Diversification

Simon Joly

BIO 6008 - Fall 2015

Contents

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.

```r
require(ape)
paramotree <- read.nexus("./data/JamesoniaPartFind_MCC.tre")
paramodata <- read.csv("./data/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$Name_in_tree
# Remove unecessary variables
paramodata <- paramodata[,-c(1:3,7:14)]
# Order the data in the same order as the tip.label of the tree. In the
# present example, this was already the case.
paramodata <- paramodata[paramotree$tip.label,]
```

Diversification rate

Diversification consists in the net accumulation of species through time. Basically, it consists of the speciation rate, formally defined as λ, minus the extinction rate, μ. If speciation rates are larger than extinction rates ($\lambda > \mu$), then the diversification rate is positive and the number of species in a group will increase. In contrast, if speciation rates are smaller than extinction rates ($\lambda < \mu$), then the diversification rate is negative and species number are diminishing through time in the group.

Diversification rates are estimated as the expected number of species that will be created from one species in a given amount of time. In a phylogenetic perspective, you can see this as the mean number of new lineages that will be created from one lineage in a given amount of time. This logic is also used in mathematical models that simulate phylogenies. They use a probability that any given lineage will speciate at any given time.

There are two main models for tree 'construction'. The first one only considers speciation and assume that there is no extinction. This is called the Yule model, or a pure birth model. The second model considers both speciation and extinction. This is the birth-death model. There are many other more complicated models, but they are less frequent in the literature and generally more complex. They often have speciation rates that vary according to other paramters (time, density-dependent, etc.).

Since there are mathematical models to simulate phylogenetic trees, this means we can also use these models to estimate parameter values using maximum likelihood. This allows to estimate diversification rates from a phylogeny. Here, we will use the package `laser` to estimate diversification rates. We will get diversification rates for a Yule model (pure birth) and a birth-death model. The corresponding functions in `laser` use branching times as input.

To get a calibrated rate of diversification per year, you need to give the total age of the tree (if it is not already defined by the branches lengths, such as when they are measured in years). In the present example, the branch lengths are already calibrated in number of years. If it was not the case, we could have calibrated the branching times in the function `scaleBranchingtimes` using the `basal =` argument.

```r
# Load packages
require(ape)
require(geiger)
require(laser)
# Estimate the diversification rate under a pure birth model. Because there
# is no extinction, the diversification rate equals the speciation rate.
diver.pb <- pureBirth(scaleBranchingtimes(getBtimes(string=write.tree(paramotree))))
# Estimate diversification times under a birth and death model
diver.bd <- bd(scaleBranchingtimes(getBtimes(string=write.tree(paramotree))))
results <- data.frame(model=c("Yule","birth-death"),
                      lambda=c(round(diver.pb$r1[[1]],3),round(diver.bd$r1,3)),
                      mu=c(0,diver.bd$a),
                      lnL=c(diver.pb$LH[[1]],diver.bd$LH),
                      aic=c(diver.pb$aic[[1]],diver.bd$aic))
# Order the lines from lowest AIC to higest AIC
results <- results[order(results$aic),]
results
```

```
##         model lambda        mu       lnL      aic
## 2 birth-death  0.032 0.7592125 -15.07987 34.15974
## 1        Yule  0.081 0.0000000 -18.52040 39.04081
```

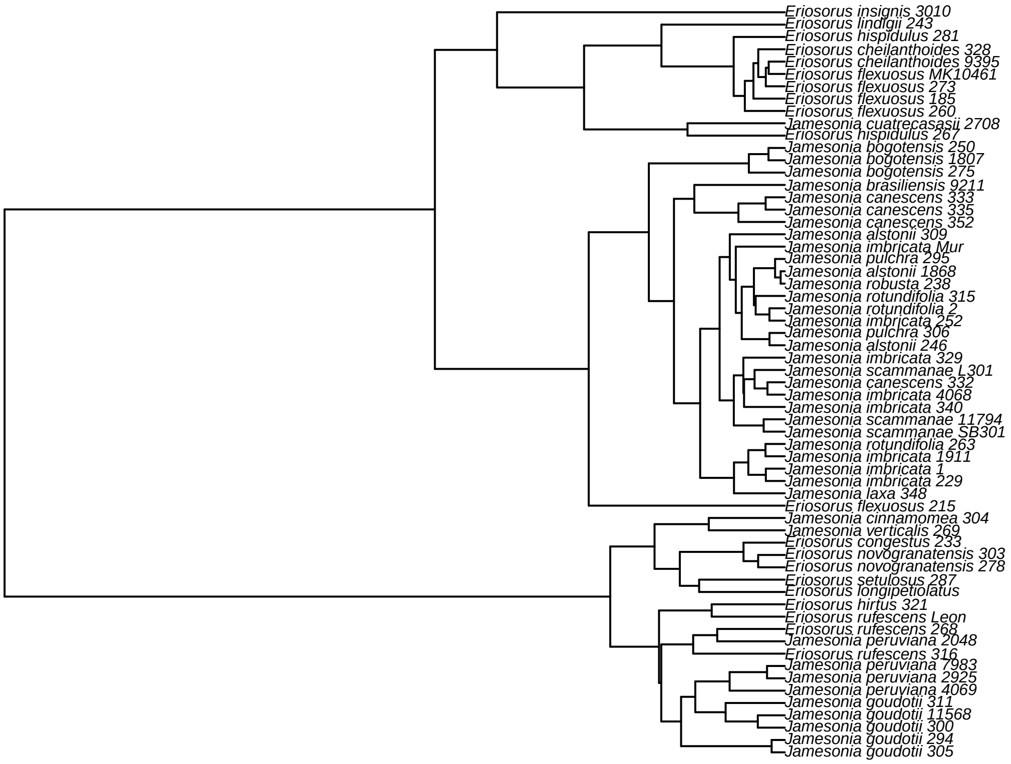
The results show that the birth-death model has a better fit (lower AIC). This model estimated a large extinction rate, which means that it indeed departs relatively importantly from a pure birth model. It suggests a speciation rate of 0.032 species per lineage per million of years, with an extinction rate of 0.759 species per lineage per million of years. You can see that it is a bit odd that the extinction rate is much higher than the speciation rate. We'll get back to this later.

