-column matrix of the interval start and end times for calculating the diversity curve. If <code>NULL</code> , calculated internally.

<code>split.int</code>	For discrete time data, should calculated/input intervals be split at discrete time interval boundaries? If FALSE, can create apparent artifacts in calculating the diversity curve. See below.
<code>plot</code>	If true, diversity curve is plotted
<code>plotLogRich</code>	If true, taxic diversity plotted on log scale
<code>drop.cryptic</code>	If true, cryptic taxa are merged to form one taxon for estimating taxon curves. Only works for objects from <code>simFossilTaxa</code> .
<code>timelims</code>	Limits for the x (time) axis for diversity curve plots. Only affects plotting. Given as either NULL (the default) or as a vector of length two as for 'xlim' in the basic R function plot.
<code>drop.ZLB</code>	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, the true standing diversity may be somewhat lower than the measured (observed) quantity, particularly if intervals are 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. For diversity data in particularly large discrete time intervals, plotting this data in smaller bins which do not line up completely with the original intervals will create a 'spiky' diversity curve, as these smaller intersecting bins will have a large number of taxa which may have been present in either of the neighboring intervals. This will give these small bins an apparently high estimated standing diversity. This artifact is avoided with the default setting `split.int=TRUE`, which will split any input or calculated intervals so that they start and end at the boundaries of the discrete-time range bins.

The `timeList` object should be a list composed of two matrices, the first matrix giving by-interval start and end times (in absolute time), the second matrix giving the by-taxon first and last appearances in the intervals defined in the first matrix, 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 documentation for `binTimeData` for more information.

`phyloDiv` will resolve polytomies to be dichotomous nodes separated 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](#), [binTimeData](#)

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(1:2)
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

layout(1)
#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)

#an example of a 'spiky' diversity curve and why split.int is a good thing
set.seed(444)
taxa <- simFossilTaxa(p=0.1,q=0.1,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,maxExtant=0)
taxaDiv <- taxicDivCont(taxa)
#simulate a fossil record with imperfect sampling with sampleRanges()
rangesCont <- sampleRanges(taxa,r=0.5)
rangesDisc <- binTimeData(rangesCont,int.length=10)
#now let's plot with taxicDivDisc() but with the intervals from taxaDiv
  #by default, split.int=TRUE
taxicDivDisc(rangesDisc,int.times=taxaDiv[,1:2],split.int=TRUE)
#look pretty
#now let's turn off split.int
taxicDivDisc(rangesDisc,int.times=taxaDiv[,1:2],split.int=FALSE)
#looks 'spiky'!

```

expandTaxonTree	<i>Extrapolating Lower-Level Taxon Phylogenies from Higher-Level Taxon Trees</i>
-----------------	--

Description

This function takes a tree composed of higher-level taxa and a vector of lower-level taxa belonging to the set of higher-level taxa included in the input tree and produces a tree composed of the lower-level taxa, by treating the higher-level taxa as unresolved monophyletic polytomies. A user can also mark higher taxa as paraphyletic such that these are secondarily collapsed and do not form monophyletic clades in the output tree.

Usage

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

Arguments

taxonTree	A phylo object where tips represent higher taxa
taxaData	Character vector of higher taxa, with elements names equal to the lower taxa. See below.

collapse	Character vector of non-monophyletic higher taxa to be collapsed
plot	If true, plots a comparison between input and output trees

Details

The output tree will probably be a rough unresolved view of the relationships among the taxa, due to the treatment of higher-level taxa as polytomies. This is similar to the methods used in Webb and Donoghue (2005) and Friedman (2009). Any analyses should be done by resolving this tree with `multi2di` in the `ape` package or via the various time-scaling functions found in this package (`paleotree`).

The `taxaData` vector should have one element per lower-level taxon that is to be added to the tree. The name of each element in the vector should be the names of the lower-level taxa, which will be used as new tip labels of the output lower-taxon tree.

The `collapse` vector should be used if there is doubt about the monophyly of any of the higher-level taxa included in the input `taxonTree`. By default, all higher-level 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-level taxa expanded into polytomies and the lower-level taxa as the tip labels.

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 maximum likelihood to estimate per-interval 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 = FALSE, iterations = 10000, initial
```

Arguments

timeData	A 2 column matrix with the first and last occurrences 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)
iterations	Maximum number of iterations the optimizer is run for
initial	Values used as initial parameter value to initiate the optimizing function. The same starting value is used for all parameters

Details

This function uses maximum-likelihood solutions obtained by Foote (1997). These analyses are ideally applied to data from single stratigraphic section but potentially are 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 variation allowed among taxa in model parameters (extinction rate and sampling probability/rate). Essentially, taxa are divided into different (possibly overlapping) classes which have 'average' parameter values. These average parameters are multiplicatively-combined to calculate per-taxon parameters. For example, perhaps a user hypothesizes taxa that live in a particular environment have different characteristic sampling rates/probabilities, that taxa of several different major clades have different characteristic sampling parameters and that there may be several temporal shifts in the characteristic

extinction rate. 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 included as the `n_tbins` argument. A model where taxa differ in parameters across time, clades and environments will thus be fit and the AIC calculated, such that this model's fit can be compared to other (probably simpler) models.

By default, a 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 time intervals with different characteristic model parameters which are not defined a priori. The boundaries between time intervals with different characteristic parameters will thus be additional free-floating parameters included in the AIC calculation. If you have the prior inclination that sampling/extinction changed at a specific point in time, then separate the taxa that originated before and after that point as two different groups and include those classifications as a group in the arguments.

