Supplementary materials for: Pulled Diversification Rates, Lineages-Through-Time Plots and Modern Macroevolutionary Modelling

Andrew J. Helmstetter

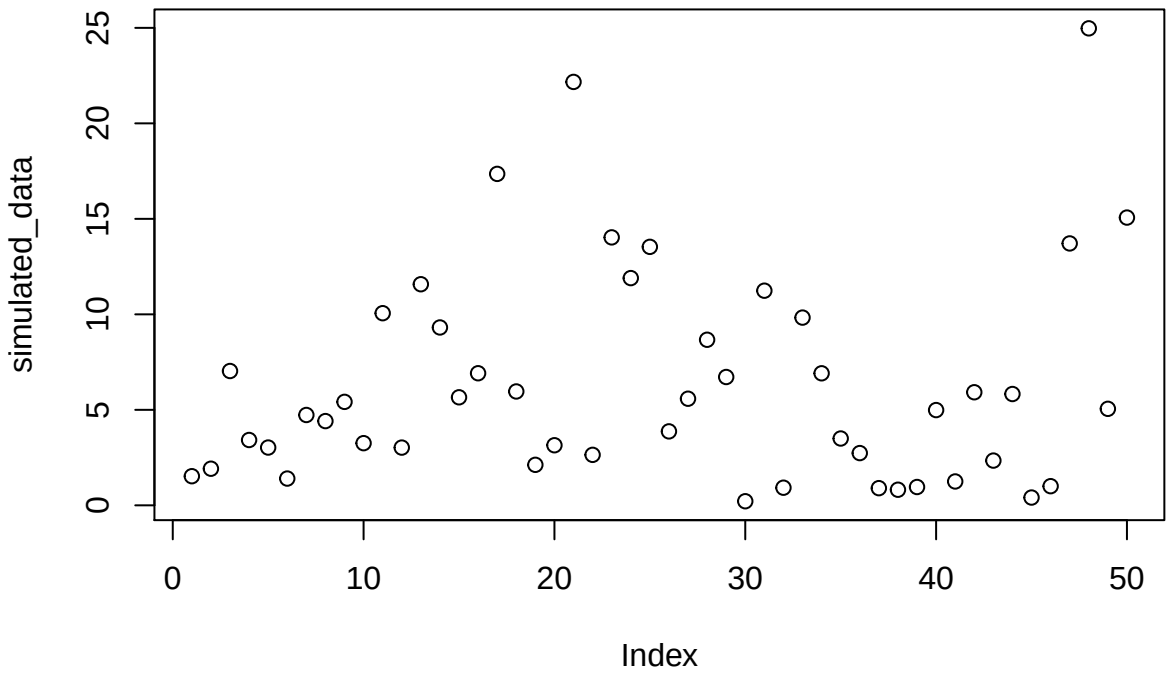
5/7/2021

Section 1: Simulating figure 2

Here we detail the simulations run to produce figure 2. Briefly, we simulated data and conducted a Bayesian Markov Chain Monte Carlo (MCMC) approach to estimate the values of two parameters, a and b when our knowledge is of the difference between these two parameters, or the slope. Though not using a birth-death model, this can be though of as a simplification of the process we go through when we try to estimate the speciation and extinction rate, when the data we have is related to their product - the net diversification rate.

First, we set the true values of the two parameters a & b, and simulate 50 data points from a gamma distribution with a rate of $a - b$.

```
#number of simulated data points
n =  50

#true value of parameter1
a_true = 0.5

#true value of parameter2
b_true = 0.25

#sum of true values
diff_true <- a_true - b_true

#simulate data with gamma distribution
simulated_data <- rgamma(n, #number of simulated data points
                         shape = 2, #shape of gamma distribution
                         rate = diff_true) #scale


#look at simulated data
plot(simulated_data)
```



Next, we generate random starting values for a and b for our chain, taking them from an exponential distribution.

```
#rate for exponential distribution
alpha = 0.5

#draw 2 values from exponential distribution
a <- rexp(1, 1 / alpha)
b <- rexp(1, 1 / alpha)
```

We define functions to calculate the likelihood and prior.

```
#function to calculate likelihood
likelihood <- function(a, b, datos) {
  if ((a - b) < 0) {
    like <- 0
    return(like) #fail
  } else{
    like <-
      prod(dgamma(datos, 2, rate = (a - b))) #product of all vectors of elements in gamma dist
    return(like)
  }
}

#function to calculate prior
prior <- function(a, b) {
  if ((a - b) < 0) {
```