Lineage through time plots

Another way to look at diversification is too 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 exponentially through time. You can check how the data departs from this expectation using lineage through time plots (ltt plots). These can be obtained using the `ltt` function of the `phytools` package.

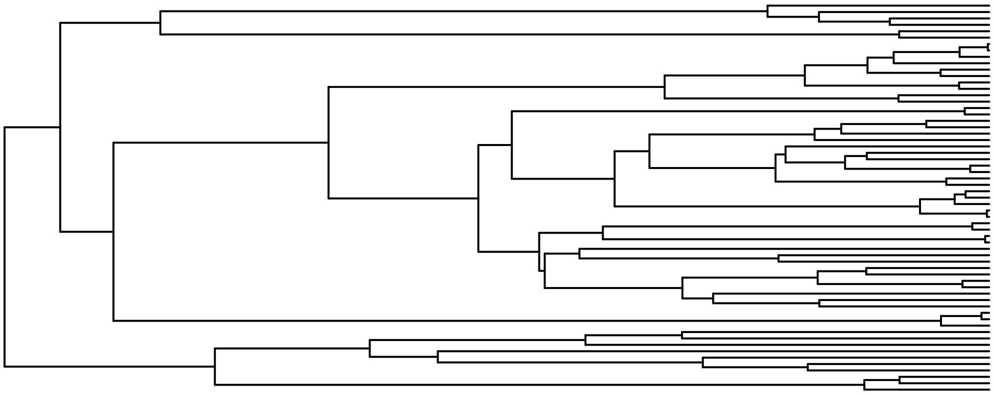
Let's first look at the species tree.

```r
plot(paramotree,show.tip.label=TRUE,cex=0.5)
```



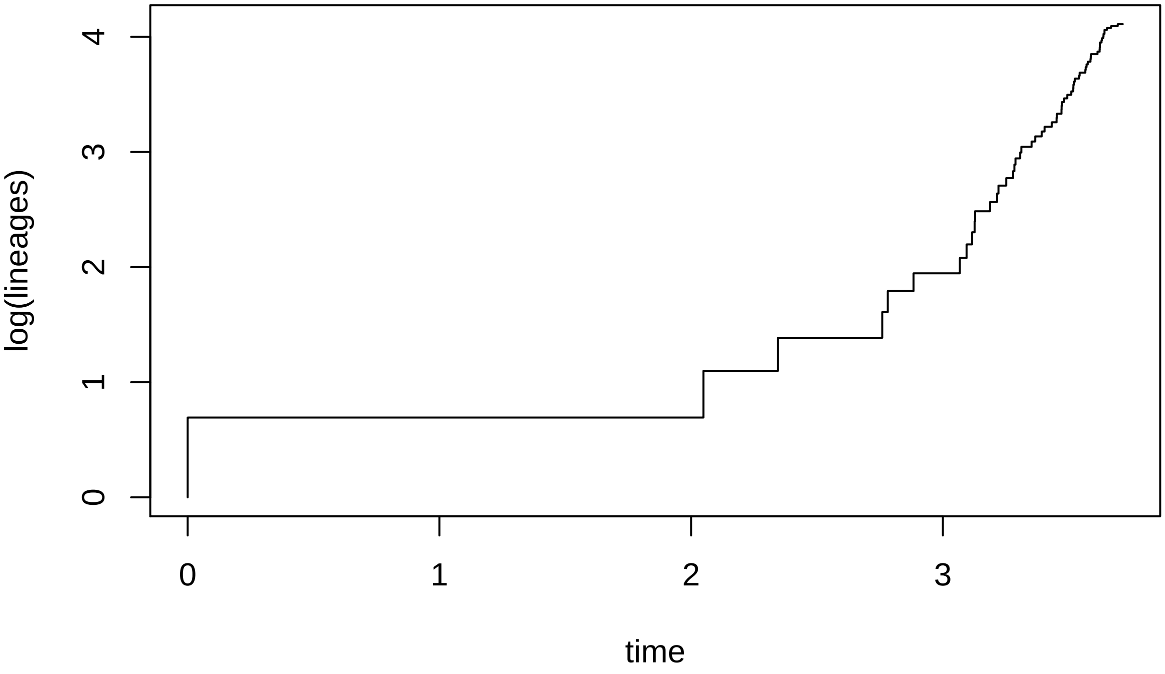
You can see that there are long branches at the base of the tree, which appears slightly unusual for a Yule tree. For comparison, see how a random Yule tree look like:

```r
require(phytools)
plot(pbtree(b=0.81,n=61),show.tip.label=FALSE)
```



Now, let's have a look at the ltt plot.

