

# Package: TreePar (via r-universe)

August 10, 2024

**Type** Package

**Title** Estimating Birth and Death Rates Based on Phylogenies

**Version** 3.3

**Date** 2015-01-02

**Author** Tanja Stadler

**Maintainer** Tanja Stadler <tanja.stadler@bsse.ethz.ch>

**Depends** ape, Matrix, subplex, TreeSim (>= 2.1), deSolve

**Description** (i) For a given species phylogeny on present day data which is calibrated to calendar-time, a method for estimating maximum likelihood speciation and extinction processes is provided. The method allows for non-constant rates. Rates may change (1) as a function of time, i.e. rate shifts at specified times or mass extinction events (likelihood implemented as LikShifts(), optimization as bd.shifts.optim() and visualized as bd.shifts.plot()) or (2) as a function of the number of species, i.e. density-dependence (likelihood implemented as LikDD() and optimization as bd.densdep.optim()) or (3) extinction rate may be a function of species age (likelihood implemented as LikAge() and optimization as bd.age.optim.matlab()). Note that the methods take into account the whole phylogeny, in particular it accounts for the "pull of the present" effect. (1-3) can take into account incomplete species sampling, as long as each species has the same probability of being sampled. For a given phylogeny on higher taxa (i.e. all but one species per taxa are missing), where the number of species is known within each higher taxa, speciation and extinction rates can be estimated under model (1) (implemented within LikShifts() and bd.shifts.optim() with groups !=0). (ii) For a given phylogeny with sequentially sampled tips, e.g. a virus phylogeny, rates can be estimated under a model where rates vary across time using bdsky.st.optim() based on likelihood LikShiftsSTT() (extending LikShifts() and bd.shifts.optim()). Furthermore, rates may vary as a function of host types using LikTypesSTT() (multitype

branching process extending functions in R package diversitree). This function can furthermore calculate the likelihood under an epidemiological model where infected individuals are first exposed and then infectious.

**License** GPL-2

**LazyLoad** yes

**Repository** <https://phylotastic.r-universe.dev>

**RemoteUrl** <https://github.com/tanja819/TreePar>

**RemoteRef** HEAD

**RemoteSha** 77d33da3f37f757e8547cae5c4b072341cf8db10

## Contents

TreePar-package . . . . .	2
addroot . . . . .	3
bd.age.optim.matlab . . . . .	4
bd.densdep.optim . . . . .	6
bd.shifts.optim . . . . .	8
bd.shifts.plot . . . . .	11
bdsky.stt.optim . . . . .	13
create.mat . . . . .	14
get.groups . . . . .	16
LikAge . . . . .	16
LikConstant . . . . .	18
LikConstantn . . . . .	19
LikDD . . . . .	20
LikShifts . . . . .	22
LikShiftsPP . . . . .	25
LikShiftsSTT . . . . .	26
LikTypesSTT . . . . .	28
<b>Index</b>	<b>32</b>

---

TreePar-package

*Estimating Birth and Death Rates Based on Phylogenies*

---

## Description

(i) For a given species phylogeny on present day data which is calibrated to calendar-time, a method for estimating maximum likelihood speciation and extinction processes is provided. The method allows for non-constant rates. Rates may change (1) as a function of time, i.e. rate shifts at specified times or mass extinction events (likelihood implemented as `LikShifts()`, optimization as `bd.shifts.optim()` and visualized as `bd.shifts.plot()`) or (2) as a function of the number of species, i.e. density-dependence (likelihood implemented as `LikDD()` and optimization as `bd.densdep.optim()`)

or (3) extinction rate may be a function of species age (likelihood implemented as LikAge() and optimization as bd.age.optim.matlab()). Note that the methods take into account the whole phylogeny, in particular it accounts for the "pull of the present" effect. (1-3) can take into account incomplete species sampling, as long as each species has the same probability of being sampled. For a given phylogeny on higher taxa (i.e. all but one species per taxa are missing), where the number of species is known within each higher taxa, speciation and extinction rates can be estimated under model (1) (implemented within LikShifts() and bd.shifts.optim() with groups !=0). (ii) For a given phylogeny with sequentially sampled tips, e.g. a virus phylogeny, rates can be estimated under a model where rates vary across time using bdsky.stt.optim() based on likelihood LikShiftsSTT() (extending LikShifts() and bd.shifts.optim()). Furthermore, rates may vary as a function of host types using LikTypesSTT() (multitype branching process extending functions in R package diversitree). This function can furthermore calculate the likelihood under an epidemiological model where infected individuals are first exposed and then infectious.

## Details

Package:	TreePar
Type:	Package
Version:	3.3
Date:	2015-01-02
License:	GPL-2
LazyLoad:	yes

## Author(s)

Tanja Stadler <<http://www.bsse.ethz.ch/cEvo>>

## References

T. Stadler. Mammalian phylogeny reveals recent diversification rate shifts. Proc. Nat. Acad. Sci., 108(15): 6187-6192, 2011.

## See Also

ape TreeSim

---

addroot

*addroot: Adds a root edge ancestral to the first branching event.*

---

## Description

addroot adds a root edge ancestral to the first branching event.

**Usage**

```
addroot(tree,rootlength)
```

**Arguments**

tree	Tree of class phylo.
rootlength	Length of the root edge to be added.

**Value**

res	Tree of class phylo with root edge added to \$edge and \$edge.length (alternative is to store root edge in \$root.edge as done in ape).
-----	---

**Author(s)**

Tanja Stadler

**Examples**

```
set.seed(1)
# Simulate a tree
tree<-sim.bd.taxa(20,1,2,1,complete=FALSE,stochsampling=TRUE)
# add the root edge to the vector tree$edge and tree$edge.length
addroot(tree[[1]],tree[[1]]$root.edge)
```

---

bd.age.optim.matlab	<i>bd.age.optim.matlab: Estimating speciation rate and age-dependent extinction rate in phylogenies.</i>
---------------------	--

---

**Description**

bd.age.optim.matlab estimates the maximum likelihood speciation and extinction rates in a (possibly incomplete sampled) phylogeny. Speciation rate is constant, time to extinction is assumed to be a Gamma distribution.

**Usage**

```
bd.age.optim.matlab(x, lambdainit=1, kinit=1, thetainit=2, sampling, model="G", root=1,
inputformat=0,precision=4,numgridpts=500, path,matfilename="setup")
```

**Arguments**

x	Vector of speciation times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using getx(TREE).
lambdainit	Speciation rate at the initial point from which optimization starts.

kinit	Shape parameter of the Gamma distribution modelling time to extinction at the initial point from which optimization starts.
thetainit	Scale parameter of the Gamma distribution modelling time to extinction at the initial point from which optimization starts.
sampling	Probability to sample a present day lineage.
model	Model assumed for time to extinction. "G": Gamma distribution, i.e. shape parameter is estimated. "E": Exponential distribution, i.e. shape parameter is set to 1.
root	If root=1, max(x) is the crown age and the likelihood is conditioned on the crown age. If root=0, max(x) is the stem age and the likelihood is conditioned on the stem age.
inputformat	If 0, then speciation time vector 'x' is supplied. If 1, then mat-file input (obtained by create.mat) in 'matfilename' is supplied, and 'x' is not considered.
precision	Number of decimal places returned from the numerical likelihood calculation.
numgridpts	Number of gridpoints (evenly spaced time points between zero and the tree age, max(x)) used in the numerical likelihood calculation. The higher this number, the more accurate the numerical results will be.
path	Path where your MCR (Matlab Compiler Runtime) is installed (for details see ?LikAge).
matfilename	Only relevant if inputformat=1; then matfilename="setup" if input file is setup.mat.

**Value**

res	Vector of four numbers: negative log likelihood value and maximum likelihood estimate of lambda, k, theta.
-----	--

**Note**

This function relies on compiled Matlab applications that need to be downloaded. For setting up the environment to run the function, please refer to 'Notes' in function 'LikAge'.

