

# Diversification

*Simon Joly*

*Fall 2019*

## Contents

Before starting: data and R packages . . . . .	1
<b>Introduction</b>	<b>1</b>
The birth-death model . . . . .	3
Tree simulation . . . . .	3
Yule model . . . . .	4
Effect on tree shape . . . . .	4
Estimating diversification rates . . . . .	5
Empirical data . . . . .	7
Practicals 1 . . . . .	8
Lineage through time plots . . . . .	8
Practicals 2 . . . . .	10
<b>Trait dependent diversification rates</b>	<b>12</b>
Bayesian BISSE analysis . . . . .	13
Other BISSE-like models . . . . .	18
The problem of pseudoreplication . . . . .	19
<b>Further reading</b>	<b>19</b>
<b>References</b>	<b>19</b>

## Before starting: data and R packages

To do the exercises of this tutorial you will need to have the following packages installed.

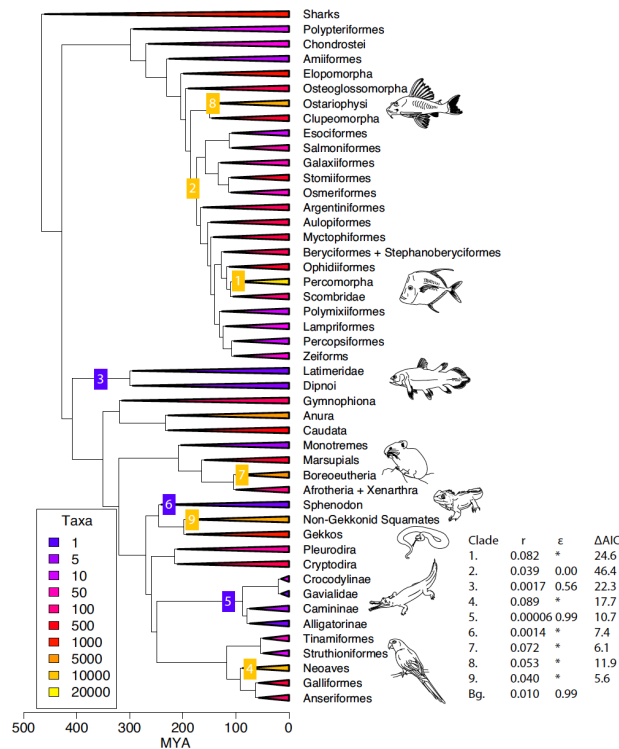
```
library(ape)
library(phytools)
library(diversitree)
library(lattice)
library(coda)
```

If some of these are not installed, install them using the `install.packages()` function.

You will also need some data sets. You can download them from the Github repository for this tutorial and place them in your working directory.

## Introduction

When we look at the tree of life or at a section of it, one of the first things we notice is that there is a great imbalance between different lineages. Some lineages are species-rich whereas others are species poor. The figure below shows the evolution of the main lineages of jawed vertebrates and clearly shows that while some lineages possess more than 1000 species, others have less than 10 (figure from Alfaro et al. 2009)!



This disparity in species number have attracted the interest of biologist for a long time. Evolutionary biologist have wanted to quantify these pattern and a key concept is that of diversification rates.

Diversification consists in the net accumulation of species through time. Conceptually, it is the difference between speciation and extinction. Mathematically, the speciation rate is normally represented by the greek character  $\lambda$  and extinction rate by the letter  $\mu$ . Diversification is thus:

$$Diversification = \lambda - \mu$$

If  $\lambda > \mu$ , the diversification rate is positive and the number of species in a group will increase through time. In contrast, if  $\lambda < \mu$ , the diversification rate is negative and species number are diminishing through time and a group characterized by negative diversification will eventually become extinct.

### What do the rates mean ?

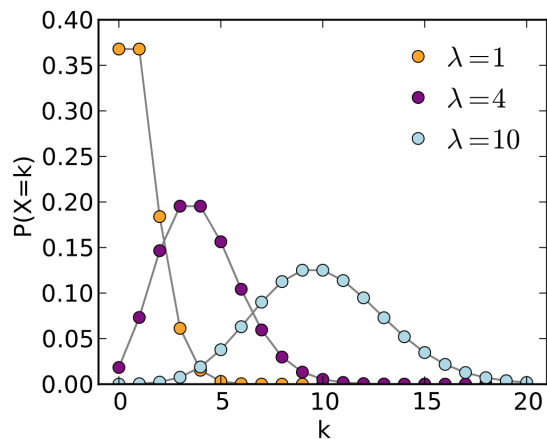
Rates can have different meaning in the field of biology. Within evolutionary biology, especially in the field of comparative methods, a rate is an attribute of a species or a lineage. For instance, the **speciation rate** is the expected number of speciation events expected to occur for a species (of along the branch of a phylogenetic tree) for a certain period of time. As such, a species could be expected to experience 1.3 speciation events per million of years on average.

Because it is an attribute of a lineage, this means that a given group could have much more frequent speciation events. For instance, a group of 10 species with a speciation rate of 1.3 will have an expected  $10 * 1.3 = 13$  speciation events per million years on average.

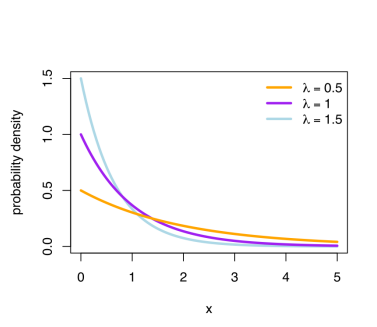
## The birth-death model

The birth death model is a continuous-time Markov process that describes how species fluctuate through time. It is a common model for phylogenetic trees and provide a way to simulate them.

Because we cannot have fraction of speciation (or extinction) event for a species for a given amount of time, we need a way to calculate the number of real events. For this, we use the Poisson distribution that describe the probability of observing different number of event given a specific rate and a given period of time.



This is useful, but even more useful when one wants to simulate a phylogenetic tree or speciation in a population is to estimate the waiting time to a next speciation event. We call this the waiting time. Because of the Poisson distribution, we know that the waiting time to a next event follows an exponential distribution with parameter  $\lambda$ . Formally,  $f(x; \lambda) = \lambda e^{-\lambda x}$ , where  $\lambda$  is the parameter of the distribution.



To get the distribution of waiting times to the next speciation event, we replace the parameter of the exponential distribuion ( $\lambda$ ) by the speciation rate (also  $\lambda$ ). Similarly, the time to the next extinction event is exponential with parameter  $\mu$  and the time to a speciation **or** an extinction event is exponential with parameter  $\lambda + \mu$ .

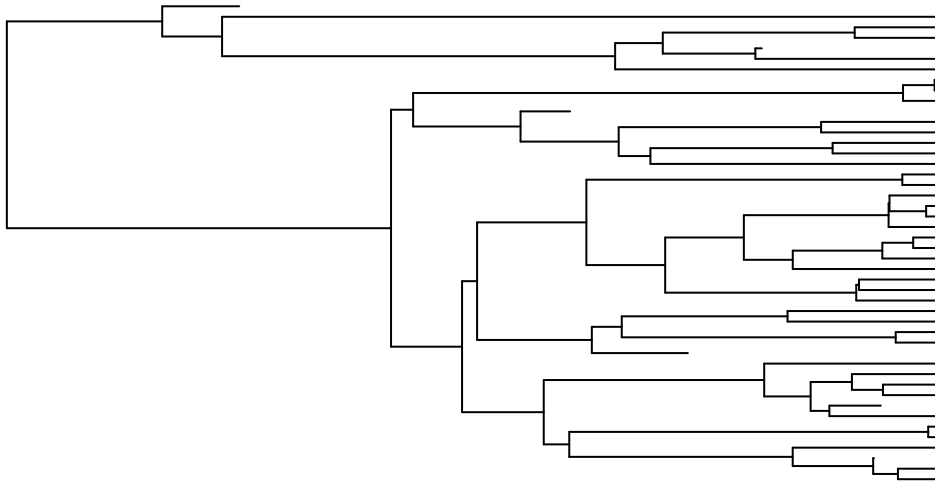
The distribution of waiting times depends on the rate and a greater rate results in shorter waiting time on average; the time for a next speciation event to occur will be shorter.