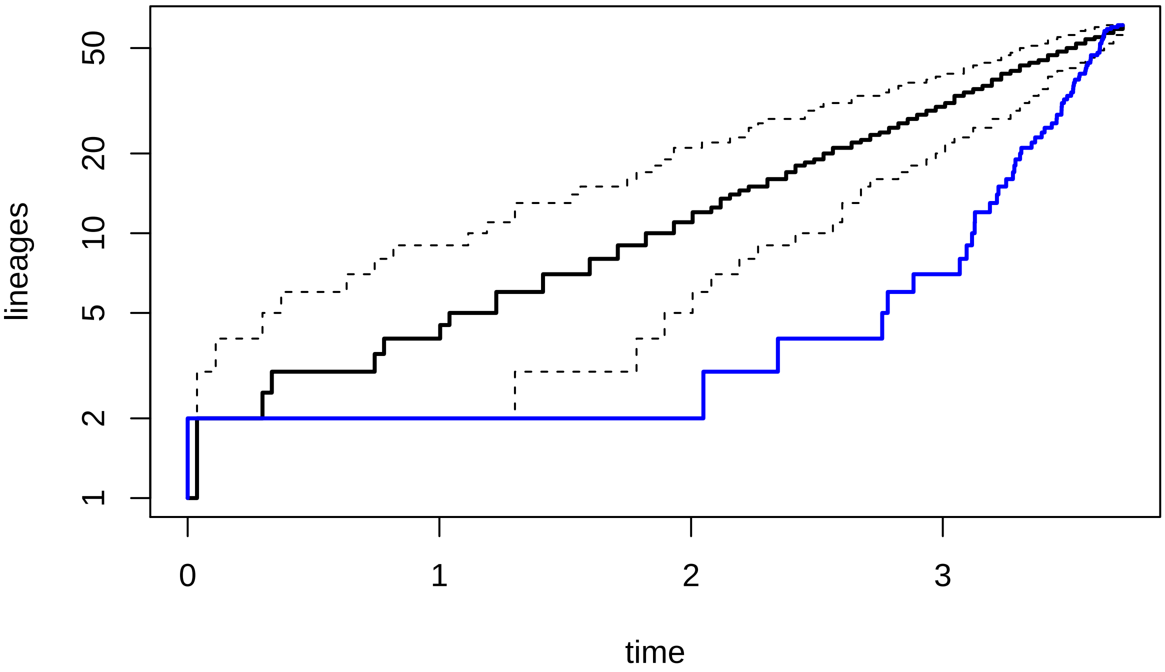
```r
res<-ltt(paramotree)
```



First, notice that the y axis is on a log scale. This is useful because the expectation for a Yule tree is that the number of species will increase exponentially with time (because each new species imply a new lineage from which new species can evolved, which will make new lineages, and so on). Consequently, this imply that on a log scale, the increase in the number of species should be linear. You can see that there is no species increase for the first 2 million years, and little until 3 million years. In contrast, the species number (on a log scale) for the last million of years has been increasing linearly with time, which implies an exponential growth for the last million of years.

You can compare this plot with expectations from a Yule model.

```r
# Simulate 100 Yule trees
yuletrees <- pbtree(n=61,scale=3.715,nsim=100)
# ltt null expectations with 95% CI
ltt95(yuletrees,gamma=FALSE,log="TRUE")
# Overlap the observed results in blue
lines(rep(res$times,each=2)[-1],rep(res$ltt,each=2)[-124],col="blue",lwd=2)
```



On this graphic, median estimates for the Yule expectations is representeby a bold line, and 95% credible intervals by dashed lines. The ltt plot for the observed data is in blue.

You can see that the observed accululation of lineages is clearly outside the 95% credible interval expected under a Yule model.

n-rates Yule models

The `laser` package also allow to fit Yule models in which there are n speciation rates through time (up to 5 in the current `laser` package; see help `?yule2rate`). These functions will find the best breakpoints where the rates are expected to have switched and will estimate the speciation rates for the different intervals.

With the present example, given that we observed a 'two step' speciation rates with the ltt plot, we will try to fit a 2-rates Yule model.

```
# Fit a 2-rates Yule model
diver.2r <- yule2rate(scaleBranchingtimes(getBtimes(string=write.tree(paramotree))))
# Store the results
results <- data.frame(model=c("Yule","birth-death","yule2rates"),
              diver.r1=c(round(diver.pb$r1[[1]],3),round(diver.bd$r1,3),round(diver.2r['r2'],3)),
              diver.r2=c(NA,NA,round(diver.2r['r1'],3)),
              shift=c(NA,NA,round(diver.2r['st1'],3)),
              mu=c(0,diver.bd$a,0),
              lnL=c(diver.pb$LH[[1]],diver.bd$LH,diver.2r['LH']),
              aic=c(diver.pb$aic[[1]],diver.bd$aic,diver.2r['AIC']))
# Reorder the lines according to the AIC of the models
results <- results[order(results$aic),]
results
```

```
##         model diver.r1 diver.r2 shift        mu       lnL      aic
## 3  yule2rates    0.020    0.091  1.69 0.0000000 -10.35521 26.71042
## 2 birth-death    0.032       NA    NA 0.7592125 -15.07987 34.15974
## 1        Yule    0.081       NA    NA 0.0000000 -18.52040 39.04081
```