**Author(s)**

Helen Alexander, Tanja Stadler

**References**

H. Alexander, A. Lambert, T. Stadler. Quantifying Age-Dependent Extinction from Species Phylogenies. Submitted.

**Examples**

```
## You need MCR to run the example
## directory where MCR is installed
# path<-"Applications/MATLAB_R2014a_node.app/"
## location of folder TreePar_Matlab
#locationMat<-"Users/tstadler/Documents/Data/Uni/Research/R/TreeParProject/TreePar_Matlab"
```

```
##
# setwd(locationMat)
# x<-read.table("branchingtimes.txt")
# numgridpts<-500
# lambda<-2
# k<-1
# theta<-1
# sampling<-1
# setwd(paste(locationMat,"/compiled",sep=""))
# LikAge(x,lambda, k, theta, sampling, path=path)
# outG <- bd.age.optim.matlab(x, lambda, k, theta, sampling, path=path)
# outE <- bd.age.optim.matlab(x, lambda, k, theta, sampling, path=path, model="E")
## compare with bd.shifts.optim for exponential lifetime
# outEcompare<-c(outE[1],1/(outE[2]*outE[3]),outE[2]-1/outE[3])
# outShifts <- bd.shifts.optim(x,sampling=c(1),survival=1)[[2]]
```

---

bd.densdep.optim	<i>bd.densdep.optim: Estimating maximum likelihood speciation and extinction rates in phylogenies under a density-dependent speciation model.</i>
------------------	---

---

## Description

bd.densdep.optim estimates the maximum likelihood speciation and extinction rates under a density-dependent speciation model. Speciation rate is a function of the number of species  $N$ ,  $\lambda(N) = \max(0, \lambda(1 - N/K))$ , where  $K$  is the carrying capacity and  $\lambda$  the speciation rate when  $N \ll K$ . Extinction rate is  $\mu$  (constant). For a computationally much faster implementation, please optimize the likelihood function with `runExpoTree` in R package `expoTree`. In contrast to `bd.densdep.optim`, `runExpoTree` can handle trees with tips sampled sequentially through time.

## Usage

```
bd.densdep.optim(x,minK,maxK,discrete=TRUE,continuous=FALSE,lambdainit=2,
muinit=1,Kinit=0,Yule=FALSE,muset=0,rho=1,model=-1)
```

## Arguments

x	Vector of speciation times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using <code>getx(TREE)</code> .
minK	Minimal value of $K$ (when <code>discrete=TRUE</code> ). Default is <code>minK = (number of species)</code> .
maxK	Maximal value of $K$ (when <code>discrete=TRUE</code> ). Default is <code>maxK = 1.5(number of species)</code> .
discrete	If <code>discrete=TRUE</code> , the likelihood function is maximized with $K$ being an integer and the minimal size being <code>minK</code> and the maximal size being <code>maxK</code> .

continuous	If continuous=TRUE, the likelihood function is maximized with K being a continuous parameter. The function subplex is used for optimization and sometimes gets stuck at a non-optimal K. Thus it is recommended to also calculate with discrete=TRUE.
lambdainit	Initial lambda value for optimization when K is continuous (default is 2).
muinit	Initial mu value for optimization when K is continuous (default is 1).
Kinit	Initial K value for optimization when K is continuous (default is Kinit=0 which automatically sets Kinit=(number of species)+1).
Yule	Yule=FALSE is default. Yule=TRUE fixes mu=0, i.e. no extinction.
muset	muset=0 (default) maximizes over the whole parameter range. muset>0 means that the optimization is done over all mu>muset. muset<0 fixes mu=-muset.
rho	rho=1 is default meaning all present-day species are sampled. 0<rho<1 assumes that the phylogeny is incomplete, and each present-day species is included with probability rho. rho=-1 means any number of present-day species $N \geq n$ has given rise to a sample of size n with probability 1. rho<-1 means that any number n,n+1,...,(-k) of present-day species may have given rise to a sample of size n with probability 1. rho>1 means that exactly rho>n present-day species gave rise to the sample n with probability 1.
model	model=-1 (default) is the density-dependent model. model=0 (only relevant for testing purposes) assumes that lambda is constant for number of species < K, and 0 for number of species $\geq K$ . model=0 is used for testing / comparing to constant rate model implemented in bd.shifts.optim.

### Value

res	Maximum likelihood speciation rate lambda and extinction rate mu and the saturation value K; the first entry, res[[1]], is the result when K is discrete (0 if discrete=FALSE) and the second entry, res[[2]], is the result when K is continuous (0 if continuous=FALSE). \$par is the maximum likelihood estimate of (lambda,mu,K). \$value is the -log likelihood value. The likelihood is calculated assuming there were two lineages at the time of the root. The likelihood is NOT conditioned on survival of the two lineages. Likelihood values from bd.shifts.optim are directly comparable (eg. using AIC) for survival = 0. Likelihood values from laser are directly comparable to those obtained by bd.densdep.optim and bd.shifts.optim for survival = 0 after the TreePar output \$value is transformed to -\$value+sum(log(2:length(x))).
-----	---

### Note

A faster version is now implemented in R package expoTree by Gabriel Leventhal. It is equivalent to the method here, see examples for function LikDD. Further, it can handle trees with sequentially sampled tips. bd.densdep.optim(x,Yule=TRUE,discrete=FALSE,continuous=TRUE) in TreePar and DDL(x) in Laser return the same results (up to transforming the -log likelihood (\$value) from TreePar via -\$value+sum(log(2:length(x))).

### Author(s)

Tanja Stadler, Gabriel Leventhal

## References

R.S. Etienne, B. Haegeman, T. Stadler, T. Aze, P.N. Pearson, A. Purvis, A.B. Phillimore. Diversity-dependence brings molecular phylogenies closer to agreement with the fossil record. *Proc. Roy. Soc. B*, 279: 1300-1309, 2012.

G. Leventhal, H. Guenthard, S. Bonhoeffer, T. Stadler. Using an epidemiological model for phylogenetic inference reveals density-dependence in HIV transmission. *Mol. Biol. Evol.*, 31(1): 6-17, 2014.

## Examples

```
set.seed(1)
x<-c(10:1)

bd.densdep.optim(x,discrete=FALSE,continuous=TRUE)

# Laser returns same result for Yule model
res <- -bd.densdep.optim(x,Yule=TRUE,discrete=FALSE,continuous=TRUE)[[2]]$value
res<-res+ sum(log(2:length(x)))
res

# library(laser)
# DDL(x)
```

---

bd.shifts.optim	<i>bd.shifts.optim: Estimating speciation and extinction rate changes and mass extinction events in phylogenies.</i>
-----------------	--

---

## Description

bd.shifts.optim estimates the maximum likelihood speciation and extinction rates together with the rate shift times  $t=(t[1],t[2] \dots, t[m])$  in a (possibly incomplete sampled) phylogeny. At the times  $t$ , the rates are allowed to change and the species may undergo a mass extinction event.

## Usage

```
bd.shifts.optim(x, sampling, grid, start, end, maxitk = 5, yule = FALSE, ME = FALSE,
all = FALSE, posdiv = FALSE, miniall = c(0), survival = 1,groups=0)
```

## Arguments

x	Vector of speciation times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using <code>getx(TREE)</code> .
sampling	Vector of length m. <code>sampling[i]</code> is the probability of a species surviving the mass extinction at time $t[i]$ . <code>sampling[1]</code> is the probability of an extant species being sampled. <code>sampling[1]=1</code> means that the considered phylogeny is complete. <code>sampling[i]=1</code> ( $i>1$ ) means that at time $t[i]$ , a rate shift may occur but no



species go extinct. If ME=TRUE, all entries but sampling[1] will be discarded as they are estimated. However, you have to input a vector sampling of the appropriate length such that the program knows how many mass extinction events you want to allow for.

grid, start, end	The model parameters are optimized for different fixed rate shift times. The fixed rate shift times are specified by being at (start, start+grid, start+2*grid .. end). I calculate the likelihood for the different rate shift times t instead of optimizing t with the function optim used for the other parameters, as the optimization performed poor for t (namely getting stuck in local optima).
yule	yule=TRUE sets the extinction rates to zero.
maxitk	Integer value defining how many iterations shall be done in the optimization. Default is 5, but needs to be increased if too many warnings "convergence problem" appear.
ME	ME=FALSE (default) uses the mass extinction fractions specified in sampling and does not estimate them. If ME=FALSE is used with sampling=c(1,1, .. , 1), no mass extinction events are considered.
all	Only relevant when ME=TRUE. all=FALSE (default and recommended) estimates one speciation and one extinction rate for the whole tree, and estimates the intensities sampling[i] (i>1) of mass extinction events. all=TRUE allows for varying speciation and extinction rates. Since the parameters might correlate, all=TRUE is not recommended.
posdiv	posdiv=FALSE (default) allows the (speciation - extinction) rate to be negative, i.e. allows for periods of declining diversity. posdiv=TRUE forces the (speciation - extinction) rate to be positive.
miniall	If you have run the bd.shifts.optim for k shifts, but you now want to have K>k shifts, then set for the subsequent analysis the following: update sampling, and set miniall=res[[2]] where res[[2]] is the output from the run with k shifts.
survival	If survival = 1, the likelihood is conditioned on survival of the process (recommended). Otherwise survival = 0.
groups	If groups != 0, the first column of groups indicates the age of higher taxa and the second column the number of species in the higher taxa (each row in groups corresponds to a leaf in the tree).