With this info, we could simulate trees!

## Tree simulation

Here is an example of how to simulate a tree in R. The function `sim.bdtree` simulates a tree with a birth death model with birth rate (speciation)  $b$ , death (extinction) rate  $d$ . The commands `stop="taxa"`, `n=40` indicate to stop the simulation when a number of species is reached, here 40.

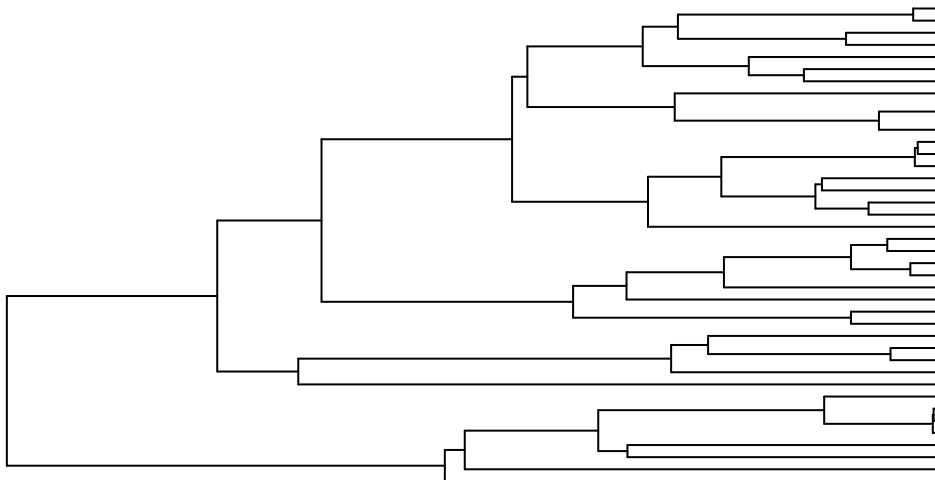
```
library(geiger)
atree <- sim.bdtree(b=0.2,d=0.05,stop="taxa",n=40,seed=46)
plot(atree,show.tip.label = F)
```



## Yule model

The Yule model is another model often used to simulate trees. It is also often called a “pure birth” model. It is a special case of the birth-death model with the extinction rate fixed to 0.

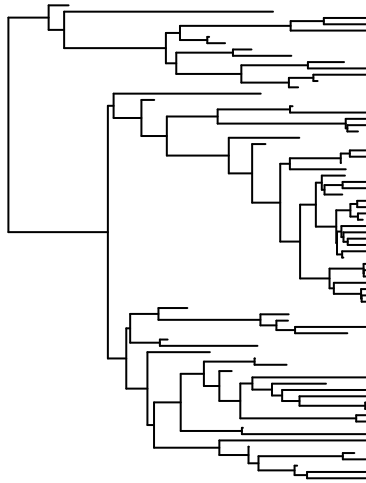
```
atree <- sim.bdtree(b=0.2, d=0, stop="taxa", n=40, seed=46)
plot(atree, show.tip.label = F)
```



## Effect on tree shape

Not suprisingly, the speciation and extinction rates have strong impacts on the shape of the tree. Here are two trees simulated under different speciation and extinction rates.

**lambda=0.2, mu=0.05**



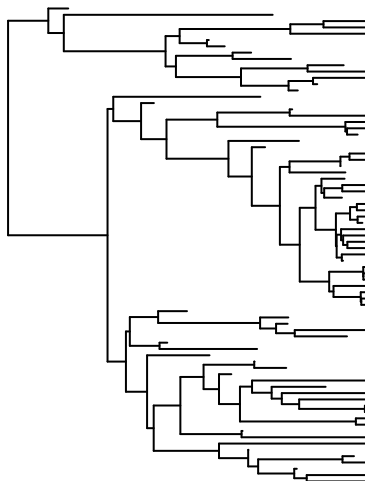
**lambda=0.6, mu=0.05**



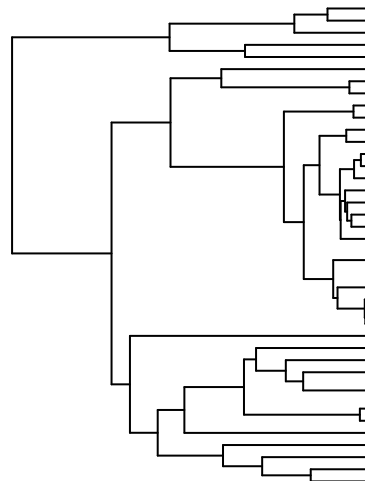
### Estimating diversification rates

One obvious application of these models is to estimate speciation, extinction and diversification rates. But this is a difficult task. Part of the difficulty is that by sampling current species, we do not have the complete information. In particular, we don't know if some species have gone extinct, when they went extinct, and where on the trees those extinct species diverged. Such information is important as it affects the estimation of waiting times and thus it might bias the estimation of speciation and extinction rates. Actually, extinction rates are especially difficult to estimate from phylogenetic tree.

#### Real tree



#### Observed data

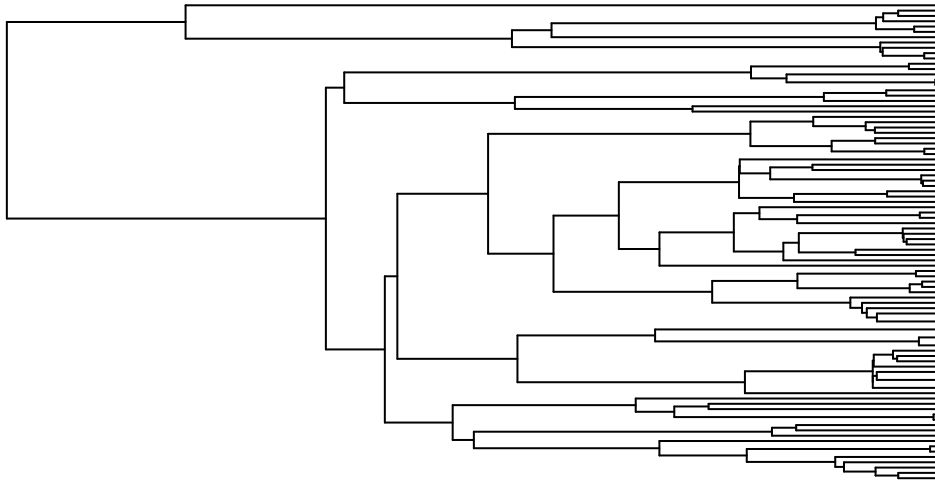


Another shortcoming occurs when we have not sampled all current (extant) species. For the same reason, this will strongly affect the estimates of speciation and extinction rates.

## An example

Let's look at an example to see how well we could infer rates from a phylogeny. For this, we will first simulate some data. We will simulate a tree of 100 species to see how well we could infer the rates we used for simulating the phylogeny. But first, we will exclude extinct species and we will simulate partial sampling by removing 10% of the species.

```
# Simulate a tree with birth-death process
tree1 <- sim.bdtree(b=0.2,d=0.05,stop="taxa",n=100,seed=46)
# Remove extinct species
tree1 <- drop.extinct(tree1)
# Sample only 90 of the living species. For this we will first sample without replacement 10
# species from the list of species simulated.
sp.not.sampled <- sample(tree1$tip.label,10,replace=FALSE)
# Now exclude those 10 "unsampled" species
tree1 <- drop.tip(tree1, sp.not.sampled)
# Look at the phylogenetic tree
plot(tree1, show.tip.label=FALSE)
```



We will now fit a birth-death model and a yule model on the tree. For this we will use the `diversitree` package.

```
require(diversitree)

## Fit a Yule model
# Make model
pbModel <- make.yule(tree1)
# Fit the model
pbMLFit <- find.mle(pbModel, 0.1)

## Fit a Birth-death model
# Make model
bdModel <- make.bd(tree1)
# Fit the model
bdMLFit <- find.mle(bdModel, c(0.1, 0.05), method = "optim", lower = 0)
```

Now that we have fit the models, we could compare them.

```
# compare models
anova(bdMLFit, pure.birth = pbMLFit)
```

```
##           Df  lnLik      AIC  ChiSq Pr(>|Chi|)
## full           2 78.972 -153.94
## pure.birth    1 77.557 -153.11 2.8288    0.09259 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This compares the full model (more complex) which here is the birth-death model because it has one more parameter. The AIC criteria suggests that the full model is better (AIC is smaller than for the Yule model), but only slightly so as a difference  $< 2$  in AIC is generally considered as not significant. The likelihood ratio test (last column;  $\text{Pr}(>|\chi|)$ ) also suggests the difference is not significant at the 5% level. This shows that with a tree of this size, using a small extinction rate (0.05), it is difficult to confirm that the data was simulated under a birth-death model.

Let's now look at the parameter estimates.