Note that shift times, like branching times, are given in divergence units before present. Therefore, the 2-rates yule model inferred a shift in diversification rate 1.69 myr before present. The diversification rate before the shift was 0.02 and after 0.091, approximately 4.5 times higher. As you can see, this 2-rates model has the best fit of the three models tested (smallest AIC). Consequently, it more likely that there has been a rate shift between two pure-birth 'periods' during the evolution of the group rather than a single birth-death regime (with a large extinction rate). This looks much more realistic than having a much larger extinction rate than speciation rate as suggested with the birth-death model.

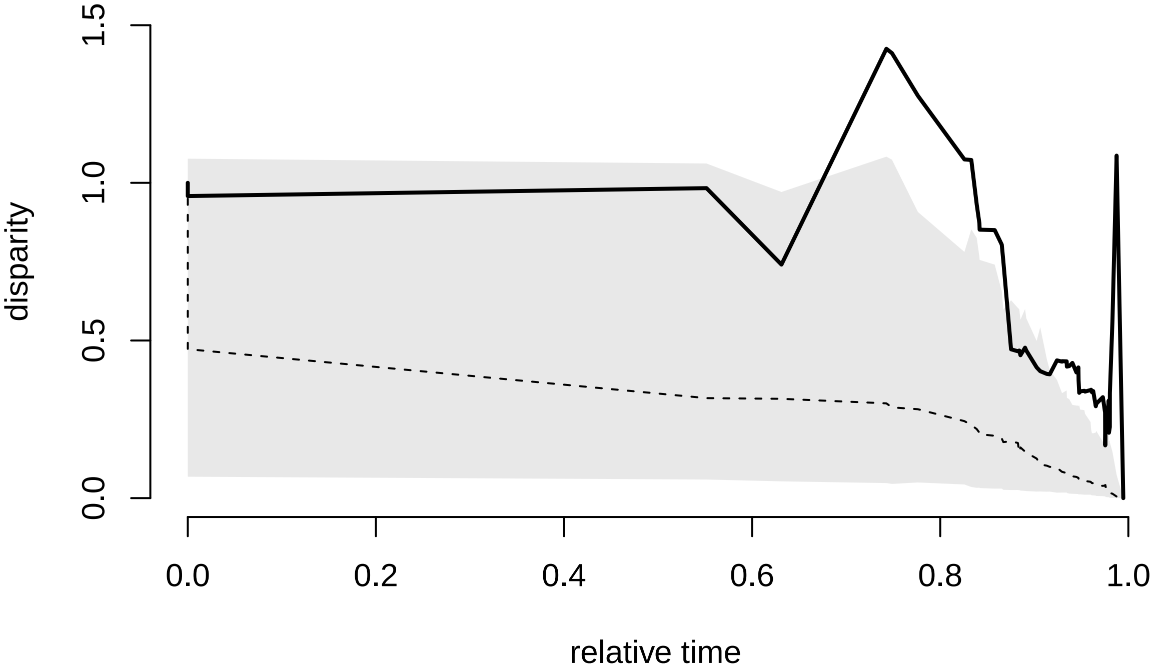
Disparity through time (dtt) plots

It is also possible the look at the diversification of one or more characters. This can be done using disparity through time (dtt) plots. The disparity is a measure of the average distance between trait values for a given group of species. In macroevolutionary studies, it is most common to use the average of the squared Euclidean distances between tips. Note that this also allow to calculate disparity for multiple traits at once.

The idea with the disparity through time function is to compare, at all time along the phylogeny, the observed disparity within clades with that between clades. This variance-based value varies between 0 and 1. Values near 0 at a specific time along the phylogeny imply that there is little disparity within clades and consequently that most ecological variation is partitioned between clades. At the opposite, values near 1 imply that most of the disparity is observed within clades. The `dtt` function compares the observed pattern with that obtained from simulations (by default using a Brownian motion model, although this can be modified manually; Joly et al. 2014).

Let's calculate a disparity through time plot for the mean species altitude for the ferm complex.

```r
# Replace NAs by the mean for the calculations
paramodata[is.na(paramodata[,"Altitude"]),"Altitude"] <- mean(paramodata[,"Altitude"],na.rm=TRUE)
# dtt plot
res_dtt <- dtt(paramotree,paramodata['Altitude'], index="avg.sq", nsim=100,
    calculateMDIp=T)
```



The plot show significantly high levels of disparity within clades around 0.5 and 0.85 (relative time) and in very recent groups. This means that during these time intervals, there is more diversity within clades than between clades, suggesting lots of diversification for this trait within these clades.

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 other words, it will estimate the speciation rates and the extinction rates for the two states of a binary character. It also has the possibility of estimating the transition rates between the two states, a bit like the mkn model that we saw in a previous lecture.

The BISSE model can have a up to 6 parameters:

- λ_1, λ_2, the speciation rates parameters for states 1 and 2
- μ_1, μ_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 speciation rates are significantly different for the two states by running two models. An unconstrained one, and another one for which the two speciation rates are constrained to be equal ($\lambda_1 = \lambda_2$). The two model could be compared as usual, for instance using the AIC.