### Value

res[[1]][[i]]	List of maximum likelihood parameter estimates for each fixed t (t determined by start, end, grid) where i-1 shifts are allowed to occur (i in 1:m). The first i entries are the turnover (extinction/speciation) estimates, for the successive intervals going back in time. The next i entries are the diversification rate estimates (speciation-extinction). The next i-1 entries are the probabilities that the lineage survives the mass extinction event (if ME=TRUE). (Note: if ME=TRUE and all=FALSE, the first entry is the turnover, the second the diversification rate, followed by the mass extinction survival probability).
res[[2]][[i]]	Maximum likelihood parameter estimates for i-1 shifts (i in 1:m). First entry is the (-log likelihood) value. The next entries are the maximum likelihood parameter estimates (see res[[1]][[i]]). The last i-1 entries are the shift times.

res[[3]]            Vector of time points where the function was evaluated.  
 res[[4]]            Array specifying the time points when there was a convergence problem: a row of res[[4]] with entry (i,t[i]) means that when adding the i-th shift at time t[i], a convergence problem was encountered.

### Note

The likelihood is calculated assuming there were two lineages at the time of the root. The likelihood is conditioned on survival of the two lineages if survival = 1. Likelihood values from bd.densdep.optim are directly comparable (eg. using AIC) for survival = 0. Likelihood values from package laser are comparable for survival = 0 after the TreePar output \$value is transformed to  $-$value + \sum(\log(2:\text{length}(x)))$ .

### Author(s)

Tanja Stadler

### References

- T. Stadler. Mammalian phylogeny reveals recent diversification rate shifts. Proc. Nat. Acad. Sci., 108(15): 6187-6192, 2011.
- T. Stadler, F. Bokma. Estimating speciation and extinction rates for phylogenies of higher taxa. Syst. Biol., 62(2): 220-230, 2013. (for groups>0 but no rate shift).
- A. Lambert, T. Stadler. Macro-evolutionary models and coalescent point processes: the shape and probability of reconstructed phylogenies. Theo. Pop. Biol., 90: 113-128, 2013. (for groups>0).

### Examples

```
set.seed(1)

# First we simulate a tree, and then estimate the parameters for the tree:
# Number of species
nspecies <- 20
# At time 1 and 2 in the past, we have a rate shift:
time <- c(0,1,2)
# Mass extinction intensities 0.5 at time 1 in past, 0.4 at time 2 in past.
# Present day species are all sampled (rho[1]=1):
rho <- c(1,0.5,0.4)
# speciation rates (between t[i],t[i+1] we have speciation rate lambda[i]):
lambda <- c(2,2,1)
# extinction rates (between t[i],t[i+1] we have extinction rate mu[i]):
mu <- c(1,1,0)
# Simulation of a tree:
tree<-sim.rateshift.taxa(nspecies,1,lambda,mu,frac=rho,times=time,complete=FALSE)
# Extracting the speciation times x:
x<-sort(getx(tree[[1]]),decreasing=TRUE)

# When estimating the the rate shift times t based on branching times x,
# we allow the shift times to be 0.6, 0.8, 1, 1.2, .. ,2.4:
start <- 0.6
```

```

end <- 2.4
grid <- 0.2

# We fix rho and estimate time, lambda, mu:
res <- bd.shifts.optim(x,rho,grid,start,end)[[2]]
res
# res[[2]] tells us about the maximum likelihood estimate given one rate shift:
# - log lik = 17.330862988.
# rate shift at time 2.2.
# turnover (extinction/speciation) = 0.186301549 more recent than 2.2,
#   and = 0.939681843 more ancestral than 2.2.
# net diversification (speciation-extinction) rate = 0.958947381 more recent than 2.2,
#   and = 0.000100009 more ancestral than 2.2.

#test if i shifts explain the tree significantly better than i-1 shifts, here i=1:
i<-1
test<-pchisq(2*(res[[i]][1]-res[[i+1]][1]),3)
#if test>0.95 then i shifts is significantly better than i-1 shifts at a 5% error

# We fix rho=1 and mu=0 and then estimate time, lambda:
resyule <- bd.shifts.optim(x,rho,grid,start,end,yule=TRUE)
resyule[[2]]
# We estimate time, rho, lambda, mu:
resrho <- bd.shifts.optim(x,rho,grid,start,end,ME=TRUE)
resrho[[2]]
# Data analysis in Stadler & Bokma, 2012:
# Number of species in each order from Sibley and Monroe (1990)
data(bird.orders)
S <- c(10, 47, 69, 214, 161, 17, 355, 51, 56, 10, 39, 152, 6, 143,
358, 103, 319, 23, 291, 313, 196, 1027, 5712)
groups<-get.groups(bird.orders,S,0)
groupscut<-get.groups(bird.orders,S,96.43)
x<-branching.times(bird.orders)
# transforming molecular timescale into calendar timescale
groups[,1]<-groups[,1]/0.207407
x<-x/0.207407
bd.shifts.optim(x,sampling=c(1),survival=1,groups=groups)[[2]]
bd.shifts.optim(x,sampling=c(1),survival=1,groups=groupscut)[[2]]
# allowing one shift in rates:
bd.shifts.optim(x,sampling=c(1,1),grid=1,start=20,end=25,survival=1,groups=groupscut)[[2]]

```

---

bd.shifts.plot

*bd.shifts.plot: Plots the diversification rate estimates obtained with the function bd.shifts.optim.*


---

## Description

bd.shifts.plot plots the diversification rate estimates obtained with the function bd.shifts.optim.

**Usage**

```
bd.shifts.plot(resall, shifts, timemax=100, ratemin=-1, ratemax=1, plotturnover=FALSE)
```

**Arguments**

resall	When k trees were analyzed, a list of length k with entries being the component [[2]] of the output from bd.shifts.optim for each tree.
shifts	resall contains the maximum likelihood parameter estimates for 0...m shifts. shifts specifies for how many shifts you want to plot the estimated diversification rates (you can determine the number of significant shifts using a likelihood ratio test).
timemax	Specifies the upper end of the x-axis (time in past). Lower end is always 0.
ratemin, ratemax	Specifies the upper and lower end of the y-axis (diversification rates).
plotturnover	The net diversification (speciation-extinction) rate is plotted. If plotturnover=TRUE, also turnover=extinction/speciation is plotted.

**Author(s)**

Tanja Stadler

**References**

T. Stadler. Mammalian phylogeny reveals recent diversification rate shifts. Proc. Nat. Acad. Sci., 108(15): 6187-6192, 2011.

