# MCMC log files

Heath Blackmon

4/14/2021

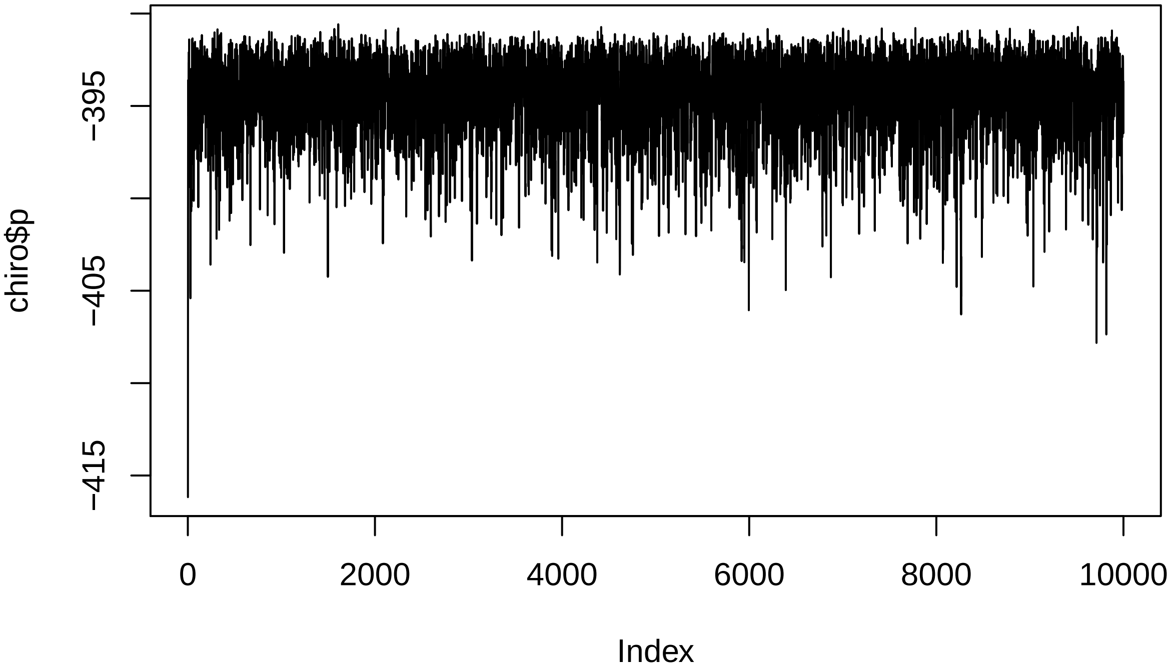
### MCMC Log files

## Burnin

While MCMC provides a method to analyze and fit complex models they present a number of unique challenges. The first of these is convergence. An MCMC is considered to have converged once it is sampling from the target distribution. However, when we are using a real empirical dataset we don't know what that target distribution is. So how can we decide if our MCMC has run long enough? The most common way is to look at the likelihood over time.

Lets look at this for a simple MCMC that I ran to estimate the rates of chromosomal mutations in bats.

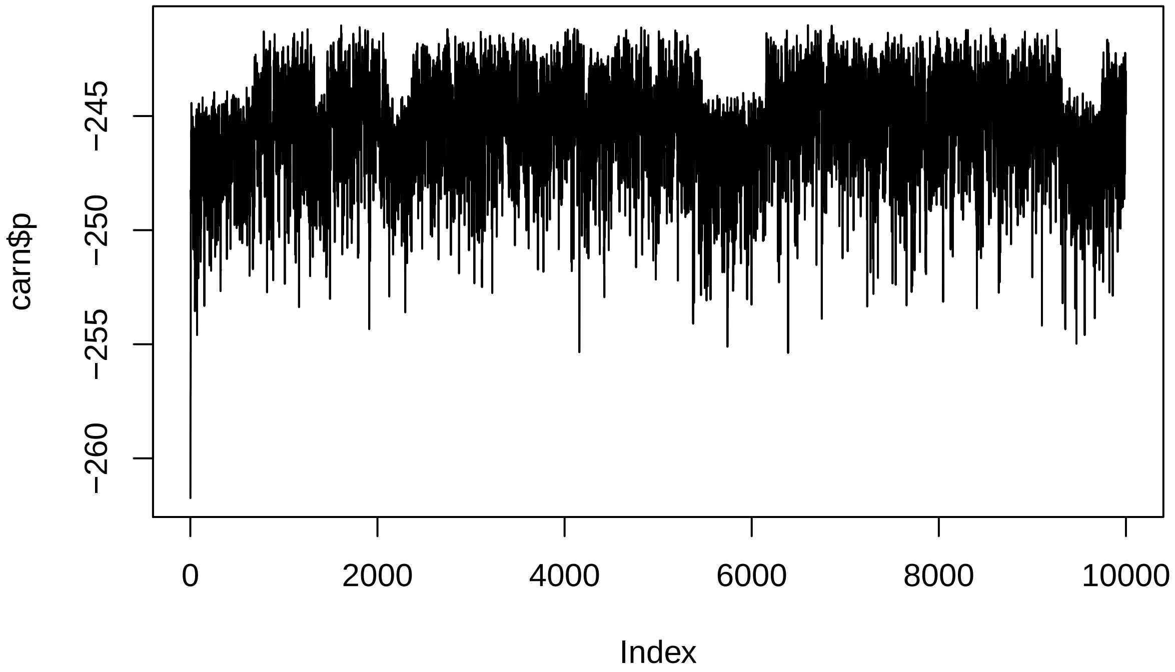
```r
chiro <- read.csv("chiro.csv")
plot(chiro$p, type="l")
```