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.

```r
require(diversitree)
# Convert the character into a numeric binary vector
char1 <- as.numeric(paramodata$ObservedMicrohabitat)
names(char1) <- row.names(paramodata)
# Create a BISSE likekihood 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)
```

```r
# 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.lambda=fit.l)
```

```
##              Df   lnLik    AIC  ChiSq Pr(>|Chi|)
## full          6 -30.253 72.506
## equal.lambda  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 suggest 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 tunning parameters, but these are hard to select *a priori*. The approach proposed by `diversitree` is to start by running a basic MCMC search to then estimate the tunning 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 diversificiation rate:
prior <- make.prior.exponential(0.1)

# Perform a first rapid search (with tunning 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 <- mcmc(lik, p, nsteps=100, prior=prior, w=10, print.every=10)
```

```
## 10: {4.7872, 4.7433, 4.0525, 3.7106, 0.5747, 1.5016} -> -52.50064
## 20: {3.5330, 3.8166, 2.1472, 2.6516, 0.1395, 0.3307} -> -49.96590
## 30: {4.1030, 2.1412, 4.5376, 1.2165, 0.0558, 0.6184} -> -49.07967
## 40: {5.1619, 0.8795, 2.2504, 0.3044, 1.0022, 1.4050} -> -52.02982
## 50: {4.9580, 2.3907, 4.9030, 1.6055, 0.8992, 1.8290} -> -51.37821
## 60: {5.5928, 1.3420, 3.8671, 3.8185, 0.8551, 0.0159} -> -52.40289
## 70: {4.9515, 1.2427, 3.1072, 0.6957, 1.5495, 1.7861} -> -52.56753
## 80: {5.4587, 0.3294, 3.7106, 2.9653, 1.7390, 0.6817} -> -51.10907
## 90: {4.0003, 1.0722, 0.9236, 8.4308, 3.2101, 0.4110} -> -53.40639
## 100: {3.4140, 0.4078, 1.6226, 3.3555, 1.5602, 0.9370} -> -52.26541
```

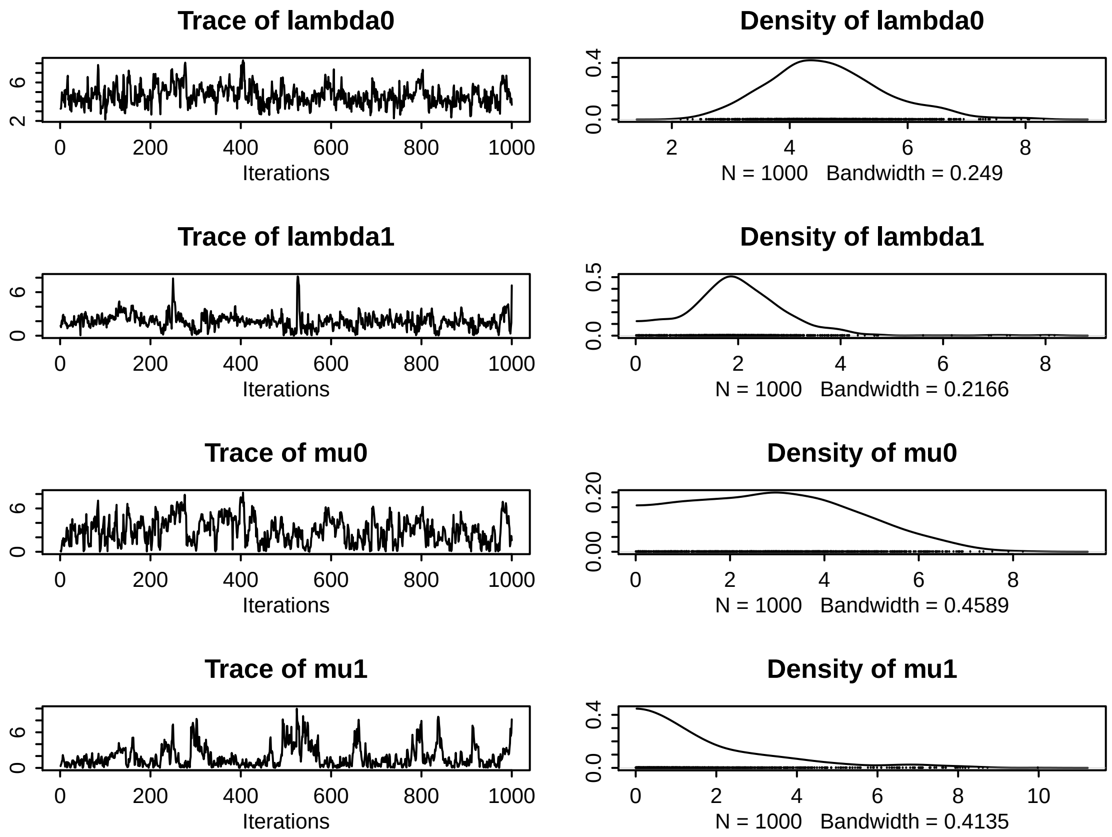