**Examples**

```
set.seed(1)

# First we simulate a tree, and then estimate the parameters for the tree:
# Number of species
nspecies <- 20
# At time 1 and 2 in the past, we have a rate shift:
time <- c(0,1,2)
# Mass extinction intensities 0.5 at time 1 in past, 0.4 at time 2 in past.
# Present day species are all sampled (rho[1]=1):
rho <- c(1,0.5,0.4)
# speciation rates (between t[i],t[i+1] we have speciation rate lambda[i]):
lambda <- c(2,2,1)
# extinction rates (between t[i],t[i+1] we have extinction rate mu[i]):
mu <- c(1,1,0)
# Simulation of a tree:
tree<-sim.rateshift.taxa(nspecies,1,lambda,mu,frac=rho,times=time,complete=FALSE)
# Extracting the speciation times x:
x<-sort(getx(tree[[1]]),decreasing=TRUE)

# When estimating the shift times t for x, we allow the shift times to be 0.6, 0.8, 1, 1.2, .. ,2.4:
start <- 0.6
end <- 2.4
```

```

grid <- 0.2

# We fix rho and estimate time, lambda, mu:
res <- bd.shifts.optim(x,rho,grid,start,end)
res[[2]]
# We plot the result for 2 shifts:
bd.shifts.plot(list(res[[2]]),2,3,0,2)

```

---

bdsky.stt.optim	<i>bdsky.stt.optim: Estimating piecewise constant birth and death rates in phylogenies with sequentially sampled tips.</i>
-----------------	--

---

### Description

bdsky.stt.optim estimates the maximum likelihood birth and death rates together with the rate shift times  $t=(t[1],t[2] \dots, t[m])$  for a given phylogenetic tree with sequentially sampled tips. At the times  $t$ , the rates are allowed to change.

### Usage

```
bdsky.stt.optim(x, ttype=0, rho=0, sampprob=c(0), constdeath=0, root=0)
```

### Arguments

x	Vector of branching and sampling times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using <code>getx(TREE,sersampling=TRUE)</code> .
ttype	Vector of same length as x. If <code>ttype[i]=0</code> , then <code>x[i]</code> denotes a branching event. If <code>ttype[i]=1</code> , then <code>x[i]</code> denotes a sampling event.
rho	Probability of sampling individuals at present. <code>rho=0</code> is default.
sampprob	Vector of length k where k is the number of different birth rates to be estimated.
constdeath	If <code>constdeath=0</code> (default) then k death rates are estimated. If <code>constdeath=1</code> , then 1 death rate is estimated.
root	<code>root=0</code> indicates that there is an edge above the root (mrca) in the tree. <code>root=1</code> indicates that there is no edge above the root.

### Value

out[[1]]	Entry <code>[[j]]</code> are the maximum likelihood parameter estimates for $j-1$ shifts, $j=1\dots k$ . The first entry of <code>out[[1]][[j]]</code> is the $-\log$ likelihood value, followed by the maximum likelihood parameter estimates. The parameters are stated in the following order: first the $j$ turnover estimates (from recent to ancient), then the $j$ diversification (speciation-extinction) rate estimates (from recent to ancient), then the $j-1$ shift times.
----------	--

out[[2]] Matrix where in each row, the first entry denotes the type of convergence problem, the second entry denotes the number of shifts in the problematic calculation, and the third entry denotes in which interval it happened.

### Note

bdsky.stt.optim extends the function bd.shifts.optim to trees with sequentially sampled tips.

### Author(s)

Tanja Stadler

### References

T. Stadler, D. Kuehnert, S. Bonhoeffer, A. Drummond. Birth-death skyline plot reveals temporal changes of epidemic spread in HIV and hepatitis C virus (HCV). Proc. Nat. Acad. Sci., 110(1): 228-233, 2013.

### Examples

```
set.seed(1)

# simulation of a tree with one rate shift at 0.5:
lambda<-c(3,4)
mu<-c(1,1)
sampprob<-c(0.5,0.5)
time<-c(0,0.5)
n<-10
tree<- sim.bdsky.stt(n,lambda,mu,time,sampprob)
tree2<-addroot(tree[[1]],tree[[1]]$root.edge)
summary<-getx(tree2,sersampling=TRUE)
times<-summary[,1]
ttype<-summary[,2]

# Maximum likelihood parameter estimation:
out <- bdsky.stt.optim(x=times,ttype=ttype,sampprob=sampprob,root=0)
```

---

create.mat

*create.mat: Generates input for LikAge and bd.age.optim.matlab.*

---

### Description

create.mat generates input for LikAge and bd.age.optim.matlab.

### Usage

```
create.mat(x, numgridpts=500, path, matfilename="setup")
```

**Arguments**

x	Vector of speciation times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using <code>getx(TREE)</code> .
numgridpts	Number of gridpoints to be used in the numerical likelihood calculation in <code>LikAge</code> or <code>bd.age.optim.matlab</code> .
path	Path where your MCR (Matlab Compiler Runtime) is installed (for details see <code>?LikAge</code> ).
matfilename	File <code>matfilename.mat</code> is created which contains input (speciation times) in a valid format for <code>LikAge</code> and <code>bd.age.optim.matlab</code> .

**Value**

res	0 is returned. The file for downstream analysis is saved as <code>matfilename.mat</code> .
-----	--

**Note**

This function relies on compiled Matlab applications that need to be downloaded. For setting up the environment to run the function, please refer to 'Notes' in function 'LikAge'. This function is useful to evaluate if `LikAge` or `bd.age.optim.matlab` are called several times for one tree. Then the input for the matlab functions is created once and used in `LikAge` or `bd.age.optim.matlab` through `inputformat=1` and specifying `matfilename`.

**Author(s)**

Helen Alexander, Tanja Stadler

**References**

H. Alexander, A. Lambert, T. Stadler. Quantifying Age-Dependent Extinction from Species Phylogenies. Submitted.

**Examples**

```
## You need MCR to run the example
## directory where MCR is installed
# path<-"Applications/MATLAB_R2014a_node.app/"
## location of folder TreePar_Matlab containing compiled Matlab applications (see ?LikAge)
# locationMat<-"Users/tstadler/Documents/Data/Uni/Research/R/TreeParProject/TreePar_Matlab"
##
# x<-1:20
# setwd(paste(locationMat,"/compiled",sep=""))
# out<-create.mat(x,path=path)
```

---

<code>get.groups</code>	<i>get.groups: Generates input for <code>bd.shifts.optim</code> when <code>groups!=0</code>.</i>
-------------------------	--

---

**Description**

`get.groups` generates input for `bd.shifts.optim` if the phylogeny is not resolved on the species level (`groups!=0`).

**Usage**

```
get.groups(tree,S,xcut=0)
```

**Arguments**

<code>tree</code>	Phylogenetic tree to be analyzed with <code>bd.shifts.optim</code> .
<code>S</code>	<code>S[i]</code> : Number of species in tree represented by leaf <code>i</code> .
<code>xcut</code>	Age of the higher taxa. If <code>xcut=0</code> : age of higher taxa is the length of the edge corresponding to the higher taxa. If <code>xcut&gt;0</code> : age of each higher taxa is <code>xcut</code> .

**Author(s)**

Tanja Stadler

**References**

T. Stadler, F. Bokma. Estimating speciation and extinction rates for phylogenies of higher taxa. *Syst. Biol.*, 62(2): 220-230, 2013.

**Examples**

```
# see manual of bd.shifts.optim()
```

---

LikAge	<i>LikAge: Calculates the likelihood of speciation and extinction rates for an ultrametric phylogeny under an age-dependent extinction model conditioning on the age of the tree.</i>
--------	---

---

**Description**

LikAge calculates the likelihood of speciation and extinction rates for an ultrametric phylogeny under an age-dependent extinction model conditioning on the age of the tree. Speciation rate is constant. Time to extinction is a Gamma distribution.



**Usage**

```
LikAge(x, lambda, k, theta, sampling, root=1, inputformat=0,
precision=4, numgridpts=500, path, matfilename="setup")
```

**Arguments**

x	Vector of speciation times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using <code>getx(TREE)</code> .
lambda	Speciation rate.
k	Shape parameter of the Gamma distribution modelling time to extinction.
theta	Scale parameter of the Gamma distribution modelling time to extinction.
sampling	Probability to sample a present day lineage.
root	If <code>root=1</code> , <code>max(x)</code> is the crown age and the likelihood is conditioned on the crown age. If <code>root=0</code> , <code>max(x)</code> is the stem age and the likelihood is conditioned on the stem age.
inputformat	If 0, then speciation time vector 'x' is supplied. If 1, then mat-file input (obtained by <code>create.mat</code> ) in 'matfilename' is supplied, and 'x' is not considered.
precision	Number of decimal places returned from the numerical likelihood calculation.
numgridpts	Number of gridpoints (evenly spaced time points between zero and the tree age, <code>max(x)</code> ) used in the numerical likelihood calculation. The higher this number, the more accurate the numerical results will be.
path	Path where your MCR (Matlab Compiler Runtime) is installed (for details see <code>?LikAge</code> ).
matfilename	Only relevant if <code>inputformat=1</code> ; then <code>matfilename="setup"</code> if input file is <code>setup.mat</code> .