This function does not implement the finite window of observation modification for dealing with data that leads up to the recent (Foote, 1997). 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 intervals in `timeData` should be non-overlapping sequential intervals of roughly equal length. These 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 `$message` and `$convergence` elements of the output to make sure that convergence occurred. The likelihood surface can be very flat in some cases, particularly for small datasets (<100 taxa). If convergence was not reached, a warning message is communicated. If the optimizer does not converge, consider increasing iterations or changing the starting values.

Value

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

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

<code>Title</code>	Gives details of the analysis, such as the number and type of parameters included and the number of taxa analyzed
<code>parameters</code>	Maximum-likelihood parameters of the sampling model
<code>log.likelihood</code>	The maximum support (log-likelihood) value
<code>AICc</code>	The second-order Akaike's Information Criterion, corrected for small sample sizes

convergence	A number indicating status of convergence; if 0 then convergence was reached; see help file for optim for the respective error list
message	Messages output by optim; check to make sure that model convergence occurred

If multi-class models are fit, the element \$parameters will not be present, instead there will be several different elements that simply describe the characteristic parameter 'components' for each class, rather than representing the parameters of actual taxa in that class. As noted in the \$title, these should not be interpreted as the actual rates/probabilities of any real taxa but rather as factors which must be multiplied in combination with the estimates for 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 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)
print(SPres1) #let's look at the results
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$interval.boundaries[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 maximum likelihood to estimate instantaneous 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 = FALSE, iteration
```

Arguments

timeData	Two-column matrix of per-taxon first and last occurrences in absolute continuous 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)
iterations	Maximum number of iterations the optimizer is run for
initial	Values used as initial parameter value to initiate the optimizing function. The same starting value is used for all parameters

Details

This function uses maximum-likelihood solutions obtained by Foote (1997). These analyses are ideally applied to data from single stratigraphic section but potentially are 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 variation allowed among taxa in model parameters (extinction rate and sampling probability/rate). Essentially, taxa are divided into different (possibly overlapping) classes which have 'average' parameter values. These average parameters are multiplicatively-combined to calculate per-taxon parameters. For example, perhaps a user hypothesizes taxa that live in a particular environment have different characteristic sampling rates/probabilities, that taxa of several different major clades have different characteristic sampling parameters and that there may be several temporal shifts in the characteristic extinction rate. 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 included as the n_tbins argument. A model where taxa differ in parameters across time, clades and environments will thus be fit and the AIC calculated, such that this model's fit can be compared to other (probably simpler) models.

By default, a 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 time intervals with different characteristic model parameters which are not defined a priori. The boundaries between time intervals with different characteristic parameters will thus be additional free-floating parameters included in the AIC calculation. If you have the prior inclination that sampling/extinction changed at a specific point in time, then separate the taxa

that originated before and after that point as two different groups and include those classifications as a group in the arguments.

This function does not implement the finite window of observation modification for dealing with data that leads up to the recent (Foote, 1997). 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 and \$convergence elements of the output to make sure that convergence occurred. The likelihood surface can be very flat in some cases, particularly for small datasets (<100 taxa). If convergence was not reached, a warning message is communicated. If the optimizer does not converge, consider increasing iterations or changing the starting values.

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
parameters	Maximum-likelihood parameters of the sampling model
log.likelihood	The maximum support (log-likelihood) value
AICc	The second-order Akaike's Information Criterion, corrected for small sample sizes
convergence	A number indicating status of convergence; if 0 then convergence was reached; see help file for optim for the respective error list
message	Messages output by optim(); check to make sure that model convergence occurred

If multi-class models are fit, the element \$parameters will not be present, instead there will be several different elements that simply describe the characteristic parameter 'components' for each class, rather than representing the parameters of actual taxa in that class. As noted in the \$title, these should not be interpreted as the actual rates/probabilities of any real taxa but rather as factors which must be multiplied in combination with the estimates for 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 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)
print(SRres1) #that's all the results...

#pulling out the sampling rate from the results
sRate <- SRres1[[2]][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))
#the most simple model (the first value) has the lowest AICc (most informative!)
```

modifyTerminalBranches

Modify or Drop Terminal Branches of Various Types

Description

These functions modify terminal branches or drop certain terminal branches based on various criteria.

Usage

```
dropZLB(tree)
dropExtant(tree, tol = 0.01)
dropExtinct(tree, tol = 0.01, ignore.root.time = FALSE)
addTermBranchLength(tree, addtime = 0.001)
```

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
addtime	Extra amount of time to add to all terminal branch lengths.

Details

DropZLB drops tip-taxa that are attached to the tree via zero-length branches ("ZLBs"). This is sometimes useful for paleo-trees, as various time-scaling methods often produce these 'ZLBs'. This is advised for analyses of diversification/diversity, as these will appear as instantaneous speciation+extinction events, when in reality they probably represent ancestor-descendant relationships. Note this function only drops terminal tips that are on a ZLB. If you want to collapse 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 be the point in time when the tip furthest from the root (the latest tip) terminates.

If the input tree has 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.

addTermBranchLength adds 'addtime' to the terminal branch lengths of the tree. If there is a \$root.time element, this is increased by an amount equal to addtime. A negative amount can be input to reduce the length of terminal branches. However, if negative branch lengths are produced, the function fails and a warning is produced.