```r
# Then estimate the tunning parameters for the real analysis. For this
# we will remove the first ten generations.
w <- diff(sapply(tmp[-(1:10),2:7], range))

# Then run the true analysis. We will use 1000 steps here. Again, you
# may need to run the MCMC chain longer for large analyses.
samples <- mcmc(lik, fit$par, nsteps=1000, prior=prior, w=w,
                print.every=50)
```

```
## 50: {4.0752, 2.4304, 3.7813, 0.0148, 0.1113, 0.8621} -> -49.40855
## 100: {2.1487, 2.5037, 1.3729, 1.7698, 0.6461, 1.0042} -> -52.85691
## 150: {6.5724, 2.1571, 5.4051, 0.1380, 0.0378, 1.6185} -> -48.93692
## 200: {5.0756, 2.6262, 2.7358, 1.2223, 0.3486, 0.5850} -> -48.88515
## 250: {6.0397, 7.8981, 5.0947, 7.3159, 0.4690, 0.4209} -> -56.67969
## 300: {4.0689, 0.4684, 0.5878, 3.5682, 2.8714, 0.8003} -> -51.85444
## 350: {5.7830, 1.3940, 3.8684, 0.1719, 0.6039, 0.9384} -> -49.20436
## 400: {6.5330, 2.3465, 7.5576, 0.4752, 0.2944, 1.6118} -> -51.22570
## 450: {3.0605, 2.0898, 0.7550, 1.2310, 0.2981, 0.1928} -> -48.57884
## 500: {3.7889, 0.4717, 1.9420, 3.7396, 2.2273, 0.0405} -> -50.64669
## 550: {4.1337, 1.1624, 0.0685, 2.9496, 2.8932, 0.1641} -> -53.32183
## 600: {4.2125, 2.2787, 4.2347, 0.9527, 0.2412, 0.9653} -> -47.95932
## 650: {4.3529, 1.1158, 0.6999, 4.0945, 3.8608, 1.3748} -> -54.76683
## 700: {5.0765, 2.3273, 3.6233, 0.9518, 0.1295, 0.9552} -> -47.33326
## 750: {3.9390, 2.4826, 5.4314, 0.7083, 0.0947, 1.4484} -> -51.16095
## 800: {6.4342, 3.7524, 5.1377, 7.9285, 2.1549, 0.6392} -> -57.56364
## 850: {4.0742, 2.6543, 3.1954, 1.3534, 0.6215, 1.1880} -> -48.89511
## 900: {4.5907, 1.7410, 1.4791, 0.2728, 0.0549, 0.4134} -> -46.04402
## 950: {2.9520, 1.9754, 0.0618, 1.4128, 0.0716, 0.2970} -> -46.29177
## 1000: {3.6759, 6.9373, 1.5918, 8.1792, 1.5078, 1.1268} -> -57.69369
```