**Value**

res	negative log likelihood of the parameters given the tree. The likelihood conditions on sampling of at least 1 extant tip (analog to <code>survival=1</code> in other likelihood functions of <code>TreePar</code> ).
-----	--

**Note**

You do not need Matlab software to use the compiled Matlab applications. First you need to download the Matlab Compiler Runtime (MCR), freely available from Mathworks at: <http://www.mathworks.com/products/compiler>. You should download the same version with which the applications were compiled: if you use the compiled code we provide, this is v8.3. Note the directory where you install MCR, it has to be inputted in `LikAge` as 'path'. When you install MCR, the Installer gives the message: "On the target computer, append the following to you `DYLD_LIBRARY_PATH` environment variable:..." then it gives a path name which should be copied and pasted before closing the installer. Second you have to download the folder `TreePar_Matlab` which is available on our github account: <https://github.com/cevo-public>. The compiled Matlab applications are in the subfolder `TreePar_Matlab/compiled` and should run on Mac OS version 10.7 and newer. If our compiled code does not run on your system, then you need to compile our Matlab code (m-files) in `TreePar_Matlab/source`

to produce applications compatible with your system. This requires a licensed copy of the Matlab software (see <http://www.mathworks.com/>). Open Matlab. Change to directory TreePar\_Matlab/source. Then execute the following commands to compile the m-files into stand-alone applications using the Matlab compiler MCC: » `mcc -m -R -nodisplay -R -nojvm EvalLFcn.m`; » `mcc -m -R -nodisplay -R -nojvm MaxLFcn.m`; » `mcc -m -R -nodisplay -R -nojvm ReadTreeFcn.m`. Copy the three resulting .sh and .app files to the folder TreePar\_Matlab/compiled. Note that if you have a licensed copy of Matlab and do not wish to use the R interface, the m-files we provide in TreePar\_Matlab/source can also be used directly in Matlab for likelihood computations.

### Author(s)

Helen Alexander, Tanja Stadler

### References

H. Alexander, A. Lambert, T. Stadler. Quantifying Age-Dependent Extinction from Species Phylogenies. Submitted.

### Examples

```
## You need MCR to run the example
## directory where MCR is installed
# path<-"/Applications/MATLAB_R2014a_node.app/"
## location of folder TreePar_Matlab
# locationMat<-"/Users/tstadler/Documents/Data/Uni/Research/R/TreeParProject/TreePar_Matlab"
#
# x<-1:20
# numgridpts<-500
# lambda<-2
# k<-1
# theta<-1
# sampling<-1
## Provide path where the folder TreePar_Matlab is available
# setwd(paste(locationMat,"/compiled",sep=""))
# LikAge(x,lambda, k, theta, sampling, path=path)
## If the shape parameter is 1, we have a constant extinction rate,
## corresponding to the function LikShift. Then the following commands return the same result.
# LikAge(x,1, 1, 2, sampling, numgridpts=1000, path=path)
# LikShifts(x,c(0),c(1),c(1/2),c(1))
```

---

LikConstant

*LikConstant: Calculates the likelihood of constant birth and death rates for a given phylogenetic tree.*

---

### Description

LikConstant calculates the likelihood of constant speciation and extinction rates for a given phylogenetic tree, conditioning on the age of the tree.

**Usage**

```
LikConstant(lambda,mu,sampling,x,root=1,survival=1)
```

**Arguments**

lambda, mu	Speciation and extinction rate.
sampling	Sampling is the probability of an extant species being sampled and included into the tree.
x	Vector of speciation times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using getx(TREE).
root	If root = 1 then max(x) is the mrca (crown age), if root = 0 then max(x) is the age of a branch ancestral to the mrca (stem age).
survival	If survival = 1: The likelihood is conditioned on survival of the process (recommended). Otherwise survival = 0.

**Value**

res	-log likelihood of the birth and death rates given the tree.
-----	--

**Author(s)**

Tanja Stadler

**References**

T. Stadler. On incomplete sampling under birth-death models and connections to the sampling-based coalescent. *Jour. Theo. Biol.* 261: 58-66, 2009.

**Examples**

```
# see manual of LikShifts()
```

---

LikConstantn	<i>LikConstantn: Calculates the likelihood of constant birth and death rates for a given phylogenetic tree.</i>
--------------	---

---

**Description**

LikConstantn calculates the likelihood of constant birth and death rates for a given phylogenetic tree, conditioning on the age of the tree and the number of tips.

**Usage**

```
LikConstantn(lambda,mu,sampling,x,root=1)
```

**Arguments**

lambda, mu	Speciation and extinction rate.
sampling	Sampling is the probability of an extant species being sampled and included into the tree.
x	Vector of speciation times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using <code>getx(TREE)</code> .
root	If <code>root = 1</code> then <code>max(x)</code> is the mrca (crown age), if <code>root = 0</code> then <code>max(x)</code> is the age of a branch ancestral to the mrca (stem age).

**Value**

res	-log likelihood birth and death rate given the phylogeny.
-----	---

**Author(s)**

Tanja Stadler

**References**

T. Stadler. On incomplete sampling under birth-death models and connections to the sampling-based coalescent. *Jour. Theo. Biol.* 261: 58-66, 2009.

**Examples**

```
# see manual of LikShifts()
```

---

LikDD	<i>LikDD: Calculates the likelihood of speciation and extinction rates for an ultrametric phylogeny under a density-dependent speciation model conditioning on the age of the tree.</i>
-------	---

---

**Description**

LikDD calculates the likelihood of speciation and extinction rates for a phylogeny under a density-dependent speciation model. Speciation rate is a function of the number of species  $N$ ,  $\lambda(N) = \max(0, \lambda(1 - N/K))$ , where  $K$  is the carrying capacity and  $\lambda$  the speciation rate when  $N \ll K$ . Extinction rate is  $\mu$  (constant). For a computationally much faster implementation, please use function `runExpoTree` in R package `expoTree`. In contrast to LikDD, `runExpoTree` can handle trees with tips sampled sequentially through time.

**Usage**

```
LikDD(par, model=-1, x, Ndec=-1, minN=0, psi=0, sampling=1, root=0, ki=0, muset=0, vec=0)
```

**Arguments**

par	Parameters (lambda,mu,K) for which to calculate the likelihood: lambda is speciation rate, mu is extinction rate, and K is carrying capacity.
model	model = -1: speciation rate is $\lambda(N) = \max(0, \lambda(1-N/K))$ . model = 0: speciation rate is $\lambda(N) = \lambda$ for $N \leq K$ , and $\lambda(N) = 0$ for $N > K$ .
x	Vector of speciation times and ancient lineage sampling times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using <code>getx(TREE)</code> .
Ndec	Help variable when optimizing.
minN	Help variable when optimizing.
psi	Rate for sampling lineages through time.
sampling	Probability to sample a present day lineage.
root	If root = 1 then $\max(x)$ is the mrca (crown age), if root = 0 then $\max(x)$ is the age of a branch ancestral to the mrca (stem age).
ki	Help variable when optimizing.
muset	Help variable when optimizing.
vec	vec=0 returns likelihood of model parameters for a tree given 1 lineage at beginning. vec=1 returns vector of likelihoods given i lineages at beginning.

**Value**

res	-log likelihood of the parameters given the tree.
-----	---

**Note**

A faster version is now implemented in R package `expoTree` by Gabriel Leventhal. It is equivalent to the method here, see examples in `LikShifts()` manual. Further, `expoTree` can handle trees with sequentially sampled tips.

**Author(s)**

Tanja Stadler, Gabriel Leventhal

**References**

R.S. Etienne, B. Haegeman, T. Stadler, T. Aze, P.N. Pearson, A. Purvis, A.B. Phillimore. Diversity-dependence brings molecular phylogenies closer to agreement with the fossil record. *Proc. Roy. Soc. B*, 279: 1300-1309, 2012.