```
coef(bdMLFit)
```

```
##      lambda      mu
## 0.2485543 0.1176836
```

You can see that the estimates of the parameters are not very far from the simulated data (speciation = 0.2, extinction = 0.05). So the results are not so bad!

Finally, to estimate the diversification rate, you only have to subtract the extinction rate from the speciation rate.

```
# Diversification rate
speciation <- coef(bdMLFit)[[1]]
extinction <- coef(bdMLFit)[[2]]
(diversification <- speciation - extinction)
```

```
## [1] 0.1308707
```

## Empirical data

For the present tutorial, we will use data from Sánchez-Baracaldo and Thomas (2014) that studied a group of monophyletic ferns from the Jamesonia-Eriosorus Complex from the Andes. They studied mostly the impact of the colonization of the Paramo habitat in diversification rates and on morphology.

Let's read the data and prepare it for the analyses to come.

```
require(ape)
# Read tree
paramotree <- read.nexus("./JamesoniaPartFind_MCC.tre")
# Read data
paramodata <- read.csv("./JamesoniaTraits2014Analysis.csv")
# Remove species for which we don't have complete data
paramodata <- paramodata[!is.na(paramodata$Name_in_tree),]
# Remove species in the tree that are not in the data matrix
species.to.exclude <- paramotree$tip.label[!(paramotree$tip.label %in%
                                             paramodata$Name_in_tree)]
paramotree <- drop.tip(paramotree,species.to.exclude)
rm(species.to.exclude)
# Name the rows of the data.frame with the species codes used as tree labels
rownames(paramodata) <- paramodata$Name_in_tree
# Remove unnecessary variables
paramodata <- paramodata[,-c(1:3,7:14)]
# Order the data in the same order as the tip.label of the tree. In the
```

```
# present example, this was already the case.
paramodata <- paramodata[paramotree$tip.label,]
```

The tree is stored in the variable `paramotree` and the data in the variable `paramodata`.

## Practicals 1

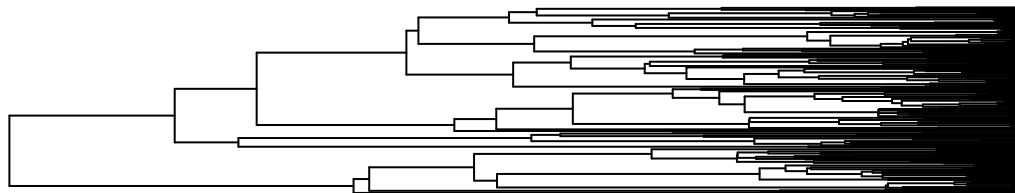
Using the approach used above, estimate the speciation, extinction and diversification rates for the *Jamesonia-Eriosorus* dataset.

## Lineage through time plots

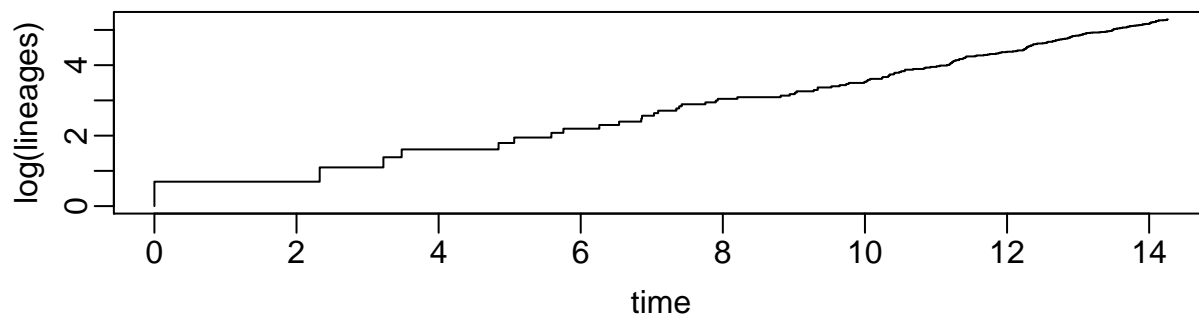
Another way to look at diversification is to look at the speed at which lineages have accumulated through time. According to a Yule model (i.e., no extinction), the species number should increase according to a semi-log scale through time. This is because each new species implies a new lineage from which new species can evolve, which will make new lineages, and so on. This means that if you plot the time in *x* with the log number of lineages in *y*, you will obtain a straight line. We call these plots lineages through time plots, or ltt plots. These can be obtained with the `ltt` function of the `phytools` package. Let's look at this with simulated data.

```
## Loading required package: maps
```

### Yule phylogeny (no extinction)



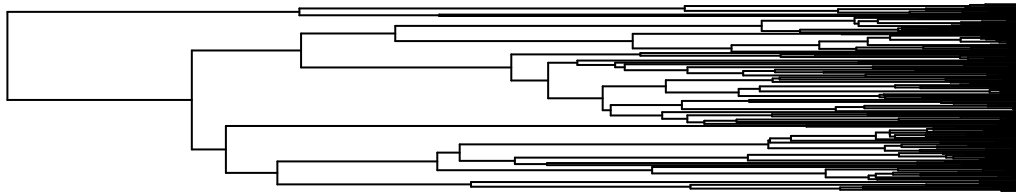
### Lineage through time (ltt) plot



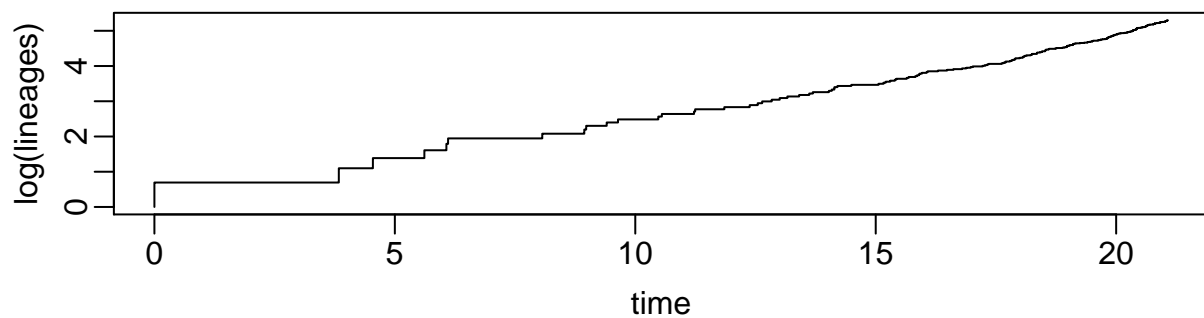
In contrast, with extinction, the expectations are different. Because recent lineages are less likely to go extinct than old ones, the plots often show a “pull to the present”, where the number of lineages raises more quickly near the present.



## Birth-death phylogeny



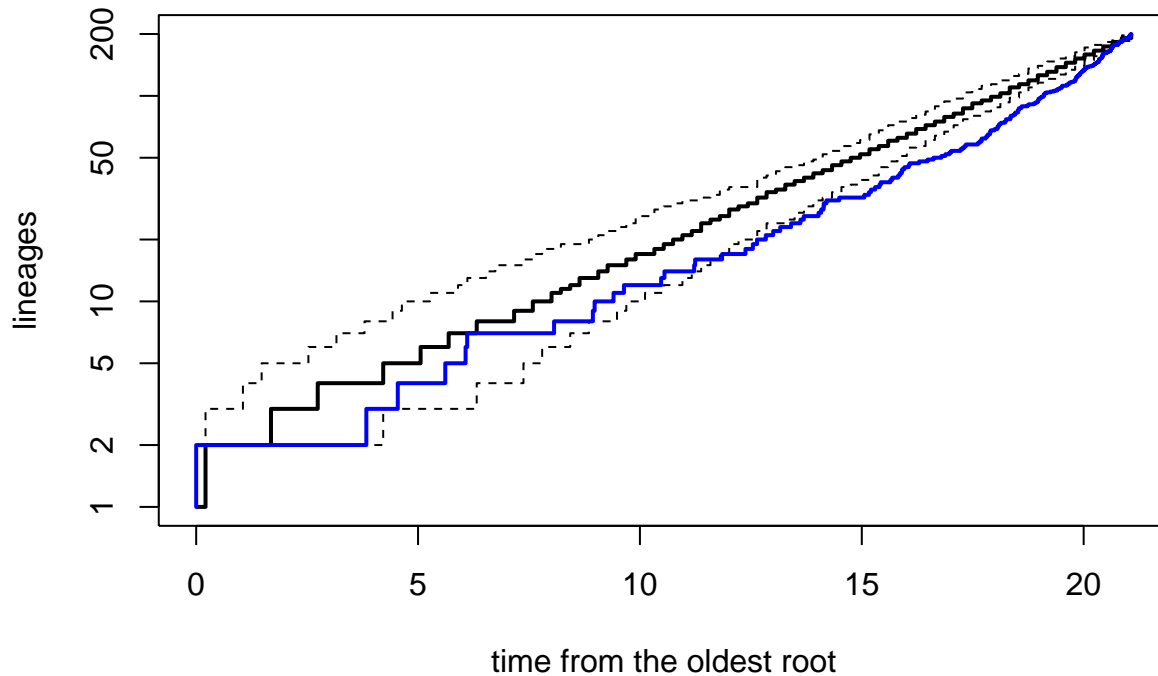
## Lineage through time (ltt) plot



### Testing models using ltt plots

ltt plots provides another way to test if a tree fits a give model, for instance a yule model. We could plot the ltt of the tree alongside of the yule expectation for a tree of the same size. Let's doo this