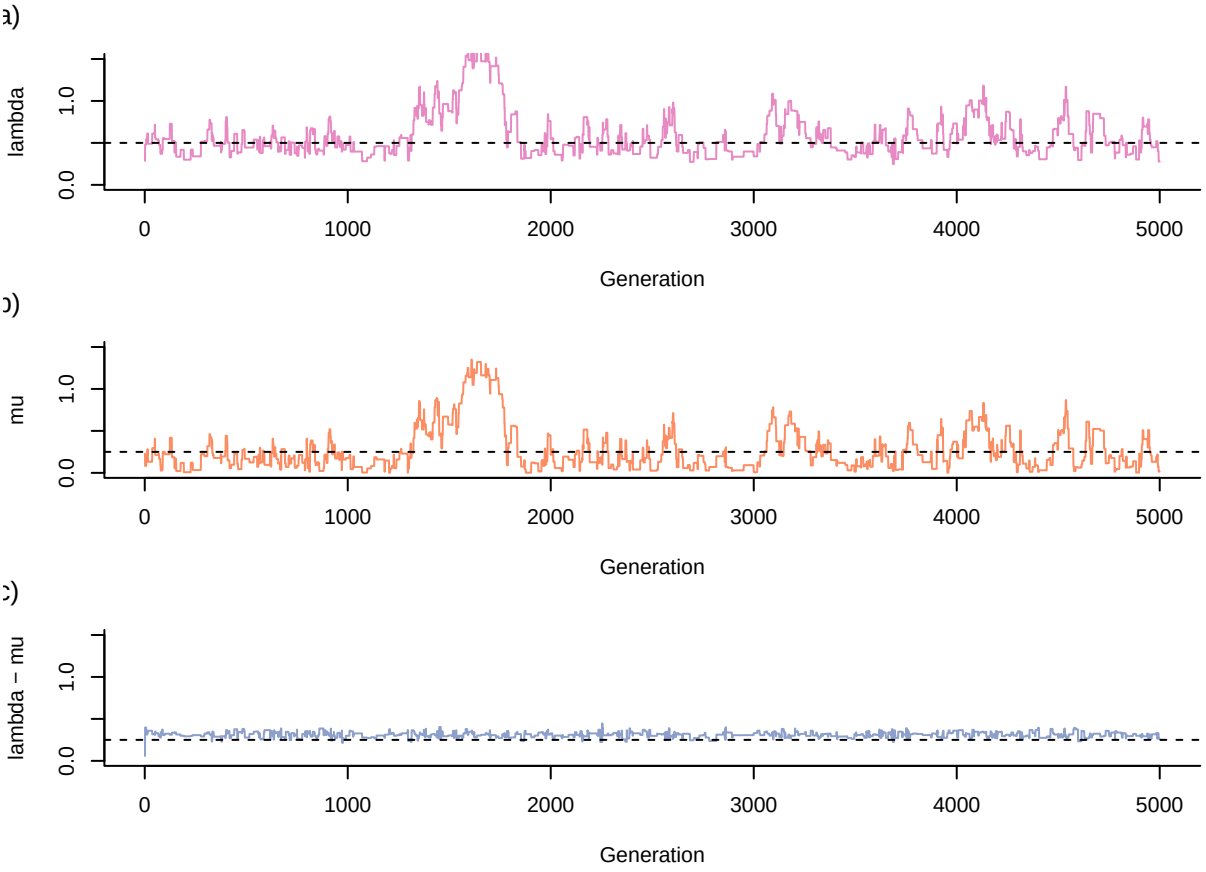
```
    prior.val <- 0
    return(prior.val)
  } else{
    prior.val <- dexp(a, 1 / alpha) * dexp(b, 1 / alpha)
    return(prior.val)
  }
}
```

Then we run an MCMC chain of 5000 generations, logging the results of the run as it progresses.