Here what we see is the likelihood starting out very low and quickly increasing and then bouncing around the same value for many many generations with no consistant increase over time. This is a good sign that things are working well.

Lets contrast this with an MCMC I ran using the same model but in Carnivores.

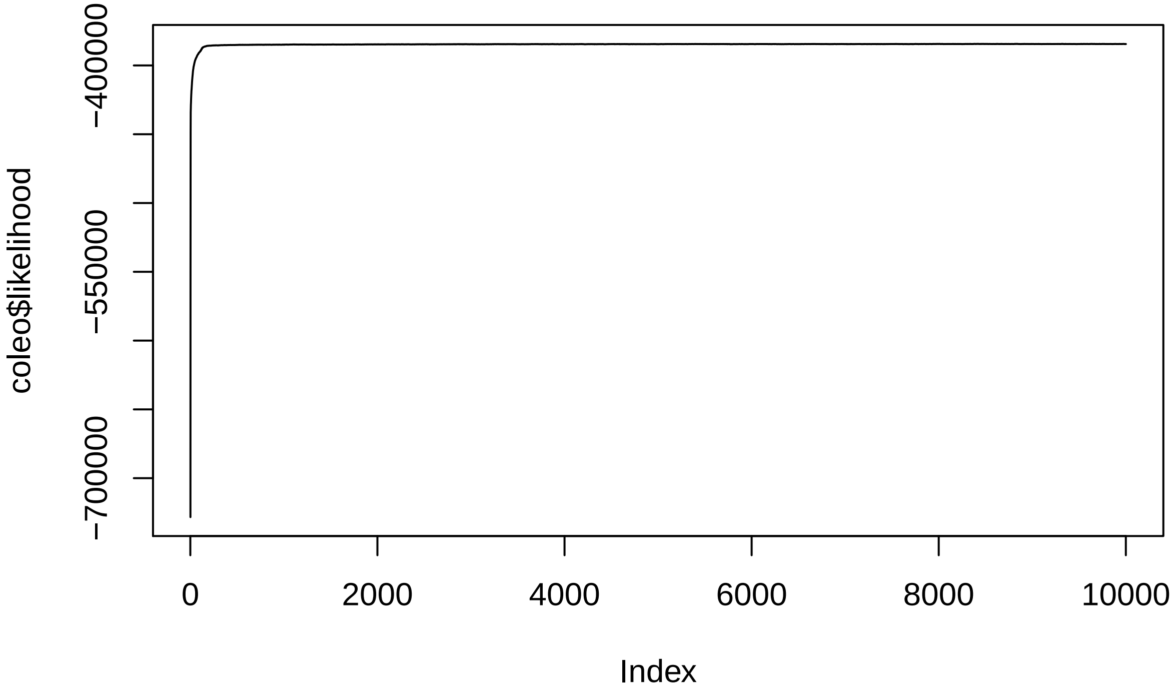
```r
carn <- read.csv("carn.csv")
plot(carn$p, type="l")
```



This looks very different. What we see here is that at different times during the run it is sampling from parameters that lead to a likelihood that is lower and it is gettign "stuck" in this region sampling from a different distribution for 100s of generations before moving back to a different set of parameters that have higher likelihood.

Another common problem that we find is that our likelihood hasn't stopped increasing during our MCMC run. This is a sign that we simply haven't allowed it to run for long enough. Lets look at an example of this with an MCMC run from a phylogenetic inference I did for beetles. This analysis was done with beast and I ran it for 60 million generations taking one sample of the MCMC every 6000 generations. When we will need to run our MCMC for a very long time we often only record the current values from the MCMC at some interval to avoid having a log file that is too large to deal with.