```
# Simulate a BD tree
bdtree <- sim.bdtree(b=0.4,d=0.2,stop="taxa",n=200,seed=46)
bdtree <- drop.extinct(bdtree)
# Calculate Pure-birth prediction from this tree
h<-max(nodeHeights(bdtree))
b<-(log(Ntip(bdtree))-log(2))/h
# Simulate 100 pure birth trees
trees<-pbtree(b=b,n=Ntip(bdtree),t=h,nsim=100,method="direct",quiet=TRUE)
# Plot ltt confidence intervals
ltt95(trees,gamma=FALSE,log=TRUE)
# Plot ltt from the initial data.
ltt(bdtree, add=TRUE, log.lineages=FALSE, col="blue", lwd=2)
```



```
## Object of class "ltt" containing:
##
## (1) A phylogenetic tree with 200 tips and 199 internal nodes.
##
## (2) Vectors containing the number of lineages (ltt) and branching times (times) on the tree.
##
## (3) A value for Pybus & Harvey's "gamma" statistic of 3.2206, p-value = 0.0013.
```

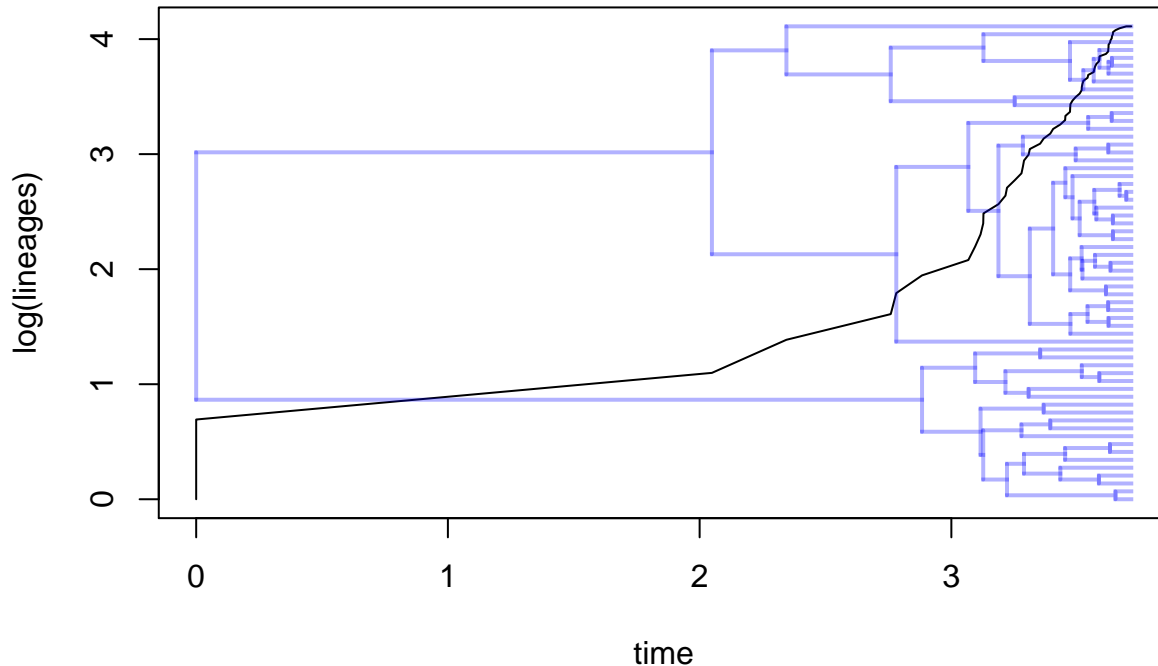
You can see that the ltt plot of the original tree (in blue) departs from what is expected under a Yule model, but not so much.

## Practicals 2

Estimate a ltt plot for the Jamesonia-Eriosorus dataset and compare it to trees simulated under a Yule model.

```
res<-ltt(paramotree, plot=FALSE)
plot(res,type="l",show.tree=TRUE,log.lineages=TRUE,main="LTT plot")
```

## LTT plot

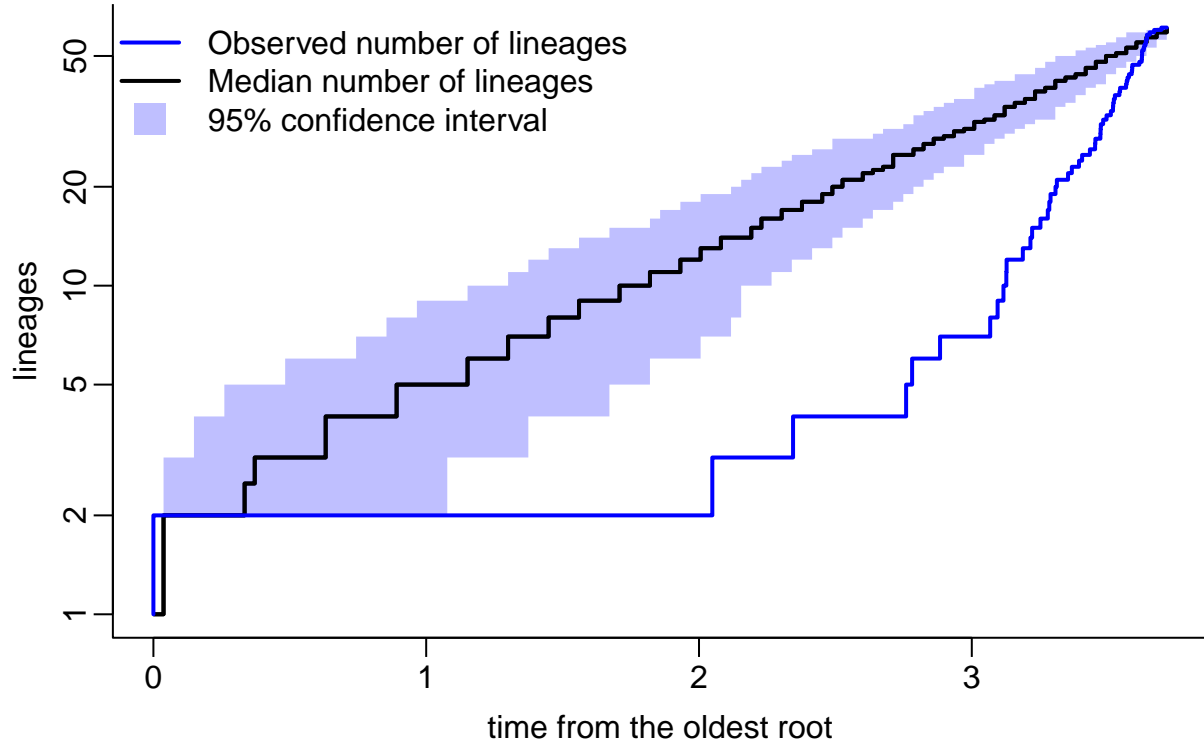


You can see that there are long branches at the base of the tree. Then there is a rapid increase and a slowdown as it reaches the present. Let's compare it to a Yule tree.