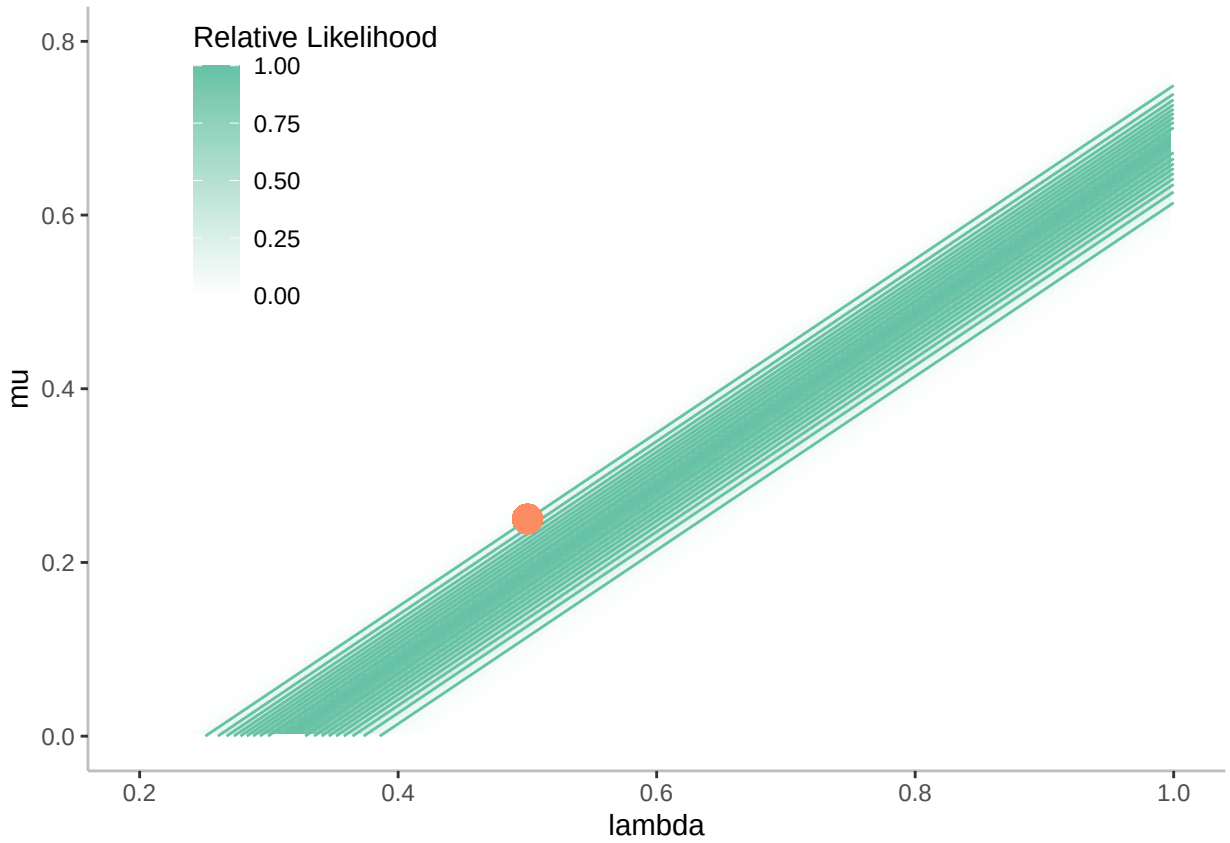
```
#number of generations to run chain
generations <- 5000

#limits on sampling (+ or - this value)
delta <- 0.25

#prepare output matrix
output <- matrix(rep(0, 6 * generations), ncol = 6)

for (i in 1:generations) {
  #modify params (step)
  a_prime <- a + runif(1, -delta, delta)
  b_prime <- b + runif(1, -delta, delta)

  #calculate ratio of likelihoods of new values / old values
  like_odds <-
    likelihood(a_prime, b_prime, simulated_data) / likelihood(a, b, simulated_data)

  #calculate ratio of prior of new values / old values
  prior_odds <- prior(a_prime, b_prime) / prior(a, b)

  #calculate posterior odds
  R <- like_odds * prior_odds

  #randomly draw from uniform distribution
  u <- runif(1)

  #if posterior odds are greater than a random value, keep new values of parameter
  if (u < R) {
    a = a_prime
    b = b_prime
  }

  #calculate posterior (likelihood*prior)
  posterior <- likelihood(a, b, simulated_data) * prior(a, b)

  #store output
  output[i, 1] <- i
  output[i, 2] <- prior(a, b)
  output[i, 3] <- likelihood(a, b, simulated_data)
  output[i, 4] <- posterior
  output[i, 5] <- a
  output[i, 6] <- b
}
```

```
#format output data
output <- data.frame(output)
names(output) <-
  c("iteration", "prior", "likelihood", "posterior", "lambda", "mu")
```

We can then plot the results of our chain. Here we plot the values of a and b over time, showing how they vary dramatically and are highly correlated. However, when we plot $a - b$ we find that we are much better at approximating the value of the difference between these parameters. Even when a and b vary wildly our estimates of $a - b$ remain stable.



Finally, we plot the relative likelihoods across a range of values we are interested in for a and b. We do this by finding the pair of values for a and b that produce the maximum likelihood, and then divide the likelihood of all other combinations of a and b by the maximum likelihood. This produces a likelihood surface that clearly shows the high correlation between a and b, but also that different pairs of values can be equally likely. The true values of a and b are shown as the orange dot. Even though this falls within a region of high likelihood as estimated by our model, it provides no reliable estimate of the absolute values of a and b due to unidentifiability.


Relative Likelihood
1.00
0.75
0.50
0.25
0.00
mu
0.0
0.2
0.4
0.6
0.8
lambda
0.2
0.4
0.6
0.8
1.0

Section 2: Simulations for figures 3 and 4

First we generated values of speciation and extinction rate over time. In Figure 3 we set speciation rate to be constant over time. For Figure 4 we used a slightly more complex a function in which speciation rate increased gradually over time, centred at 100 Ma, while extinction rate remained constant. With these known values of speciation and extinction rate we were able to calculated pulled speciation, diversification and extinction rates using the equations in Louca & Pennell, 2020.

We then used our functions of speciation and extinction rates over time to simulate 50 trees under a birth death model using the function rbdtree() from the R package 'ape'. We generated Lineages-Through-Time (LTT) plots for the resulting trees (Figs. 3e, 4e) and calculated the slopes for each LTT using a loess function. We plotted the values of the slopes at each step (or event) in each LTT to show how the change in these slopes over time is captured by pulled speciation rate (Figs. 3f, 4f).

The github repository https://github.com/ajhelmstetter/lp_mim contains the full code for reproducing figures 3 and 4. This research compendium was made with the help of rcompendium.

Section 3: How does variation in r affect r_p ?