Value

Gives back a modified phylogeny as a phylo object

Author(s)

David W. Bapst

See Also

[phyloDiv](#), [drop.tip](#), [compareTermBranches](#)

Examples

```
set.seed(444)
#Simulate some fossil ranges with simFossilTaxa
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(1:2)
plot(ladderize(tree))
plot(ladderize(dropZLB(tree)))
layout(1)

#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)

#example using addTermBranchLength
set.seed(444)
treeA <- rtree(10)
treeB <- addTermBranchLength(treeA,1)
compareTermBranches(treeA,treeB)

```

Description

Calculates multiple diversity curves from a list of datasets of taxic ranges and/or phylogenetic trees, for the same intervals for all the individual datasets. A median curve with 95 percent quantile bounds can be calculated and plotted.

Usage

```
multiDiv(data, int.length = 1, plot = TRUE, split.int = TRUE, drop.ZLB=TRUE, drop.cryptic = FALSE, plotL
```

```
plotMultiDiv(results, plotLogRich = FALSE, timelims = NULL)
```

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?
drop.cryptic	If true, cryptic taxa are merged to form one taxon for estimating taxon curves. Only works for objects from simFossilTaxa.

<code>split.int</code>	For discrete time data, should calculated/input intervals be split at discrete time interval boundaries? If FALSE, can create apparent artifacts in calculating the diversity curve.
<code>results</code>	The output of a previous run of multiDiv
<code>plotLogRich</code>	If true, taxic diversity plotted on log scale
<code>timelims</code>	Limits for the x (time) axis for diversity curve plots. Only affects plotting. Given as either NULL (the default) or as a vector of length two as for 'xlim' in the basic R function plot.

Details

This function is essentially a wrapper for the individual diversity curve functions included in paleotree. multiDiv will intuitively decide whether input datasets are continuous-time taxic ranges, discrete-time (binned interval) taxic ranges or phylogenetic trees, as long as they are formatted as required by the respective diversity curve functions. A list that contains a mix of data types is entirely acceptable. A list of matrices output by simFossilTaxa is allowable, and treated as input for taxicDivCont. Data of an unknown type gives back an error.

The argument `split.int` splits intervals, if and only if discrete interval time data is included among the datasets. See the help file for taxicDivDisc to see an explanation of why `split.int=TRUE` by default is probably a good thing.

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 the author's 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 relationship to the standard error.

Value

A list composed of three elements will be invisibly returned:

<code>int.times</code>	A two column matrix giving interval start and end times
<code>div</code>	A matrix of measured diversities in particular intervals by rows, with each column representing a different dataset included in the input
<code>median.curve</code>	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 include: [phyloDiv](#), [taxicDivCont](#) and [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 = TRUE, 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 reconstructed 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 functions `traitgram` in the library `picante` and `phenogram` in the library `phytools`.

Examples

```
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	<i>Sampling Taxon Ranges</i>
--------------	------------------------------

Description

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

Usage