```
# Pure-birth prediction:
h<-max(nodeHeights(paramotree))
b<-(log(Ntip(paramotree))-log(2))/h
# Pure birth simulations
trees<-pbtree(b=b,n=Ntip(paramotree),t=h,nsim=100,method="direct",quiet=TRUE)
# Get ltt confidence intervals
object <- ltt95(trees,gamma=FALSE,log=TRUE,plot=FALSE)
# Plot
par(bty="l", mar=c(4,3,2,1), mgp=c(1.8,0.5,0))
# Plot confidence intervals with median
plot(object,shaded=TRUE)
# Plot ltt of observed data
ltt(paramotree, add=TRUE, xaxis="flipped", log.lineages=FALSE, col="blue", lwd=2)
```

```
## Object of class "ltt" containing:
##
## (1) A phylogenetic tree with 61 tips and 60 internal nodes.
##
## (2) Vectors containing the number of lineages (ltt) and branching times (times) on the tree.
##
## (3) A value for Pybus & Harvey's "gamma" statistic of 2.1495, p-value = 0.0316.
```

```
# Add legend
legend(x="topleft",c("Observed number of lineages","Median number of lineages", "95% confidence interval"),
      lty=c(1,1,0),lwd=c(2,2,0),pch=c(NA,NA,22),pt.cex=2.5,
      pt.bg=c("black","black",rgb(0,0,1,0.25)),col=c("blue","black","transparent"),bty="n")
```



## Trait dependent diversification rates

In the previous section, we saw how to estimate diversification rates for a whole tree. However, it is often of interest to estimate the diversification rate of species that have a special morphological characteristic or that are in a given environment. This is exactly what the BiSSE method does (Maddison et al. 2007). This method estimate diversification rates for states of a binary character.

In the BiSSE model, two processes are modeled. First the evolution of the character of interest. Second, the speciation and extinction rates for each state of the character. The idea is that according to the trait a species has, it will have distinct speciation or extinction rates.

The BiSSE model can have a up to 6 parameters:

- $\lambda_1$ ,  $\lambda_2$ , the speciation rates parameters for states 1 and 2
- $\mu_1$ ,  $\mu_2$ , the extinction rates parameters for states 1 and 2
- $q_{1 \rightarrow 2}$ ,  $q_{2 \rightarrow 1}$ , the transition rates from state 1 to 2 and from state 2 to 1, respectively

Note, however, that as with any model, it is possible to use simplified models, such as a model with equal transition rates between the two states ( $q_{1 \rightarrow 2} = q_{2 \rightarrow 1}$ ) or with equal extinction rates ( $\mu_1 = \mu_2$ ).

It is also possible to test specifically some hypotheses by using constrained models. For instance, you can test if the two character states are associated with the same speciation rate. To do this, you need to fit a model where the speciation rates are constrained to be the same for the two states of the character ( $\lambda_1 = \lambda_2$ ). Then, you can test if the more complex model fit the data significantly better than the simple model.

The BiSSE approach is implemented in the R package `diversitree`. We will apply it on the Paramo example to compare two models. The first model will be unconstrained and will evaluate the speciation and extinction rates for species that are either exposed or sheltered. This specific model was not tested in the original paper. They instead focused on diversification rates in species growing or not in the Paramo ecosystem. This model will be compared with a constrained model in which the speciation rates and the extinction rates will be

equal for the two states ( $\lambda_1 = \lambda_2$ ,  $\mu_1 = \mu_2$ ). This will allow to test whether exposed species have different speciation or extinction rates than sheltered species.

```
require(versitree)
# Convert the character into a numeric binary vector
char1 <- as.numeric(paramodata$ObservedMicrohabitat)
names(char1) <- row.names(paramodata)
# Create a BISSE like likelihood model
lik <- make.bisse(paramotree, char1)
# Create starting points for the search
p <- starting.point.bisse(paramotree)
# Perform a ML search
fit <- find.mle(lik, p, method="subplex")
# lnL of the model
logLik(fit)

## 'log Lik.' -30.25288 (df=6)

# The fitted parameters
round(coef(fit), 2)

## lambda0 lambda1      mu0      mu1      q01      q10
##      3.62      1.55      0.00      0.42      0.00      0.36

# Test a constrained model where the speciation rates and extinction rates
# are set equal
lik.l <- constrain(lik, lambda1 ~ lambda0, mu1 ~ mu0)
fit.l <- find.mle(lik.l, p[-c(1,3)], method="subplex")
logLik(fit.l)

## 'log Lik.' -33.75665 (df=4)

# Fitted parameters
round(coef(fit.l), 2)

## lambda1      mu1      q01      q10
##      3.53      2.68      0.01      0.44

# Test for statistical difference between the two models
anova(fit, equal.diver=fit.l)

##           Df  lnLik    AIC  ChiSq Pr(>|Chi|)
## full           6 -30.253 72.506
## equal.diver    4 -33.757 75.513 7.0075    0.03008 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

You can see that the more complex model is significantly better than the constrained model. This suggests that the diversification rates are different for the two groups of species.

## Bayesian BISSE analysis

It is also possible to perform the same analysis in a Bayesian framework.