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.

```r
require(lattice)
require(coda)
# Read the BayesTrait results in coda format
chain1 <- 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)])
```



Trace of lambda0

Density of lambda0

Trace of lambda1

Density of lambda1

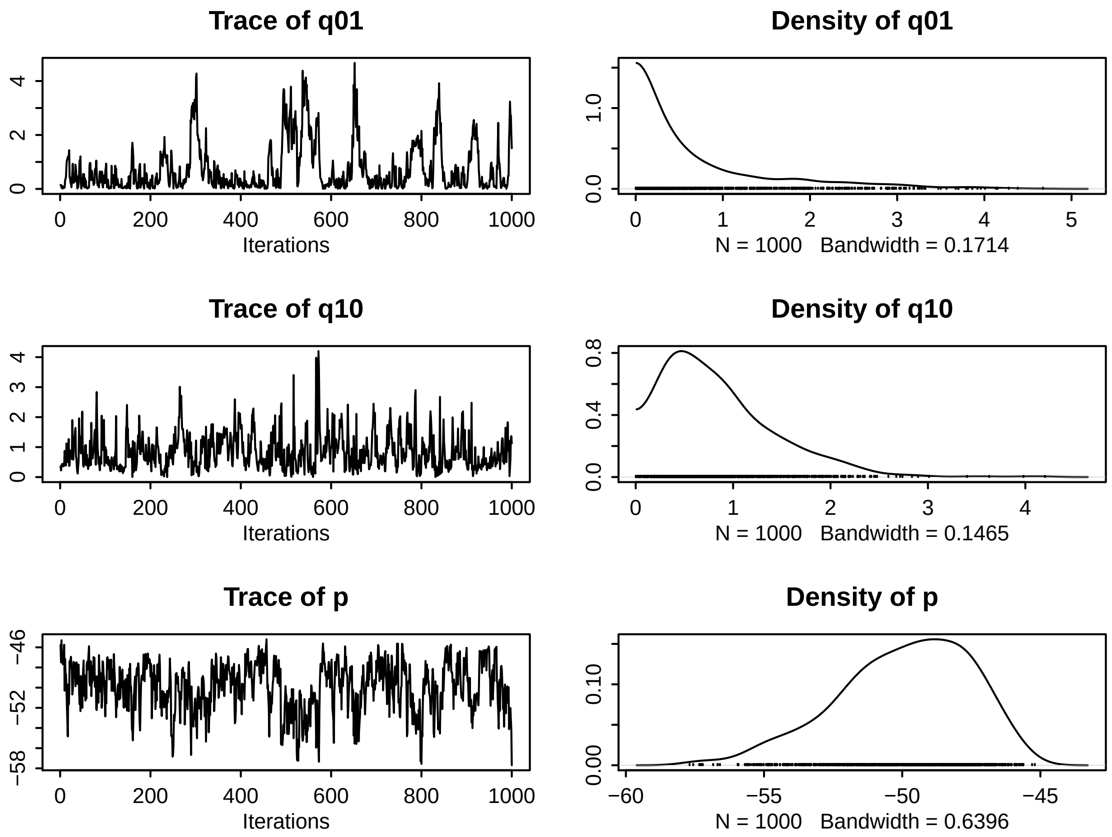
Trace of mu0

Density of mu0

Trace of mu1

Density of mu1

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



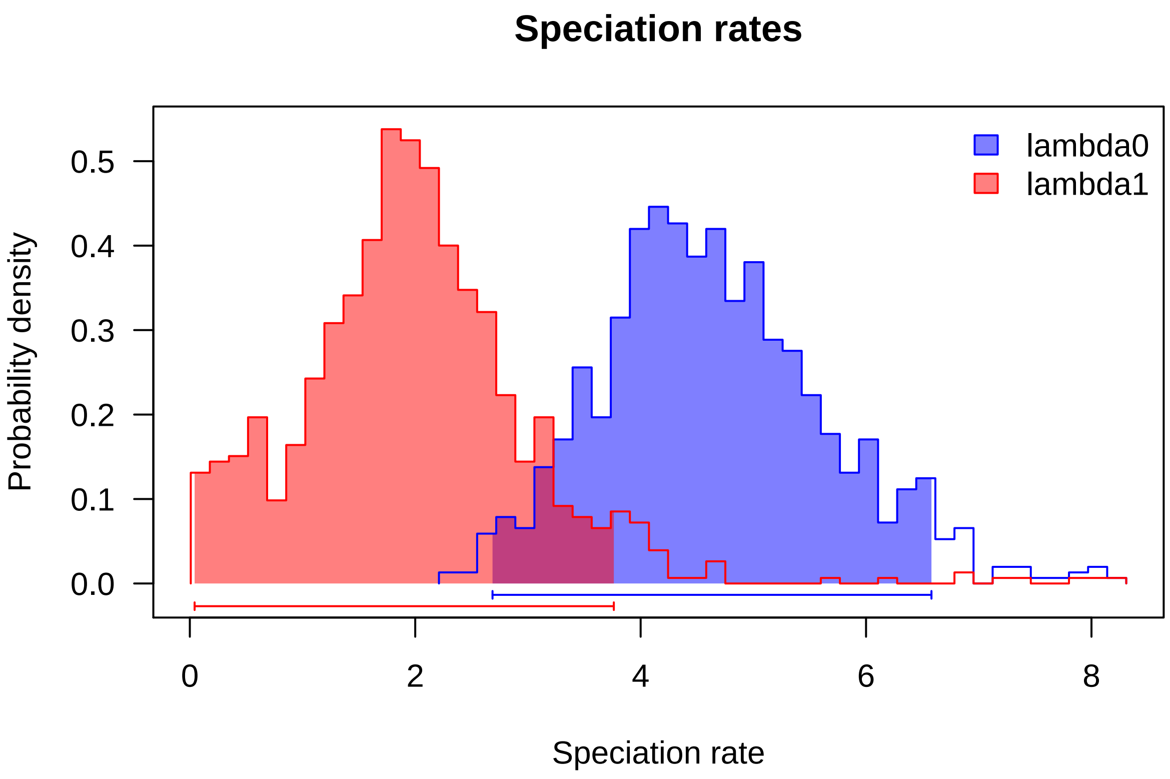
```
# Look the effective sizes of the parameters after removing
# autocorrelation effects
effectiveSize(chain1[,c(2:7)])
```

```
##   lambda0   lambda1       mu0       mu1       q01       q10
## 152.93769 162.63528  99.18270  47.27265  40.63164 234.50245
```