```
sampleRanges(taxad, r, alpha = 1, beta = 1, rTimeRatio = 1, modern.samp.prob = 1, min.taxa = 2,
  ranges.only = TRUE, minInt = 0.01, merge.cryptic = TRUE, alt.method = FALSE, plot = FALSE)
```

Arguments

taxad	A two-column matrix of per-taxon ranges. The five-column matrix output of simFossilTaxa can also be supplied.
r	Instantaneous average sampling rate per lineage time units; given as a vector of length one or length equal to the number of taxa
alpha	Alpha parameter of beta distribution; given as a vector of length one or length equal to the number of taxa
beta	Beta parameter of beta distribution; given as a vector of length one or length equal to the number of taxa
rTimeRatio	Ratio of most recent sampling rate over earliest sampling rate; given as a vector of length one or length equal to the number of taxa
modern.samp.prob	Probability of sampling living taxa at the present day (time=0), see below.
min.taxa	Minimum number of taxa sampled. The default is 2.
ranges.only	Give time of all sampling events as a list or taxon ranges only?
minInt	Minimum interval size used for simulating complex models
merge.cryptic	Should sampling events for cryptic species be merged into one taxon?
alt.method	Use the alternative method of discretizing time even if a simple model of sampling is being simulated?
plot	If true, plots the sampling models for each taxon against time

Details

This function implements a range of sampling models in continuous time. By default, sampling is simulated under the simplest model, where each sampling event occurs as a Poisson process under an instantaneous sampling rate (r) which is homogenous through time and across lineages (Foote, 1997). Under this model, the waiting times to sampling events are exponentially distributed, with an average waiting time of $1/r$. This useful property allows sampling to be rapidly simulated for many taxa under this simple model in sampleRanges, by repeatedly drawing waiting times between sampling events. This is the model that is run as long as alpha, beta and rTimeRatio are set to 1.

In addition to this simple model, `sampleRanges` also can consider a range of additional models, in particular including the `hatP` and `incP` options of Liow et al. (2010). To describe the behavior of these models, users alter the default values for `alpha`, `beta` and `rTimeRatio`. These parameters, and `r`, can either be a single value which describes the behavior of the entire dataset or vector, of same length as the number of taxa, which describes the per-taxon value. When any `rTimeRatio`, `alpha` or `beta` value is not equal to one, then the sampling rate will vary across duration of a taxon's temporal range. In general, setting `alpha` and `beta` equal to a value above 2 produce "hat" or bell-shaped curves where sampling rates peak at the midpoint of taxon ranges, while setting them unequal will produce asymmetric bell curves as in the beta function (Liow et al., 2010; Liow et al. set `alpha=beta=4`). `rTimeRatio` is the ratio of the sampling rate of the latest/most recent time divided by the earliest/oldest time.

The input `r` values will be interpreted differently based on whether one `r` value or per-taxon values were used. If one value was input, then it is assumed that `r` represent the grand mean `r` for entire dataset for purposes of time-varying `r`, such that if `rTimeRatio` is not equal to 1, taxa near the end and start of the dataset will have very different per-taxon mean sampling rate. If per-taxon values of `r` were input, then each `r` is consider the per-taxon mean sampling rate. These will not be changed, but any within-lineage variation is distributed so that the mean is still the input per-taxon value. This also changes the interpretation of `rTimeRatio`, such that when a single `r` value and `rTimeRatio` is given, it is assumed the ratio describes the change in sampling rates from the start of the dataset to the end, while if multiple values are given for either `r` or `rTimeRatio` will instead see the value as describing the ratio at the first and last times of each taxon.

These distinctions between models are important: it means that all models simulated in `sampleRanges` are effectively nested inside the most general model which contains `alpha`, `beta` and `rTimeRatio`.

Because sampling rate varies over time within lineages under any of these more complex models, sampling events cannot be quickly simulated as waiting times pulled from an exponential model. Instead, the taxon durations are discretized into a large number of small time intervals of length `minInt` (see above; `minInt` should be small enough that only one event could feasibly happen per interval) and the probability of an event occurring within each interval calculated and used to simulate sampling events. In general, this method is slower but otherwise comparable to the quicker waiting times method. See examples for a small test of this.

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

If `min.taxa` is set to zero, the simulation may produce output in which no taxa were ever sampled.

If `modern.samp.prob` is set to 1.0 (the default), then living taxa will always be sampled at least at the present day (if there are any living taxa). If the probability is less than 1, they will be sampled with that probability at the modern day.

By default, this function will merge sampling events from morphologically cryptic taxa, listing them as occurrences for the earliest member of that group. To change this behavior, set `merge.cryptic` to `FALSE`.

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` is `TRUE`, 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` is `FALSE` (the default), 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(1:2)
#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 might observe!

#let's try more complicated models!

#a pull-to-the-recent model with x5 increase over time similar to Liow et al.'s incP
layout(1:2)
rangesCont1 <- sampleRanges(taxa,r=0.5,rTimeRatio=5,plot=TRUE)
taxicDivCont(rangesCont1)

#a hat-shaped model
layout(1:2)
rangesCont1 <- sampleRanges(taxa,r=0.5,alpha=4,beta=4,plot=TRUE)
taxicDivCont(rangesCont1)

#a combination of these
layout(1:2)
```

```

rangesCont1 <- sampleRanges(taxa,r=0.5,alpha=4,beta=4,rTimeRatio=5,plot=TRUE)
taxicDivCont(rangesCont1)

#testing with cryptic speciation
layout(1)
taxaCrypt <- simFossilTaxa(p=0.1,q=0.1,prop.cryptic=0.5,nruns=1,mintaxa=20,maxtaxa=30,maxtime=1000,maxExtant=0,
rangesCrypt <- sampleRanges(taxaCrypt,r=0.5)
taxicDivCont(rangesCrypt)

## Not run:
#simulate a model where sampling rate evolves under brownian motion
tree<-taxa2phylo(taxa,obs=taxa[,3])
sampRateBM <- rTraitCont(tree)
sampRateBM <- sampRateBM-min(sampRateBM)
layout(1:2)
rangesCont1 <- sampleRanges(taxa,r=sampRateBM,plot=TRUE)
taxicDivCont(rangesCont1)

#evolving sampling rate, hat model and pull of the recent
layout(1:2)
rangesCont1 <- sampleRanges(taxa,r=sampRateBM,alpha=4,beta=4,rTimeRatio=5,plot=TRUE)
taxicDivCont(rangesCont1)
layout(1)

#the simpler model is simulated by pulling waiting times from an exponential
#more complicated models are simulated by discretizing time into small intervals
#are these two methods comparable?
#let's look at the number of taxa sampled under both methods
summary(replicate(100,sum(!is.na(sampleRanges(taxa,r=0.5,alt.method=FALSE)[,1]))))
summary(replicate(100,sum(!is.na(sampleRanges(taxa,r=0.5,alt.method=TRUE)[,1]))))
#they look pretty similar!

## End(Not run)

```

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)
pqRate2sProb(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 (λ)
q	Instantaneous rate of extinction (μ)
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 instantaneous 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 sensitivity 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, anag.rate = 0, prop.bifurc = 0, prop.cryptic = 0, nruns = 1,
  mintaxa = 1, maxtaxa = 1000, mintime = 1, maxtime = 1000, minExtant = 0,
  maxExtant = NULL, min.cond = TRUE, print.runs = FALSE, plot = FALSE)
```