The MCMC search needs tuning parameters, but these are hard to select *a priori*. The approach proposed by `versitree` is to start by running a basic MCMC search to then estimate the tuning parameters from the range of observed values observed in this first search. Here is how it works.

```

# Generate priors. Since there are relatively few species, this is
# important. We will use an exponential prior with rate 1/(2r),
# where r is the character independent diversification rate:
prior <- make.prior.exponential(0.1)

# Perform a first rapid search (with tuning set arbitrarily to 10)
# that will be used to estimate the tuning parameters. We will run
# it for 100 steps only, but this might have to be adjusted for more
# complex datasets.
tmp <- diversitree::mcmc(lik, p, nsteps=100, prior=prior, w=10, print.every=10)

## 10: {4.9441, 1.3027, 4.2664, 0.2709, 0.3587, 1.0216} -> -48.21728
## 20: {4.1444, 2.6456, 2.4307, 1.9158, 0.4225, 0.6509} -> -48.28575
## 30: {4.0428, 0.9812, 0.4197, 0.4786, 0.0417, 0.3518} -> -47.77474
## 40: {3.8320, 0.4828, 2.1335, 3.9868, 2.4332, 0.8137} -> -52.65975
## 50: {4.7667, 1.3118, 0.8592, 8.2244, 3.5312, 0.1969} -> -53.09431
## 60: {4.1169, 0.3579, 2.5143, 7.7144, 2.4441, 0.0140} -> -55.63579
## 70: {5.4418, 0.3445, 2.8193, 2.7238, 2.2310, 0.6984} -> -52.32689
## 80: {4.2073, 1.8022, 1.2878, 1.1617, 0.4169, 0.4675} -> -47.52367
## 90: {4.6461, 1.8556, 3.4612, 0.7778, 0.0297, 0.4829} -> -46.63297
## 100: {5.3883, 0.4489, 1.9689, 4.6793, 4.0488, 2.3820} -> -55.55654

# Then estimate the tuning parameters for the real analysis. For this
# we will remove the first ten generations.
w <- diff(sapply(tmp[-(1:10),2:7], range))

# Then run the true analysis. We will use 2000 steps here. Again, you
# may need to run the MCMC chain longer for large analyses.
samples <- diversitree::mcmc(lik, fit$par, nsteps=2000, prior=prior, w=w,
                             print.every=50)

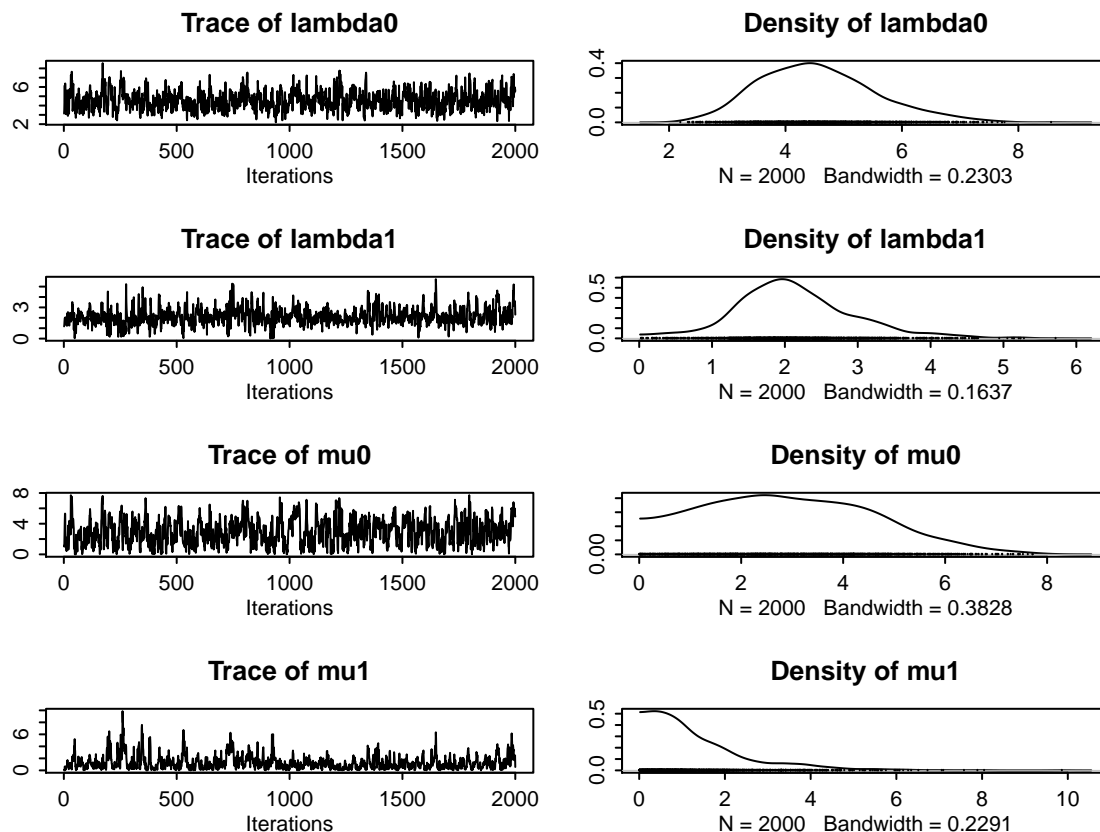
## 50: {3.4633, 0.8437, 0.4629, 3.3105, 2.1237, 0.9877} -> -52.00939
## 100: {4.9472, 2.0698, 1.2073, 1.2063, 0.1078, 0.2664} -> -47.13574
## 150: {4.0045, 2.3413, 3.2729, 0.6460, 0.5440, 1.2682} -> -48.68175
## 200: {6.7390, 1.0160, 4.5077, 2.6669, 1.9155, 1.6645} -> -55.06889
## 250: {6.7296, 0.6615, 4.4857, 0.4262, 1.1606, 1.1091} -> -54.01186
## 300: {3.9611, 1.0319, 1.3926, 0.2237, 0.2361, 0.9845} -> -49.37023
## 350: {4.4086, 3.9189, 2.5751, 6.0739, 1.5601, 0.2968} -> -53.15971
## 400: {3.5320, 1.1746, 0.7770, 0.2364, 0.4251, 0.3577} -> -47.30006
## 450: {6.5925, 1.8653, 6.2045, 0.5923, 0.0407, 0.9757} -> -47.86160
## 500: {3.4687, 1.8751, 1.6040, 0.8343, 0.7043, 0.9109} -> -48.43689
## 550: {4.6195, 2.0600, 4.3403, 1.1857, 0.4399, 0.6025} -> -49.26140
## 600: {5.6667, 2.5409, 4.5172, 0.5172, 0.0191, 1.2371} -> -48.46555
## 650: {5.5412, 2.1508, 5.6144, 0.7035, 0.0720, 1.2662} -> -47.28273
## 700: {3.2550, 2.5083, 1.5033, 0.5249, 0.6334, 0.9705} -> -50.15200
## 750: {4.2617, 5.2291, 3.7920, 4.6775, 0.3990, 0.4816} -> -51.61731
## 800: {6.1373, 1.9140, 6.2527, 0.7358, 0.3515, 0.6355} -> -51.44309
## 850: {4.3572, 1.4311, 1.3808, 0.7705, 0.5167, 0.3859} -> -47.69409
## 900: {3.8967, 2.2702, 2.1617, 1.4843, 0.0460, 0.3200} -> -46.64913
## 950: {3.2056, 2.2872, 2.2471, 0.8597, 0.0509, 0.8307} -> -47.87575
## 1000: {5.0346, 2.3960, 4.2488, 0.2151, 0.9305, 1.9769} -> -52.11050
## 1050: {4.0348, 2.1250, 1.6991, 0.8732, 0.1339, 0.4040} -> -46.45477
## 1100: {4.2386, 1.8443, 1.3563, 0.6864, 0.7396, 0.5141} -> -48.50665
## 1150: {4.1693, 2.8342, 3.3643, 1.6653, 0.4288, 1.5270} -> -49.27452
## 1200: {4.9863, 1.7707, 4.7648, 0.8115, 0.6273, 1.6092} -> -49.64246