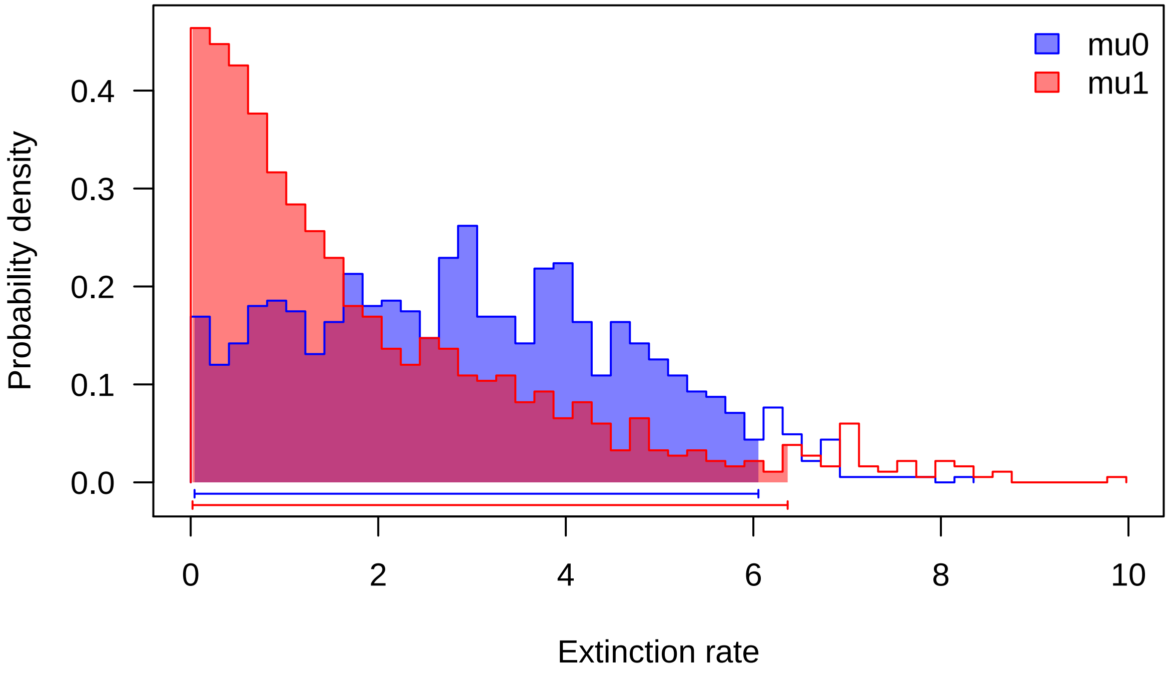
You can see that the chain is stable and the effective sample sizes (ESS) are correct (the chain should probably have run a bit longer). Ideally, you should run the analysis at least twice to confirm convergence.

Now, let's look at the profile plots of the parameters.

```
# 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")
```

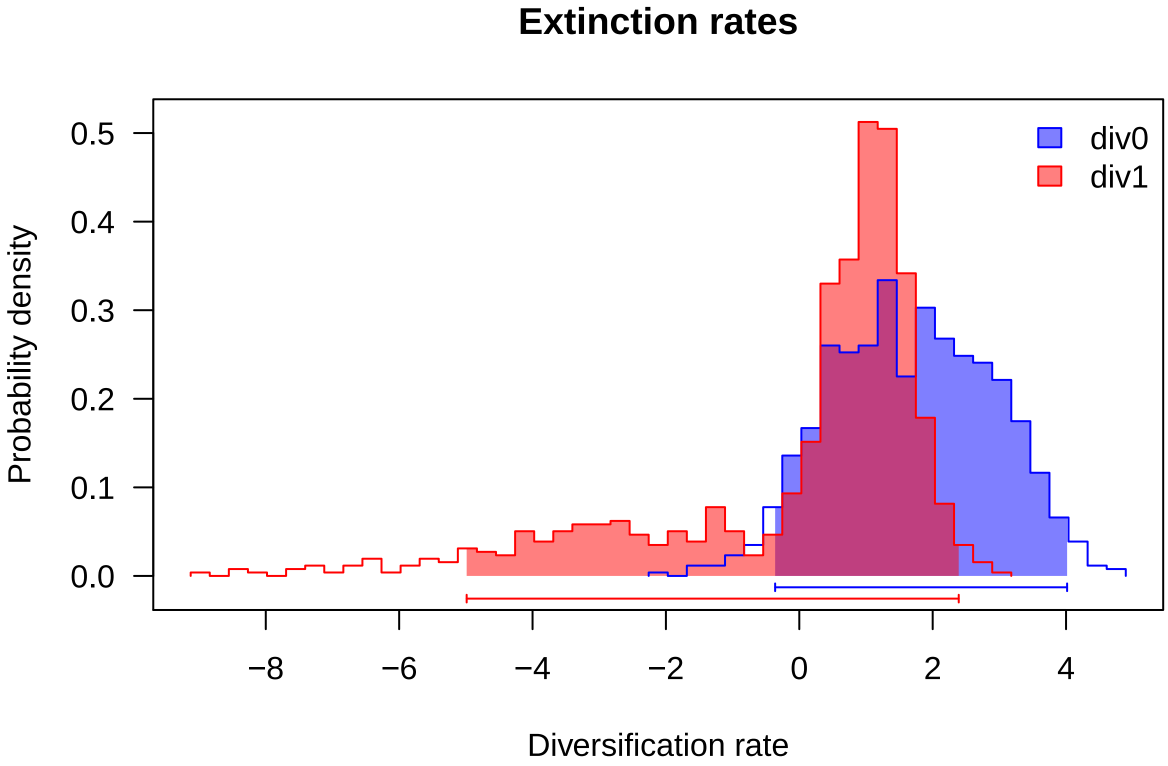


```
profiles.plot(samples[-(1:100),c("mu0", "mu1")], col.line=col, las=1,
              xlab="Extinction rate", legend="topright")
```



These results suggest that the two groups are evolving under different models, although the distribution overlap considerably for both speciation and extinction. It other words, the difference in speciation and extinction rates between the two states is not significant. Yet, 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="Extinction rates")
```



With this, you can clearly see that the two group do not seem to have significant differences in net diversifiaction 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 <- mcmc(lik.l, p[-c(1,3)], nsteps=100, prior=prior, w=.1, print.every=1)
```

Other BISSE-like models

BISSE is restricted to only binary models, but there has been extentions of the BISSE model to other type 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 repeatidly 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 α 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.

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.

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.