```
simFossilTaxa_SRCCond(r, avgtaxa, p, q, anag.rate = 0, prop.bifurc = 0,
  prop.cryptic = 0, nruns = 1, maxtime = 1000, maxExtant = NULL,
  print.runs = FALSE, plot = FALSE)
```

Arguments

p	Instantaneous rate of speciation/branching
q	Instantaneous rate of extinction
anag.rate	Instantaneous rate of pseudoextinction/anagenesis
prop.bifurc	Proportion of morphological branching by bifurcating cladogenesis relative to budding cladogenesis
prop.cryptic	Proportion of cryptic speciation by relative to morphological branching, such as bifurcating and budding
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 enmeshes the simulation of speciation and extinction with explicit models of how lineages are morphologically differentiated in the fossil record, as morphotaxa are the basic units of paleontological estimates of diversity and phylogenetics.

simFossilTaxa runs many simulations of diversification and runs are only accepted for output if and when they meet the conditioning criteria, both minima and maxima. If min.cond is true (the default), simulations will be stopped and immediately accepted when clades satisfy mintime, mintaxa, minExtant and maxExtant (if the later is set). This stopping is technically almost immediate, see below. To reduce the effect of one conditioning criterion, simply set that limit to an arbitrarily low or high number (depending if it is a minimum or maximum constraint). If min.cond is false, simulation runs are not stopped and evaluated for output acceptance until they (a) go completely extinct or (b) hit either maxtaxa or maxtime. Whether the simulation runs are accepted or not for output is still dependent on mintaxa, mintime, minExtant and maxExtant. Note that some combinations of conditions, such as attempting to condition on a specific non-zero value of minExtant and maxExtant, may take a long time to find any acceptable simulation runs.

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, the run is accepted as a dataset for output.

Please note that mintaxa and maxtaxa refer to the number of static morphotaxa 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 under 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.

This function now also includes the ability to simulate cryptic cladogenesis. The available patterns of morphological speciation can be seen as a gradient, where cryptic cladogenesis has no morphological shifts in either daughter branches after a branching event, budding cladogenesis results in one morphological shift and shifts occur along both daughter lineages in bifurcating cladogenesis. The argument `prop.cryptic` dictates what proportion of cladogenesis events (with rate p) are cryptic versus those that have some morphological divergence; `prop.bifurc` controls the proportion of morphologically divergent cladogenesis which is bifurcating relative to budding. Thus, for example, the probability of a given cladogenesis event being budding is $(1 - \text{prop.cryptic}) * \text{prop.bifurc}$.

For the purposes of `mintaxa` and `maxtaxa`, the number of unique morphologically distinguishable taxa is checked (i.e. the number of unique values in column 6 of the simulated data). This is not true of `minExtant` and `maxExtant`, which just looks at the number of lineages present at the modern day. See below about the output data structure to see how information about cryptic cladogenesis is recorded. New features have been added to `taxa2phylo`, `taxa2cladogram` and `taxicDivCont` which change how cryptic species are handled.

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` simulates single taxa until they go extinct or exceed `maxtime`. This means the function may have fully simulated some lineages for thousands of time-steps while others are not yet simulated, and thus sometimes overshoot constraints on the number of taxa. This function will automatically discard any runs where the number of taxa exceeds $2 \times \text{maxtaxa}$ to avoid blowing up computation time. This is likely to happen under a pure-birth scenario; I suggest using low `maxtime` settings if doing a pure-birth simulation.

`simFossilTaxa_SRCond` is a wrapper for `simFossilTaxa` for when clades of a particular size are desired, post-sampling. This function accomplishes this task by first calculating the expected proportion of taxa sampled, given the sampling rate and the rates which control lineage termination: extinction, anagenesis and bifurcation. The average original clade size needed to produce the number of sampled taxa given by `avgtaxa` is calculated with the following equation:

$$N = (\text{averagenumberoftaxadesired}) / (1 - \exp(-r / (q + \text{anag.rate} + (\text{prop.bifurc} * p))))$$

We will call that quantity N . Note that the quantity $(\text{prop.bifurc} * p)$ describes the rate of bifurcation when there is no cryptic cladogenesis, as `prop.bifurc` is the ratio of budding to bifurcating cladogenesis. This equation will diverge in ways that are not easily predicted as the rate of cryptic speciation increases.

Next, this value is used with `simFossilTaxa`, where `mintaxa` is set to N and `maxtaxa` set to $2 * N$. `simFossilTaxa_SRCond` will generally produce simulated datasets that are generally of that size or larger post-sampling (although there can be some variance). Some combinations of parameters 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.

More details on this function's design can be read here: <http://nemagraptus.blogspot.com/2012/04/simulating-fossil-record.html>

Value

Both of these functions give back a list containing `nruns` number of taxa datasets, where each element is a matrix. If `nruns=1`, the output is not a list but just a single matrix. Sampling has not been simulated in the output for either function; the output represents the 'true' history of the simulated clade.

For each dataset, the output is a six 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). The sixth column (named "looks.like") gives information about the morphological distinguishability of taxa; if they match the taxon ID, they are not cryptic. If they do not match, then this column identifies which taxon id they would be identified as.

Each matrix of simulated data also has rownames, generally of the form "t1" and "t2", where the number is the taxon id. Cryptic taxa are instead named in the form of "t1.2" and "t5.3", where the first number is the taxon which they are a cryptic descendant of (i.e. column 6 of the matrix, "looks.like"). The second number, after the period, is the rank order of taxa in that cryptic group of taxa. Taxa which are the common ancestor of a cryptic lineage are also given a unique naming convention, of the form "t1.1" and "t5.1", where the first number is the taxon id and the second number communicates that this is the first species in a cryptic lineage.

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.
- Hartmann, K., D. Wong, and T. Stadler. 2010 Sampling Trees from Evolutionary Models. *Systematic Biology* **59**(4):465–476.
- 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, anag.rate=0.05, prop.bifurc=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)

#some other options with cryptic speciation
taxaCrypt1 <- simFossilTaxa(p=0.1,q=0.1, anag.rate=0, prop.bifurc=0, prop.crypt=0.5, mintaxa=30, maxtaxa=60, maxExtant=0, nruns=1, plot=TRUE)
tree1 <- taxa2phylo(taxaCrypt1, plot=TRUE)
taxaCrypt2 <- simFossilTaxa(p=0.1,q=0.1, anag.rate=0.05, prop.bifurc=0.5, prop.crypt=0.5, mintaxa=30, maxtaxa=60, maxExtant=0, nruns=1, plot=TRUE)
tree2 <- taxa2phylo(taxaCrypt2, plot=TRUE)
taxaCrypt3 <- simFossilTaxa(p=0.1,q=0.1, anag.rate=0.05, prop.bifurc=0, prop.crypt=1, mintaxa=30, maxtaxa=60, maxExtant=0, nruns=1, plot=TRUE)
tree3 <- taxa2phylo(taxaCrypt3, 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=FALSE)
#use print.run to know how many simulations were accepted of the total generated
layout(1:2)
#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")

#pure-birth example
#note that conditioning is tricky
layout(1)
taxa <- simFossilTaxa(p=0.1,q=0, mintime=10, mintaxa=100, maxtime=100, maxtaxa=100, nruns=10, plot=TRUE)

#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(1:2)
#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
layout(1)
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 such that lineages may be missing and terminal branches are truncated, but node-times are known perfectly

Usage