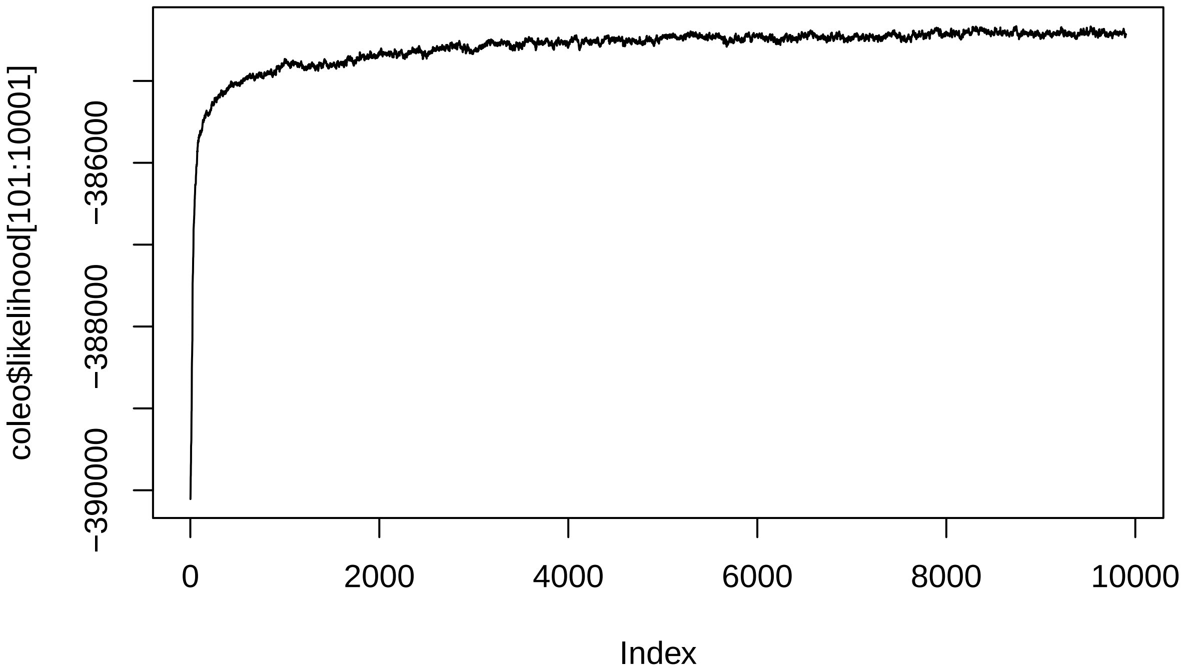
```r
coleo <- read.csv("coleoptera.csv")
plot(coleo$likelihood, type="l")
```



You might be tempted to think that that looks great it increases quickly and then stays really about the same the rest of the time. However, remember it should be bouncing around that area not a flat line. What is actually happening here is that the initial value has such a bad likelihood our scale is hiding what is going on as the MCMC ran.

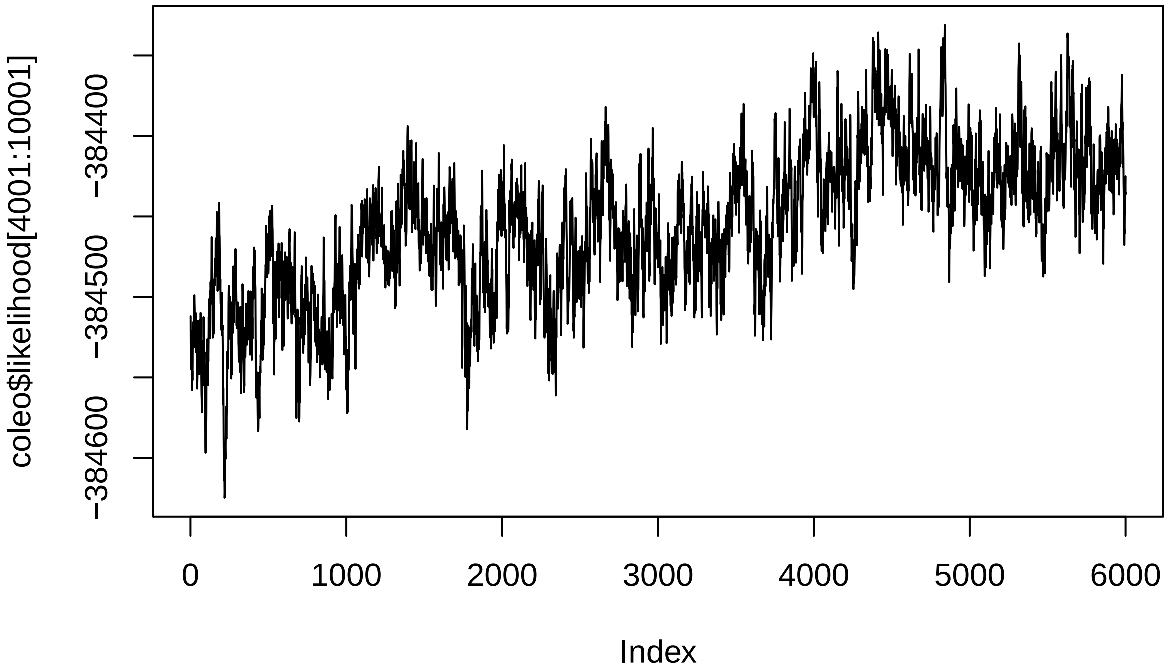
To get around this problem lets throw out the first 100 generations and see what our plot looks like

```
coleo <- read.csv("coleoptera.csv")
plot(coleo$likelihood[101:10001], type="l")
```