```

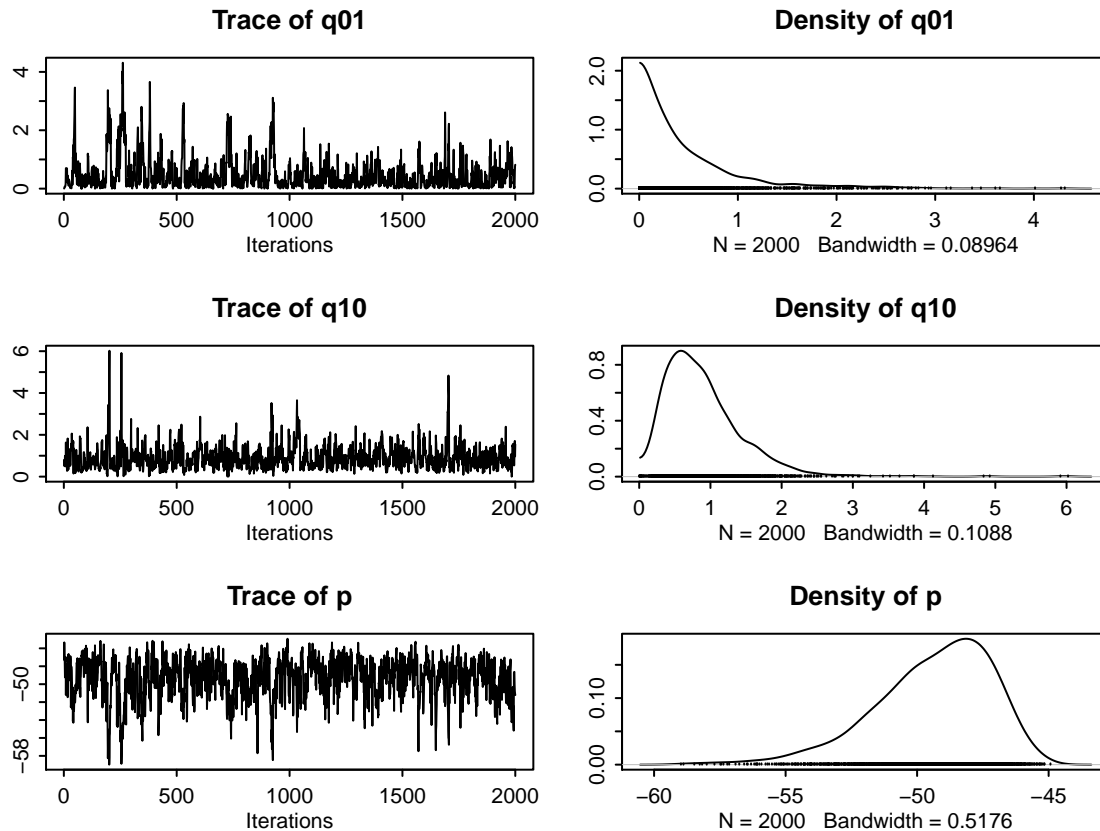
```
## 1250: {3.2966, 2.3823, 1.3403, 1.0409, 0.3136, 0.8005} -> -48.46872
## 1300: {3.6920, 1.9803, 2.3209, 0.3063, 0.0771, 0.8035} -> -46.66145
## 1350: {4.6574, 4.2888, 3.6239, 3.6588, 0.0602, 0.6375} -> -50.36257
## 1400: {5.6018, 1.4080, 2.9090, 0.3695, 0.9775, 0.4766} -> -51.51072
## 1450: {4.2435, 2.7328, 3.7943, 0.9062, 0.2983, 1.0341} -> -48.39468
## 1500: {4.3455, 1.7624, 3.5417, 0.4179, 0.1631, 0.9819} -> -46.56976
## 1550: {4.4916, 2.4927, 3.1034, 0.8262, 0.0463, 0.9154} -> -47.42348
## 1600: {4.4835, 1.9116, 5.2315, 0.1980, 0.0564, 1.4648} -> -48.08836
## 1650: {4.7910, 4.3477, 2.9314, 3.0940, 0.7874, 1.5844} -> -54.62997
## 1700: {4.6801, 1.6762, 4.2810, 0.3461, 0.7257, 3.0900} -> -52.21150
## 1750: {4.2556, 2.2771, 2.4035, 1.1964, 0.0852, 0.5151} -> -46.37698
## 1800: {6.7848, 2.5187, 6.8197, 1.5153, 0.1626, 0.7892} -> -49.56857
## 1850: {5.0261, 1.2318, 2.5519, 0.4775, 0.5980, 0.8935} -> -49.02914
## 1900: {4.6585, 2.3857, 3.8284, 0.3417, 0.1982, 1.2031} -> -47.51726
## 1950: {5.8492, 2.7413, 3.4021, 1.8611, 0.6503, 0.6947} -> -50.91695
## 2000: {5.5503, 2.3057, 4.9313, 0.5657, 0.8179, 1.5467} -> -51.21331
```

Now that we have the MCMC results, we will inspect the convergence of the MCMC chain using the coda package and we will plot the results.

```
require(lattice)
require(coda)
# Read the diversitree results in coda format
chain1 <- coda::mcmc(samples, start=min(samples$i), end=max(samples$i), thin=1)
# Trace plots of the parameters
op <- par(mar=c(3,2.5,3,1), mgp=c(1.5,0.5,0), tcl=-0.25)
plot(chain1[,c(2:5)])
```



```
plot(chain1[,c(6:8)])
```



```
# Look the effective sizes of the parameters after removing
# autocorrelation effects
effectiveSize(chain1[,c(2:7)])
```

```
##  lambda0  lambda1      mu0      mu1      q01      q10
## 347.5045 341.4298 252.0489 192.6293 160.7039 540.1457
```

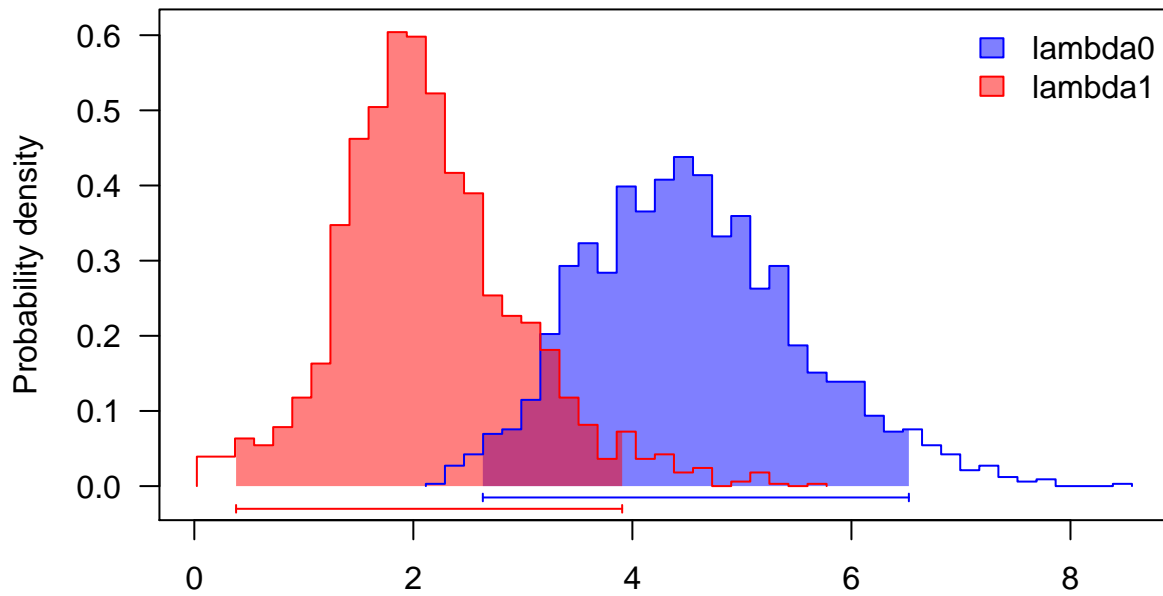
You can see that the chain is stable and the effective sample sizes (ESS) are correct (the chain should probably have run a bit longer). Ideally, you want ESS > 200 to have good parameter estimates. It is also a good advice in Bayesian analysis to run the analysis at least twice to confirm that the run have convergen on the same values.

Now, let's look at the profile plots of the parameters. Note that parameter 0 is for sheltered habitats and parameter 1 is for exposed habitats.

```
# Profile plots of the results
col <- c("blue", "red")
profiles.plot(samples[-(1:100),c("lambda0", "lambda1")], col.line=col, las=1,
              xlab="Speciation rate", legend="topright",
              main="Speciation rates")
```