```

simPaleoTrees(p, q, r, ntrees = 1, all.extinct = FALSE, modern.samp.prob = 1.0, mintime = 1,
  maxtime = 100, mintaxa = 2, maxtaxa = 500, drop.zlb = TRUE, print.runs = FALSE, plot = FALSE)

```

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.prob	Probability of sampling living taxa at the present day (time=0), see documentation for sampleRanges
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?
print.runs	Print the proportion of simulations accepted for output?
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` as `TRUE` and other minimal conditioning, so as 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.

The `print.runs` argument does not work precisely as in `simFossilTaxa`; it only counts how many accepted datasets from `simFossilTaxa` are acceptable for output after the simulation of sampling.

Value

Output is an object of class `multiphylo` containing the simulated phylogenies, unless `ntrees` is one in which case the output is a phylogeny of class `'phylo'`.

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` to each phylo object.

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,print.runs=TRUE,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.prob=TRUE,print.runs=TRUE)
#number of tips
sapply(trees,Ntip)
```

srcTimescaling

SampRate-Calibrated Timescaling of Paleo-Phylogenies

Description

Timescales an unscaled cladogram of fossil taxa, using 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 algorithm to resolve polytomies randomly and infer potential ancestor-descendant relationships, simultaneous with the time-scaling treatment.

Usage

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

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

Arguments

tree	An unscaled cladogram of fossil taxa
timeData	Two-column matrix of first and last occurrences in absolute continuous time, with rownames as the taxon IDs used on the tree
sampRate	Either a single estimate of the instantaneous 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 is 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 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 they are numbered in the tree\$edge matrix (note that in tree\$edge, the tips are given first Ntip numbers and these are ignored here). 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.
FAD.only	Should the tips represent observation times at the start of the taxon ranges? If rand.obs is 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. If TRUE, result is similar to when terminal ranges are no added on with timePaleoPhy. If FAD.only and rand.obs are both TRUE, the function will stop and a warning will be produced.
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
nonstoch.bin	If true, dates are not stochastically pulled from uniform distributions. See below for more details.
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) algorithm 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.

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.

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 algorithm) 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.

`srcTimePaleoPhy` is only applicable to datasets with taxon occurrences 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.

By setting the argument `nonstoch.bin` to `TRUE` in `bin_srcTimePaleoPhy`, the dates are NOT stochastically pulled from uniform bins but instead FADs are assigned to the earliest time of whichever interval they were placed in and LADs are placed at the most recent time in their placed interval. This option may be useful for plotting. The `sites` argument becomes arbitrary if `nonstoch.bin` is `TRUE`.

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[[2]][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 increase ntrees
ttrees <- srcTimePaleoPhy(cladogram,rangesCont,sampRate=sRate,ntrees=9,plot=FALSE)
#let's compare nine of them at once in a plot
layout(matrix(1:9,3,3));parOrig<-par(mar=c(0,0,0,0))
for(i in 1:9){plot(ladderize(ttrees[[i]]),show.tip.label=FALSE)}
#they are all a bit different!

#can plot the median diversity curve with multiDiv
layout(1); par(parOrig)
multiDiv(ttrees)

#using node.mins
#let's say we have (molecular??) evidence that node #5 is at least 1200 time-units ago
nodeDates <- rep(NA,(Nnode(cladogram)-1))
nodeDates[5]<-1200
ttree <- srcTimePaleoPhy(cladogram,rangesCont,sampRate=sRate,ntrees=1,node.mins=nodeDates,plot=TRUE)

#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[[2]][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)
#can also force the appearance timings not to be chosen stochastically
ttree1 <- bin_srcTimePaleoPhy(cladogram,rangesDisc,sampRate=sRate1,ntrees=1,nonstoch.bin=TRUE,plot=TRUE)
phyloDiv(ttree1)

```

taxa2cladogram

Convert Simulated Taxon Data into a Cladogram

Description

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

Usage