From this plot I can see that the likelihood is increasing at least out to 4000 (remember this is actually generation 4000x6000 or 24 million generations.) So lets try one more plot starting at 4000.

```
coleo <- read.csv("coleoptera.csv")
plot(coleo$likelihood[4001:10001], type="l")
```



This plot shows us that actually it was still increasing long into the MCMC run at least to around 54 million generations. The way that I dealt with this was to actually start two more MCMC runs that began at the point that this MCMC had ended. I allowed each of these to run for an additional 60 million generations and found that they had not increseased for the final 55 million or so generations. Based on this I threw away all the first 80 million generations of my MCMC as burnin and kept just the last 40 million generations. From this post burnin portion I sampled 100 trees that beast had logged during the MCMC run and used these in my paper.

## Parameter estimates

The whole reason we are running these MCMCs is to estimate something (Beta coefficients, rate parameters, phylogenies, etc.) Let us the chiroptera example to determine the rate of fissions, fusions in bats. If you will remember from my description of this study I had a binary predictor variable that I had hypothesized should impact rates of chromosome evolution. Lets see if the data from the MCMC supports this.

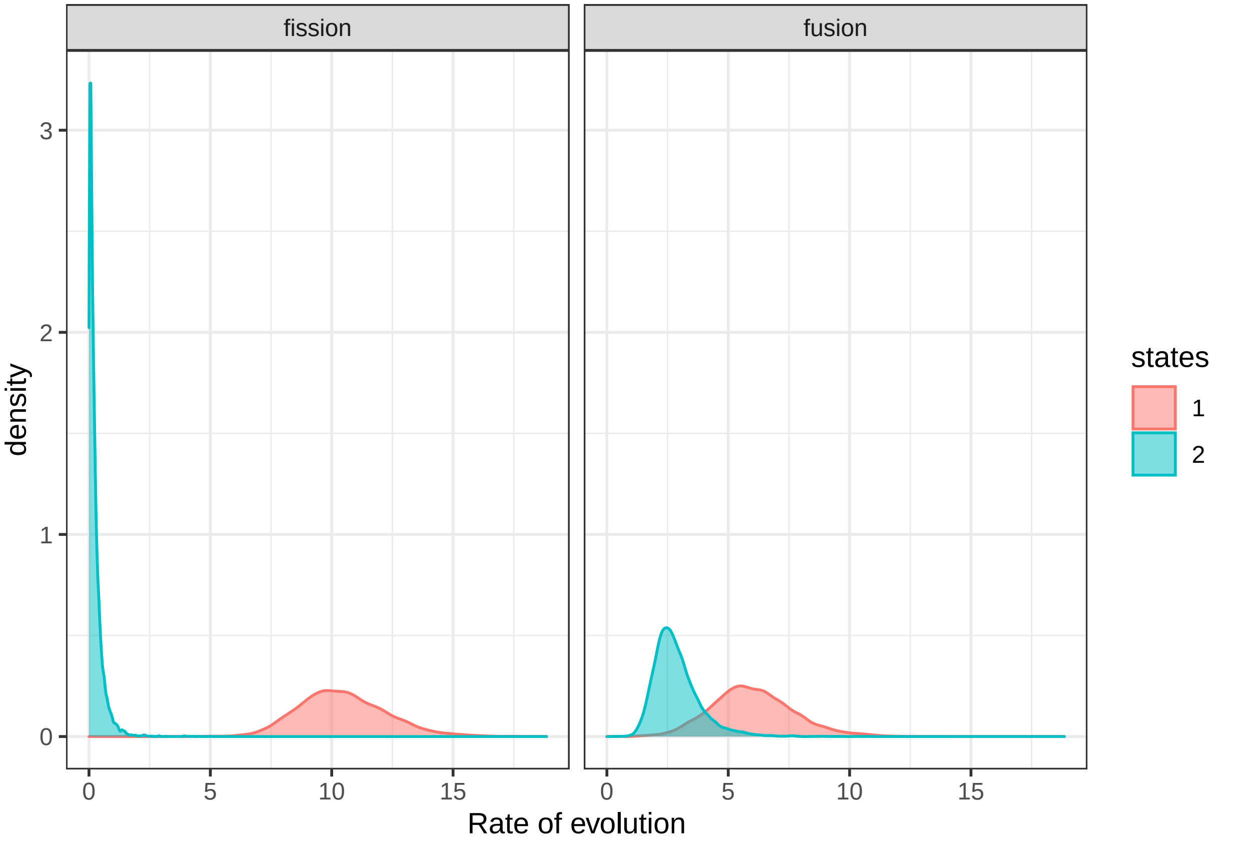
Step one is to get rid of the burnin. We have already looked at this above and we found that the MCMC reached convergence quickly. Just to be conservative though lets throw out the first 25% of the MCMC as burnin.

```
chiro.good <- chiro[2501:10000,]
chiro.good[1:10,]
```

```
##         X      asc1    desc1       asc2    desc2     tran12    tran21         p
## 2501 2501 10.862437 4.547664 0.09953007 1.974515 0.09816622 1.0288499 -392.1829
## 2502 2502  9.799941 5.283289 0.31683875 2.737887 0.04878692 0.8743853 -392.7809
## 2503 2503 12.760147 8.427741 0.13123053 2.725310 0.05943413 1.0973381 -392.9471
## 2504 2504 13.004196 8.131303 0.08790796 1.827864 0.14338673 0.8858151 -393.1570
## 2505 2505 10.412503 7.728854 0.10091535 2.247651 0.14160660 0.6451104 -392.0854
## 2506 2506 11.269622 5.469012 0.28751460 3.179241 0.18710257 0.7607755 -393.3903
## 2507 2507  7.772259 4.442336 0.63752810 5.160839 0.16769665 0.8799607 -396.4920
## 2508 2508  9.520745 3.697236 0.27597014 3.080427 0.16818077 0.5532408 -393.3659
## 2509 2509  8.648233 3.361593 0.07516096 2.490656 0.12913142 0.6133096 -392.8384
## 2510 2510  7.842161 6.708368 0.17384592 2.458964 0.20014400 0.5936476 -393.8603
```