G. Leventhal, H. Guenthard, S. Bonhoeffer, T. Stadler. Using an epidemiological model for phylogenetic inference reveals density-dependence in HIV transmission. *Mol. Biol. Evol.*, 31(1): 6-17, 2014.

**Examples**

```
# see manual of LikShifts()
```

---

LikShifts	<i>LikShifts: Calculates the likelihood of time-dependent birth and death rates given a phylogenetic tree.</i>
-----------	--

---

### Description

LikShifts calculates the likelihood of speciation and extinction rates and shift times for a given phylogenetic tree, conditioning on the age of the tree.

### Usage

```
LikShifts(x, t, lambda, mu, sampling, posdiv=FALSE, survival=1, groups=0)
```

### Arguments

x	Vector of speciation times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using <code>getx(TREE)</code> .
t	Vector of length m specifying the time of rate shifts ( <code>t[1]=0</code> is required, being the present). An entry in t may not coincide with an entry in x.
lambda, mu	Vectors of the same length as t. <code>l[i]</code> (resp. <code>mu[i]</code> ) specifies the speciation (resp. extinction rate) prior to <code>t[i]</code> .
sampling	Vector of same length as t. <code>sampling[i]</code> is the probability of a species surviving the mass extinction at time <code>t[i]</code> . <code>sampling[1]</code> is the probability of an extant species being sampled. <code>sampling[1]=1</code> means that the considered phylogeny is complete. <code>sampling[i]=1</code> ( $i > 1$ ) means that at time <code>t[i]</code> , a rate shift may occur but no species go extinct.
posdiv	Not relevant when using LikShifts without optimizing (for <code>bd.shifts.optim</code> : <code>posdiv=FALSE</code> (default) allows the (speciation-extinction) rate to be negative, i.e. allows for periods of declining diversity. <code>posdiv=TRUE</code> forces the (speciation-extinction) rate to be positive).
survival	If <code>survival = 1</code> , the likelihood is conditioned on survival of the process (recommended). Otherwise <code>survival = 0</code> .
groups	If <code>groups != 0</code> : the first column of groups indicates the age of higher taxa and the second column the number of species in the higher taxa (each row in groups corresponds to a leaf in the tree).

### Value

res	-log likelihood of the model parameters for the given phylogenetic tree. The likelihood is calculated assuming there were two lineages at the time of the root. The likelihood is conditioned on survival of the two lineages if <code>survival = 1</code> . Likelihood values from <code>bd.densdep.optim</code> are directly comparable (eg. using AIC) for <code>survival = 0</code> . Likelihood values from <code>laser</code> are directly comparable to the <code>TreePar</code> output for <code>survival = 0</code> after the <code>TreePar</code> output <code>\$value</code> is transformed to <code>-\$value+sum(log(2:length(x)))</code> .
-----	---

**Author(s)**

Tanja Stadler

**References**

T. Stadler. Mammalian phylogeny reveals recent diversification rate shifts. Proc. Nat. Acad. Sci., 108(15): 6187-6192, 2011.

**Examples**

```

timevec<-c(0,0.15,0.25)
lambdavec<-c(2.5,2,3)
muvec<-c(0.5,0.7,0.6)
x<-c(0.3,0.19,0.1)
x1<-c(x,max(x)*1.1)
x2<-c(x,max(x))
sampling<-0.4
grouptime<- rep(min(x)*0.95,length(x)+1)
group<- cbind(grouptime,grouptime*0+1)
group2 <- group
group2[1,2] <- 4
group2[2,2] <- 5
group2[3,2] <- 3
group3<-group
group3[2,2]<-10

### calculate likelihoods with root = 1

## Shifts in speciation / extinction rates (Stadler, PNAS 2011; Smrckova & Stadler, Manuscript 2014)
for (survival in c(0,1)) {
print(LikShiftsPP(x,timevec,lambdavec,muvec,sampling,survival=survival))
print(LikShifts(x,timevec,lambdavec,muvec,c(sampling,1,1),survival= survival))
print(LikShifts(x,timevec,lambdavec,muvec,c(sampling,1,1),survival= survival,groups=group))
print(LikShiftsSTT(par=c(lambdavec,muvec,timevec[-1]),x,x*0+1,sprob=c(0,0,0),
sampling=c(sampling,0,0),survival=survival,root=1))
print(" ")
}

## Shifts in speciation / extinction rates with group sampling
for (survival in c(0,1)) {
print(LikShifts(x,timevec,lambdavec,muvec,c(sampling,1,1),survival= survival,groups=group2))
print(LikShifts(x,timevec,lambdavec,muvec,c(sampling,1,1),survival= survival,groups=group3))
print(" ")
}

## Constant speciation and extinction rates
# condition on age of tree x[1] and number of tips n
LikShiftsPP(x,timevec[1],lambdavec[1],muvec[1],sampling,n=1)
LikConstantn(lambdavec[1],muvec[1],sampling,x)
print(" ")
# condition on age of tree x[1]
for (survival in c(0,1)) {

```

```

print(LikConstant(lambdavec[1],muvec[1],sampling,x,root=1,survival=survival))
print(LikShiftsSTT(par=c(lambdavec[1],lambdavec[1],muvec[1],muvec[1],1),x,x*0+1,
sprob=c(0,0),sampling=c(sampling,1),survival=survival,root=1))
print(LikShiftsPP(x,c(0),lambdavec[1],muvec[1],sampling,root=1,survival=survival))
print(LikShifts(x,c(0),lambdavec[1],muvec[1],c(sampling),survival=survival))
print(LikShifts(x,c(0),lambdavec[1],muvec[1],c(sampling),survival= survival,groups=group))
print(LikShifts(x,c(0),c(lambdavec[1],lambdavec[1],lambdavec[1]),
c(muvec[1],muvec[1],muvec[1]),c(sampling,1,1),survival= survival))
  if (survival == 0 ) {
    print(LikDD(c(lambdavec[1],muvec[1], 200),
    model=0 ,root=1, x=sort(x),sampling=sampling)[1])  }
if (survival == 0 ) {
  print(LikDD(c(lambdavec[1],muvec[1], 300),
  model=-1 ,root=1, x=sort(x),sampling=sampling)[1])  }
print(" ")
}

## Diversity-dependent speciation rates
# condition on age of tree x[1], survival = 0
N<-10
pars <- matrix(c(N,lambdavec[1],muvec[1],0,sampling),nrow=1)

print(LikDD(c(lambdavec[1],muvec[1],N),x=sort(x),model=-1,root=1,sampling=sampling)[1])
## If you have the R package expoTree, you can compare:
# library(expoTree)
# print(-runExpoTree(pars,sort(x2),rep(1,length(x2)),survival=0)+(length(x)-1)*log(2))
print(" ")

### calculate likelihoods with root = 0

## Constant speciation and extinction rates
# condition on age of tree x[1] and number of tips n
print(LikShiftsPP(x,timevec[1],lambdavec[1],muvec[1],sampling,root=0,n=1))
print(LikConstantn(lambdavec[1],muvec[1],sampling,x,root=0))
print(" ")
# condition on age of tree x[1]
for (survival in c(0,1)){
print(LikShiftsPP(x,c(0),lambdavec[1],muvec[1],sampling,root=0,survival=survival))
print(LikConstant(lambdavec[1],muvec[1],sampling,x,root=0,survival=survival))
if (survival == 0 ) {print(LikDD(c(lambdavec[1],muvec[1], 200),
  model=0 ,root=0, x=sort(x),sampling=sampling)[1])  }
if (survival == 0 ) {print(LikDD(c(lambdavec[1],muvec[1], 300),
  model=-1 ,root=0, x=sort(x),sampling=sampling)[1])  }
print(" ")
}
## Diversity-dependent speciation rates
# condition on age of tree x[1], survival = 0
print(LikDD(c(lambdavec[1],muvec[1],N),x=sort(x),model=-1,root=0,sampling=sampling)[1])
## If you have the R package expoTree, you can compare:
# print(-runExpoTree(pars,sort(x),rep(1,length(x)),survival=0)+(length(x)-1)*log(2))

```



---

LikShiftsPP	<i>LikShiftsPP: Calculates the likelihood of time-dependent birth and death rates given a phylogenetic tree.</i>
-------------	--

---

### Description

LikShiftsPP calculates the likelihood of speciation and extinction rates and shift times given a phylogenetic tree, conditioning on the age of the tree. This function uses the point process theory (Lambert and Stadler, 2013).