```
taxa2cladogram(taxad, drop.cryptic = FALSE, plot = FALSE)
```

Arguments

taxad	A five-column matrix of taxonomic data, as output by simFossilTaxa
drop.cryptic	Should cryptic species be dropped (except for the first)? Not dropped by default.
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, as noted by Wagner and Erwin, 1995.

If there are cryptic taxa present in the output from `simFossilTaxa`, these and any of their morphologically distinguishable descendants are collapsed into a polytomy to simulate the expected pattern of lack of phylogenetic resolution. In addition to this merging, they can be dropped, and only the first 'species' of each cryptic taxon assemblage is listed among the tip taxa (what we would actually expect to obtain, as wouldn't recognize cryptic taxa as different OTUs). By default, cryptic taxa are not dropped so that the same number of taxa as in the simulated data is retained.

Value

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

The tip labels are the rownames from the simulation input; see `simFossilTaxa` documentation for details.

Author(s)

David W. Bapst

References

- Foote, M. 1996 On the Probability of Ancestors in the Fossil Record. *Paleobiology* **22**(2):141-151.
- Wagner, P., and D. Erwin. 1995 Phylogenetic patterns as tests of speciation models. New approaches to speciation in the fossil record. Columbia University Press, New York:87-122.

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(1:2)
cladogram<-taxa2cladogram(taxa,plot=TRUE)
#compare the "real" time-scaled tree of taxon last occurrences (taxa2phylo) to the 'ideal' cladogram
tree<-taxa2phylo(taxa,plot=TRUE)

#testing with cryptic speciation
taxaCrypt<-simFossilTaxa(p=0.1,q=0.1,prop.cryptic=0.5,nruns=1,mintaxa=10,maxtaxa=20,maxtime=1000,maxExtant=0)
layout(1:2);parOrig<-par(mar=c(0,0,0,0))
cladoCrypt1<-taxa2cladogram(taxaCrypt,drop.cryptic=FALSE)
plot(cladoCrypt1)
```

```
cladoCrypt2<-taxa2cladogram(taxaCrypt,drop.cryptic=TRUE)
plot(cladoCrypt2)
par(parOrig);layout(1) #reset plotting
```

taxa2phylo

Convert Simulated Taxon Data into a Phylogeny

Description

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

Usage

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

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 described in the documentation for taxa2cladogram, the relationships among morphotaxa in the fossil record are difficult to describe in terms of traditional phylogenies. One possibility is to arbitrarily choose particular instantaneous points of time in the range of some taxa and describe the time-scaled relationships of the populations present at those dates. This is the tactic used by taxa2phylo.

By default, the dates selected ('obs-time' argument) are the last occurrences of the taxon, so a simple use of this function will produce a time-scaled tree which describes the relationships of the populations present at the last occurrence of each taxon in the sampled data. Alternatively, obs_time can be supplied with different dates within the taxon ranges.

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.

Cryptic taxa are not dropped or merged as can be done with taxa2cladogram. The purpose of taxa2phylo is to obtain the 'true' pattern of evolution for the observation times, independent of what we would actually be able to recover.

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

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.

The tip labels are the rownames from the simulation input; see simFossilTaxa documentation for details.

Note

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 and relationships perfectly, which is almost impossible given the undersampled 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.

Unlike taxa2cladogram, this function does not merge cryptic taxa in output from simFossilTaxa and I do not offer an option to secondarily drop them. The tip labels should provide the necessary information for users to drop such taxa, however. See simFossilTaxa.

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 for each taxon
rangesCont<-sampleRanges(taxa,r=0.5)
tree<-taxa2phylo(taxa,obs_time=rangesCont[,2])
phyloDiv(tree,drop.ZLB=FALSE)
#note that it drops taxa which were never sampled!

#testing with cryptic speciation
taxaCrypt<-simFossilTaxa(p=0.1,q=0.1,prop.cryptic=0.5,nruns=1,mintaxa=10,maxtaxa=20,maxtime=1000,maxExtant=0)
treeCrypt<-taxa2phylo(taxaCrypt)
layout(1)
plot(treeCrypt)
```

timePaleoPhy *Timescaling of Paleo-Phylogenies*

Description

Timescales an unscaled cladogram of fossil taxa using information on their temporal ranges, using various methods. Also can resolve polytomies randomly and output samples of randomly-resolved trees.

Usage