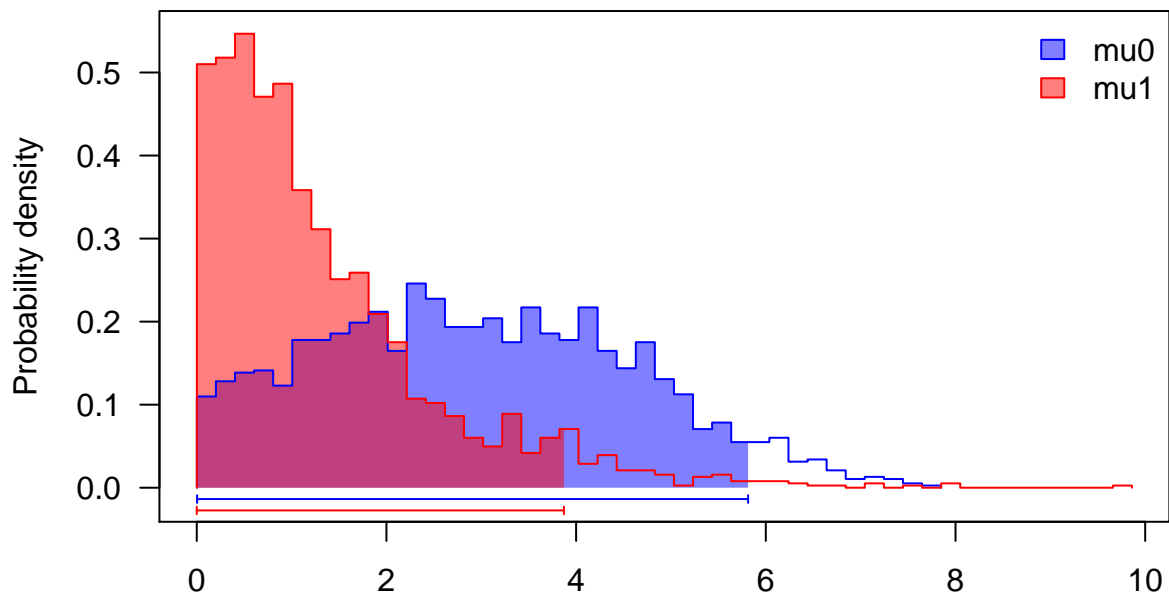
## Speciation rates



Speciation rate

```
profiles.plot(samples[-(1:100),c("mu0", "mu1")], col.line=col, las=1,
              xlab="Extinction rate", legend="topright",
              main="Speciation rates")
```

## Speciation rates



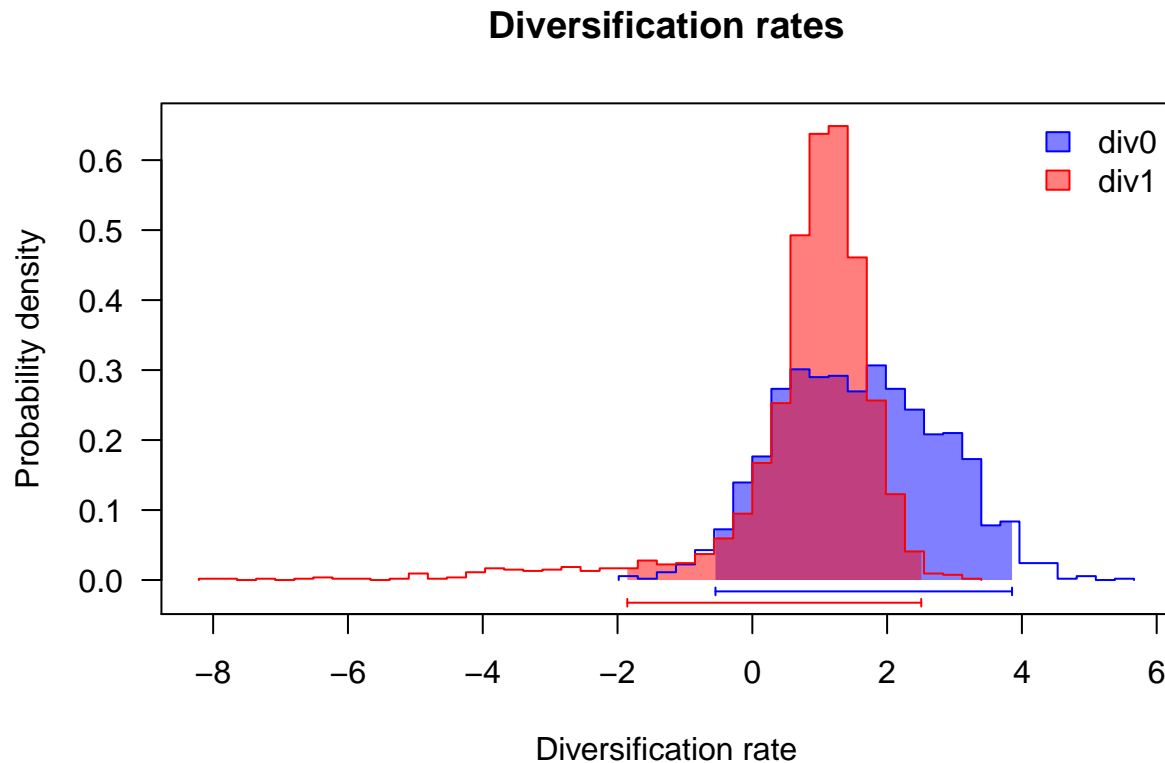
Extinction rate

These results suggest that the two groups are evolving under slightly different models because the distribution

are different. Note, however, that the distributions overlap considerably for both speciation and extinction. In other words, the difference in speciation and extinction rates between the two states is not significant. So there is a trend suggesting different rates, but it is not a clear difference.

It can also be interesting to look at the diversification rates, which is the speciation rate minus the extinction rate.

```
#diversification rates
diver<-data.frame(div0=samples$lambda0-samples$mu0,
                  div1=samples$lambda1-samples$mu1)
col <- c("blue", "red")
profiles.plot(diver[-(1:100),c("div0","div1")], col.line=col, las=1,
              xlab="Diversification rate", legend="topright",
              main="Diversification rates")
```



With this, you can clearly see that the two groups do not seem to have significant differences in net diversification rate.

Finally, it is possible to also run constrained models with the Bayesian approach. This could be useful for model testing using Bayes Factors for example. The code below would run the constrained model used above with ML, but here with a MCMC chain. Note that this analysis is not run here.

```
tmp.l <- diversitree::mcmc(lik.l, p[-c(1,3)], nsteps=100, prior=prior, w=.1, print.every=10)
```

## Other BISSE-like models

BISSE is restricted to only binary models, but there have been extensions of the BISSE model to other types of characters, such as multistate characters (MUSSE) or quantitative characters (QUASSE; Fitzjohn, 2010).

## The problem of pseudoreplication

Before finishing, I think it is important to say a word about the problem of pseudoreplication. Recently, different studies have highlighted that model testing using BISSE types of analyses can result in rejecting the null model more often than it should in certain situations (Maddison and Fitzjohn, 2015; Rabosky and Goldberg, 2015). In other words, you will reject the null hypothesis of no difference in diversification rates even if you should not.

Maddison and Fitzjohn (2015) show that this is partly due to the problem of pseudoreplication. For instance, when the states have not evolved repeatedly during the evolution of a group, then there is a chance that the conclusion will be biased. In other words, the “clades” compared might have different diversification rates, but absence of replication mean that you cannot assign these differences to the trait studied.

Rabosky and Goldberg (2015) further showed that the  $\alpha$  levels for statistical testing are biased. In other words, if you use a  $p$ -value of 5% for testing, you will reject the null hypothesis much more frequently than 5% of the time. However, this problem could be corrected using simulation to fix appropriate  $p$ -values.

If you are interesting in using these types of models, you should clearly read these two studies to be aware of the potential pitfalls with these models.

## Further reading

For further information regarding the estimation of diversification rates, I strongly recommend the very good book by Luke Harmon (2019).

## References

- Maddison W.P., FitzJohn R.G. 2015. The unsolved challenge to phylogenetic correlation tests for categorical characters. *Systematic Biology*. 64:127–136.
- FitzJohn R.G. 2010. Quantitative Traits and Diversification. *Systematic Biology*. 59:619–633.
- Harmon, L. J. 2019. Phylogenetic comparative methods. Distributed by the author. <https://lukejharmon.github.io/pcm/>
- Joly S., P.B. Heenan, P.J. Lockhart. 2014. Species radiation by niche shifts in New Zealand’s rockcresses (Pachycladon, Brassicaceae). *Systematic Biology*. 63:192–202.
- Maddison W.P., Midford P.E., Otto S.P. 2007. Estimating a binary character’s effect on speciation and extinction. *Systematic Biology*. 56:701–710.
- Rabosky D.L., Goldberg E.E. 2015. Model Inadequacy and Mistaken Inferences of Trait-Dependent Speciation. *Systematic Biology*. 64:340–355.
- Sánchez-Baracaldo P., Thomas G.H. 2014. Adaptation and Convergent Evolution within the Jamesonia-Eriosorus Complex in High-Elevation Biodiverse Andean Hotspots. *PLoS ONE*. 9:e110618.