### Usage

```
LikShiftsPP(x, t, lambda, mu, sampling, survival=1, root=1, n=0)
```

### Arguments

x	Vector of speciation times in the phylogeny. Time is measured increasing going into the past with the present being time 0. x can be obtained from a phylogenetic tree using <code>getx(TREE)</code> .
t	The time of rate shifts ( $t[1]=0$ is required, being the present). An entry in t may not coincide with an entry in x.
lambda, mu	Vectors of the same length as t. $\lambda[i]$ (resp. $\mu[i]$ ) specifies the speciation (resp. extinction rate) prior to $t[i]$ .
sampling	Sampling is the probability of an extant species being sampled and included into the tree.
survival	If <code>survival = 1</code> : The likelihood is conditioned on survival of the process (recommended). Otherwise <code>survival = 0</code> .
root	If <code>root = 1</code> then $\max(x)$ is the mrca (crown age), if <code>root = 0</code> then $\max(x)$ is the age of a branch ancestral to the mrca (stem age).
n	If $n \neq 0$ the likelihood is conditioned on n extant sampled tips and tree age $\max(x)$ .

### Value

res	-log likelihood of the model parameters given the phylogenetic tree.
-----	--

### Author(s)

Tanja Stadler

### References

T. Stadler. Mammalian phylogeny reveals recent diversification rate shifts. *Proc. Nat. Acad. Sci.*, 108(15): 6187-6192, 2011.

A. Lambert, T. Stadler. Macro-evolutionary models and coalescent point processes: the shape and probability of reconstructed phylogenies. *Theo. Pop. Biol.*, 90: 113-128, 2013.

**Examples**

```
# see manual of LikShifts()
```

---

LikShiftsSTT	<i>LikShiftsSTT: Calculates likelihood of piecewise constant birth and death rates for a given phylogenetic tree with sequentially sampled tips.</i>
--------------	--

---

**Description**

LikShiftsSTT calculates likelihood of piecewise constant birth and death rates for a given phylogenetic tree with sequentially sampled tips, conditioning on the age of the tree.

**Usage**

```
LikShiftsSTT(par, times, ttype, numbd=0, tconst=-1, sampling=0,
sprob, root=0, survival=1, tfixed=vector(), mint=0, maxt=0)
```

**Arguments**

par	First k entries are piecewise constant speciation rates backward in time, second k entries are piecewise constant extinction rates backward in time, the last k-1 entries are the times of the rate shift.
times	Vector of branching and sampling times in the phylogeny. Time is measured increasing going into the past with the present being time 0. times can be obtained from a phylogenetic tree using <code>getx(TREE, sersampling=TRUE)</code> . An time of rate shift specified in par may not coincide with an entry in times.
ttype	If <code>ttype[i]=0</code> , then <code>times[i]</code> denotes a branching event. If <code>ttype[i]=1</code> , then <code>times[i]</code> denotes a sampling event.
numbd	Help variable when optimizing.
tconst	Help variable when optimizing.
sampling	Probability of sampling individuals at present. <code>sampling=0</code> is default.
sprob	Vector of length k with the entries specifying the probability of a death event coinciding with sampling. In other words, <code>sprob</code> is the probability that an extinct node is sampled to appear in the observed phylogeny.
root	<code>root=0</code> indicates that there is an edge above the root (mrca) in the tree. <code>root=1</code> indicates that there is no edge above the root.
survival	If <code>survival = 1</code> , the likelihood is conditioned on survival of the process (recommended). Otherwise <code>survival = 0</code> .
tfixed	Help variable when optimizing.
mint	Help variable when optimizing.
maxt	Help variable when optimizing.

**Value**

res                    -log likelihood of the parameters for the given tree.

**Note**

LikShiftsSTT extends the function LikShifts to trees with sequentially sampled tips.

**Author(s)**

Tanja Stadler

**References**

T. Stadler, D. Kuehnert, S. Bonhoeffer, A. Drummond. Birth-death skyline plot reveals temporal changes of epidemic spread in HIV and hepatitis C virus (HCV). Proc. Nat. Acad. Sci., 110(1): 228-233, 2013.

**Examples**

```
#####
# Generating a tree
set.seed(1)
rootlength<-1
test0<-read.tree(text="((3:1.5,4:0.5):1,(1:2,2:1):3);")
test<-addroot(test0,rootlength)
test$Nnode<-4
test$states<-rep(1,4)
times<-getx(test,sersampling=1)[,1]
ttype<-getx(test,sersampling=1)[,2]
times0<-getx(test0,sersampling=1)[,1]
ttype0<-getx(test0,sersampling=1)[,2]
#####

# Likelihood calculation
# Tree with root edge
print(-LikShiftsSTT(c(2,1,0.8,0.5,1.5345),
times,ttype,sampling=0,sprob=c(1/3,2/3),survival=1,root=0) )
# Tree without root edge
print(-LikShiftsSTT(c(2,1,0.8,0.5,1.5345),
times0,ttype0,sampling=0,sprob=c(1/3,2/3),survival=1,root=1) )

#####
# This little example shows that in the case of constant rates,
# LikShiftsSTT and LikTypesSTT yield the same results.
# In LikShiftsSTT root=0 or 1 allowed. In LikTypesSTT only root=0 possible.
death2<-c(3/2)
sampprob2<-c(1/3)
lambda2<-c(2)
t2<-c(0)
par2<-c(lambda2,death2)
#collapse to 1 state
```

```

epsi<- 0
test$states<-rep(1,4)
root<-0
for (survival in c(0,1)) {
print(-LikShiftsSTT(par2,times,ttype,sampling=0,sprob=sampprob2,survival=survival,root=root) )
print(-LikShiftsSTT(c(lambda2,lambda2,death2,death2,(max(times)*0.2431)),
times,ttype,sampling=0,sprob=c(sampprob2,sampprob2),survival=survival,root=root) )
print(-LikShiftsSTT(par2,times,ttype,sampling=0,
sprob=c(sampprob2),root=root,survival=survival) )
print(-LikTypesSTT(c(lambda2,epsi,epsi,epsi,death2,epsi,0,0),test,
sampfrac=c(sampprob2,0),survival=survival,rtol=10e-10,atol=10e-10,freq=1,migr=0))
print(-LikTypesSTT(c(lambda2,epsi,epsi,epsi,death2,epsi,0,0),test,
sampfrac=c(sampprob2,0),survival=survival,rtol=10e-10,atol=10e-10,freq=1,migr=1))
print(" ")
}

```

---

LikTypesSTT

*LikTypesSTT: Calculates the likelihood of the 2-type birth-death model parameters for a given tree.*

---

## Description

LikTypesSTT calculates the likelihood of the 2-type birth-death model parameters given a tree, conditioning on the age of the tree. For obtaining the maximum likelihood parameter estimates use the R function `optim` (see example below).