```
timePaleoPhy(tree, timeData, type = "basic", vartime = NULL, ntrees = 1, randres = FALSE,
  add.term = FALSE, rand.obs = FALSE, node.mins = NULL, plot = FALSE)
```

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

Arguments

tree	An unscaled cladogram of fossil taxa
timeData	Two-column matrix of first and last occurrences in absolute continuous 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.

<code>node.mins</code>	Minimum ages of nodes on the tree. The minimum dates of nodes can be set using <code>node.mins</code> ; this argument takes a vector of the same length as the number of nodes, with dates given in the same order as nodes are they are numbered in the <code>tree\$edge</code> matrix (note that in <code>tree\$edge</code> , the tips are given first Ntip numbers and these are ignored here). Not all nodes need be set; those without minimum dates can be given as NA in <code>node.mins</code> .
<code>plot</code>	If true, plots the input and output phylogenies
<code>timeList</code>	A list composed of two matrices giving interval times and taxon appearance datums, as would be output by <code>binTimeData</code> . The rownames of the second matrix should be the taxon IDs
<code>nonstoch.bin</code>	If true, dates are not stochastically pulled from uniform distributions. See below for more details.
<code>sites</code>	Optional two column matrix, composed of site IDs for taxon FADs and LADs. The <code>sites</code> argument allows users to constrain the placement of dates in <code>bin_timePaleoPhy</code> by restricting multiple fossil taxa whose FADs or LADs are from the same very temporally restricted sites (such as fossil-rich Lagerstätten) to always have the same date, across many iterations of time-scaled trees from <code>bin_timePaleoPhy</code> . 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 <code>timeList</code> . 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 (Smith, 1994).

"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.

These functions cannot time-scale branches relative to reconstructed character changes along branches, as used by Lloyd et al. (2012).

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.

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.

By setting the argument nonstoch.bin to TRUE, the dates are NOT stochastically pulled from uniform bins but instead FADs are assigned to the earliest time of whichever interval they were placed in and LADs are placed at the most recent time in their placed interval. This option may be useful for plotting. The sites argument becomes arbitrary if nonstoch.bin is TRUE.

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):1485-91488.
- 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 Evolution: Discrete Character Change in the Evolution of Lungfish(Sarcopterygii, Dipnoi). *Evolution* **66**(2):330–348.
- Smith, A. B. 1994 Systematics and the fossil record: documenting evolutionary patterns. Blackwell Scientific, Oxford.

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? (See Wagner and Erwin 1995 to see why)
#can randomly resolve trees using the argument randres
#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

#if we set ntrees>1, timePaleoPhy will make multiple time-trees
ttrees <- timePaleoPhy(cladogram,rangesCont,type="basic",ntrees=9,randres=TRUE,add.term=TRUE,plot=TRUE)
#let's compare nine of them at once in a plot
layout(matrix(1:9,3,3))
for(i in 1:9){plot(ladderize(ttrees[[i]]),show.tip.label=FALSE,no.margin=TRUE)}
#they are all a bit different!

#can plot the median diversity curve with multiDiv
layout(1)
multiDiv(ttrees)

#compare different methods of timePaleoPhy
layout(matrix(1:6,3,2));parOrig <- 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();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();text(x=55,y=23,"type=aba",adj=c(0,0.5),cex=1.2)
plot(ladderize(timePaleoPhy(cladogram,rangesCont,type="z1ba",vartime=1,add.term=TRUE)))
```

```

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();text(x=55,y=23,"type=mbl",adj=c(0,0.5),cex=1.2)
layout(1);par(parOrig)

#using node.mins
#let's say we have (molecular??) evidence that node #5 is at least 1200 time-units ago
nodeDates <- rep(NA,(Nnode(cladogram)-1))
nodeDates[5] <- 1200
ttree1 <- timePaleoPhy(cladogram,rangesCont,type="basic",
randres=FALSE,node.mins=nodeDates,plot=TRUE)
ttree2 <- timePaleoPhy(cladogram,rangesCont,type="basic",
randres=TRUE,node.mins=nodeDates,plot=TRUE)

#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)
ttreeB1 <- bin_timePaleoPhy(cladogram,rangesDisc,type="basic",ntrees=1,randres=TRUE,add.term=TRUE,plot=FALSE)
#notice the warning it prints!
phyloDiv(ttreesB1)
#can also force the appearance timings not to be chosen stochastically
ttreeB2 <- bin_timePaleoPhy(cladogram,rangesDisc,type="basic",ntrees=1,nonstoch.bin=TRUE,randres=TRUE,add.term=TRUE)
phyloDiv(ttreesB2)

```

timeSliceTree

Time-Slicing a Phylogeny

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(ttrees, sliceTime, drop.extinct = FALSE, plot = TRUE)
```

Arguments

ttrees	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 ttrees 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 ttrees, 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 tree\$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 tree 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 extant-only phylogeny studies
#of diversification...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

As alternative to using this for comparative studies, see [timePaleoPhy](#)
See also `speciationalTree` in the package `geiger`, which does essentially the same thing.

Examples

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

Index

ace, 26
addTermBranchLength
 (modifyTerminalBranches), 21
ape, 3

bin_srcTimePaleoPhy (srcTimescaling), 39
bin_timePaleoPhy (timePaleoPhy), 47
bind.tree, 14
binTimeData, 3, 12, 29, 42, 50

cladogeneticTraitCont, 5
compareNodeAges (compareTimescaling), 7
compareTermBranches, 22
compareTermBranches
 (compareTimescaling), 7
compareTimescaling, 7

degradeTree, 8
depthRainbow, 9
di2multi, 9, 22
DiversityCurves, 10
drop.tip, 22
dropExtant, 52
dropExtant (modifyTerminalBranches), 21
dropExtinct, 52
dropExtinct (modifyTerminalBranches), 21
dropZLB (modifyTerminalBranches), 21

expandTaxonTree, 13

getAncStates, 26
getSampProbDisc, 15, 21, 32
getSampRateCont, 17, 18, 32, 42

ltt.plot, 12

modifyTerminalBranches, 21
multi2di, 14, 42, 50
multiDiv, 12, 23

paleotree (paleotree-package), 2

paleotree-package, 2
phyloDiv, 7, 22, 24, 52
phyloDiv (DiversityCurves), 10
plotMultiDiv (multiDiv), 23
plotTraitgram, 25
pqsRate2sProb (SamplingConv), 30
probAnc (SamplingConv), 30

qsProb2Comp, 17
qsProb2Comp (SamplingConv), 30
qsRate2Comp, 21
qsRate2Comp (SamplingConv), 30

sampleRanges, 4, 27, 32, 36, 38
SamplingConv, 30
simFossilTaxa, 4, 6, 29, 32, 38, 44, 46
simFossilTaxa_SRCond (simFossilTaxa), 32
simPaleoTrees, 36, 37
sProb2sRate, 17
sProb2sRate (SamplingConv), 30
sRate2sProb, 21
sRate2sProb (SamplingConv), 30
srcTimePaleoPhy, 50
srcTimePaleoPhy (srcTimescaling), 39
srcTimescaling, 39

taxa2cladogram, 36, 43, 46
taxa2phylo, 7, 36, 38, 44, 45
taxicDivCont, 4, 24
taxicDivCont (DiversityCurves), 10
taxicDivDisc, 24
taxicDivDisc (DiversityCurves), 10
timePaleoPhy, 42, 47, 53
timeSliceTree, 12, 51

unitLengthTree, 6, 52