We are interested in the four columns asc1, desc1, asc2, desc2. Asc and desc stand for ascending and descending chromosome number or fissions (ascending change in chromosome number) and fusions (descending change in chromosome number). The 1 and 2 stand for state 1 or state 2 of the predictor variable that I believed would lead to different rates. lets make a plot that illustrates the difference in these different rate parameters.

```
library(ggplot2)
# first lets get the data in long format
types <- rep(c("fission","fusion"), each=15000)
states <- rep(c("1","2","1","2"), each=7500)
new.dat <- data.frame(c(chiro.good$asc1,
                        chiro.good$asc2,
                        chiro.good$desc1,
                        chiro.good$desc2),
                      types,
                      states)
colnames(new.dat)[1] <- "rates"
ggplot(new.dat, aes(x=rates)) +
  geom_density(aes(fill=as.factor(states), colour=states, y=..density..),     stat="density",alpha=0.5)
  facet_grid(. ~ types) +
  theme_bw() +
  guides(fill=guide_legend(title="states")) +
  xlab("Rate of evolution") + ylab("density")
```



Now we have a plot of the values sampled during the post-burnin portion. As mentioned earlier this semester it would be inappropriate to calculate a simple frequentist confidence interval for these values (we could run the MCMC indefinitely long to have a near infinite sample size leading to an arbitrarilly small confidence interval.) Instead we should use something like a Bayesian credible interval which we can get quite easily usign the R package CODA.

```
library(coda)
HPDinterval(as.mcmc(chiro.good[,2:5]))
```

```
##              lower      upper
## asc1  7.056663e+00 13.8910029
## desc1 2.778325e+00  9.3741892
## asc2  1.548243e-05  0.8177126
## desc2 1.288941e+00  4.7650005
## attr(,"Probability")
## [1] 0.95
```

This shows us that the credible interval of fissions does not overlap but that the credible interval of fusions overlaps slightly.