## Usage

```

LikTypesSTT(par,phylo,fix=rbind(c(0,0),c(0,0)),sampfrac,
survival=0,posR=0,unknownStates=FALSE,rtol=1e-12,atol=1e-12,migr=0,freq=0,cutoff=10^12)

```

## Arguments

<code>par</code>	Parameters of the 2-type branching model in the order <code>lambda11</code> , <code>lambda12</code> , <code>lambda21</code> , <code>lambda22</code> , <code>death1</code> , <code>death2</code> , <code>gamma12</code> , <code>gamma21</code> . Currently only <code>gamma=0</code> is possible. Note that it is possible to only include a subset of these 8 parameters in <code>par</code> , the remaining ones are specified in <code>fix</code> (using <code>optim</code> , only the parameters specified in <code>par</code> will be estimated).
<code>phylo</code>	Phylogenetic tree for which the likelihood of the parameters is calculated.
<code>fix</code>	Determines which parameters are constrained when optimizing is performed. First row of <code>fix</code> specifies the parameters being constrained (1 for <code>lambda11</code> , 2 for <code>lambda12</code> etc). Second row of <code>fix</code> specifies the constrained parameters: (i) If entry <code>[2,j]</code> is non-negative, say <code>x</code> , then parameter <code>[1,j]</code> is fixed to <code>x</code> . (ii) If entry <code>[2,j]</code> is negative but not equal to <code>-0.4</code> , then parameter <code>[1,j]</code> is fixed to parameter <code>-m</code> times entry <code>[3,j]</code> (exception is <code>m=-0.4</code> : then the parameter <code>lambda22</code> is fixed to <code>lambda21*lambda12/lambda11</code> , used in Stadler & Bonhoeffer (2013) full reference below).

sampfrac	Vector of length 2. sampfrac[j] denotes the probability of sampling an individual in state j upon death (i.e. include the individual into the tree).
survival	survival=1 conditions the likelihood on sampling at least one tip. survival=0 default.
posR	posR=1 constrains the parameters (when optimizing) on the basic reproductive number $R_0 = >1$ . $R_0$ for two types is calculated using TreePar:::R0types. posR=0 default.
unknownStates	If unknownStates=FALSE (default), phylo\$states are used for the analysis. If unknownStates=TRUE, then the likelihood is calculated ignoring the tip states (used e.g. for identifying superspreader dynamics).
rtol	Relative tolerance parsed to the differential equation solver lsoda from package deSolve.
atol	Absolute tolerance parsed to the differential equation solver lsoda from package deSolve.
migr	If migr=0 then rate changes only at branching events (i.e. across-state transmission); if migr=1 then rate changes only along lineages (i.e. migration); if migr=2 then we assume a model with exposed state (1) and infectious state (2) where an infected individual transmits with rate $\lambda_{21}$ giving rise to a new exposed individual, and an exposed individual moves to the infectious class with rate $\lambda_{12}$ .
freq	Specifies the probability of the origin of the tree being of type 1. If freq=0 then the equilibrium frequency given by the parameters par is used (see Supplement in Stadler & Bonhoeffer (2013) for details).
cutoff	We assume that prior to time cutoff, the sampling probability is 0. This setting acknowledges no sampling effort prior to a certain time. Cutoff must be older than the most ancestral tip. Default $10^{12}$ means sampling throughout the whole time.

**Value**

out	-log probability density of the (oriented) tree given the parameters (i.e. - log likelihood of the parameters for a fixed tree).
-----	--

**Note**

This likelihood function extends the likelihood framework in the R package diversitree to trees with sequentially sampled tips. Our Ebola 2014 analyses highly relied on the function LikTypesSTT. Scripts used for this Ebola analysis are provided on our webpage [www.bsse.ethz.ch/cevo](http://www.bsse.ethz.ch/cevo).

**Author(s)**

Tanja Stadler

**References**

T. Stadler, S. Bonhoeffer. Uncovering epidemiological dynamics in heterogeneous host populations using phylogenetic methods. *Phil. Trans. Roy. Soc. B*, 368 (1614): 20120198, 2013.

**Examples**

```

set.seed(1)
lambda11<-15
lambda12<-3
lambda21<-1
lambda22<-3
death1<-4
death2<-4
sampprob1<-0.05
sampprob2<-0.05
l<-rbind(c(lambda11,lambda12),c(lambda21,lambda22))
d<-c(death1,death2)
s<-c(sampprob1,sampprob2)
n<-20
init<- -1

tree<-sim.bdtypes.stt.taxa(n,l,d,s,init)
tree<-addroot(tree,tree$root.edge)

# Calculate likelihood for lambda11=15,lambda12=lambda21=lambda22=mu1=mu2=2,gamma=0
LikTypesSTT(par=c(2,2,2,2),phylo=tree,
fix=rbind(c(1,6,7,8),c(15,-5,0,0),c(1,1,1,1)),samppfrac=s,survival=0,posR=0)
# Calculate maximum likelihood parameter estimates of lambda12,lambda21,
# lambda22,mu1 constraining lambda11=15,mu2=mu1 and gamma=0.
out<-try(optim(c(2,2,2,2),LikTypesSTT,phylo=tree,fix=rbind(c(1,6,7,8),c(15,-5,0,0),c(1,1,1,1)),
samppfrac=s,survival=0,posR=0,control=list(maxit=10000)))

### Likelihood calculation assuming a model with exposed class (migr = 2)
# Simulating a tree with exposed class
set.seed(2)
# simulate tree with expected incubation period of 14 days,
# infectious period of 7 days, and R0 of 1.5:
mu <- c(0,1/7)
lambda <- rbind(c(0,1/14),c(1.5/7,0))
# sampling probability of infectious individuals is 0.35:
sampprob <-c(0,0.35)
# we stop once we have 20 samples:
n <- 20
# we simulate one tree:
numbsim<-1
# We mark first eliminate=10 tips such that we can easily drop them later
# (if deleting these 10 tips, we mimic no sampling close to the outbreak)
trees<-lapply(rep(n,numbsim),sim.bdtypes.stt.taxa,lambda=vector=mu,deathvector=mu,
sampprobvector=sampprob,EI=TRUE,eliminate=10)
tree<-trees[[1]]
origin<-max(getx(tree,sersampling=1)[,1])+tree$root.edge
# delete first eliminate=10 tips:
droptip<-tree$tip.label[which(tree$states == 1)]
phylo<-drop.tip(tree,droptip)
br<-getx(phylo,sersampling=1)
# we only sample after cutoff time:

```

```

cutoff<-max(br[which(br[,2]==0),1])*1.01
# add time since origin:
phylo<-addroot(phylo,origin-max(br))
# all tips were infectious:
phylo$states<-rep(2,length(phylo$tip.label))
##
# Evaluate likelihood at parameters used for simulation:
LikTypesSTT(c(0,lambda[1,2],lambda[2,1],0,0,mu[2],0,0),phylo,
sampfrac=sampprob,migr=2,cutoff=cutoff,freq=1)

#####
#This little verifies the correctness of the implementation by permuting both states and rates

#####
test<-read.tree(text="((C:1.5,D:0.5):1,(A:1,B:1):3);")
test<-addroot(test,0.1)
par1<-2
par2<-1
par3<-0.5
par4<-3
par5<-1
par6<-0.5
par7<-1/3
par8<-0.5
#####

for (survival in c(0,1)) {
test$states<-c(2,1,2,1)
print(-LikTypesSTT(c(par4,par3,par2,par1,par6,par5,0,0), test,
sampfrac=c(par8,par7),survival=survival,rtol=10e-14,atol=10e-14,migr=0,freq=0.5))
test$states<-c(1,2,1,2)
print(-LikTypesSTT(c(par1,par2,par3,par4,par5,par6,0,0), test,
sampfrac=c(par7,par8),survival=survival,rtol=10e-14,atol=10e-14,migr=0,freq=0.5))
print(" ")
test$states<-c(2,1,2,1)
print(-LikTypesSTT(c(par4,par3,par2,par1,par6,par5,0,0), test,
sampfrac=c(par8,par7),survival=survival,rtol=10e-14,atol=10e-14,migr=1,freq=0.5))
test$states<-c(1,2,1,2)
print(-LikTypesSTT(c(par1,par2,par3,par4,par5,par6,0,0), test,
sampfrac=c(par7,par8),survival=survival,rtol=10e-14,atol=10e-14,migr=1,freq=0.5))
print(" ")
}

```

# Index

`addroot`, [3](#)

`bd.age.optim.matlab`, [4](#)

`bd.densdep.optim`, [6](#)

`bd.shifts.optim`, [8](#)

`bd.shifts.plot`, [11](#)

`bdsky.stt.optim`, [13](#)

`create.mat`, [14](#)

`get.groups`, [16](#)

`LikAge`, [16](#)

`LikConstant`, [18](#)

`LikConstantn`, [19](#)

`LikDD`, [20](#)

`LikShifts`, [22](#)

`LikShiftsPP`, [25](#)

`LikShiftsSTT`, [26](#)

`LikTypesSTT`, [28](#)

`TreePar` (`TreePar`-package), [2](#)

`TreePar`-package, [2](#)