

Hello! Welcome to my beginner notes on making a BiSSE model in R. Here is a first model and the notes I took to remember what I did.

How I structure my notes:

Describe the step

Simple goal of the code

Actual code I ran

Detailed description of the code

Checkpoint before the next step

STEPS:

Step 1: Loading your data and assigning states:

```
tree <- read.tree("/Users/gracecopping/Desktop/RB_bisse_tutorial/anolis.phy")
data <-
read.csv("/Users/gracecopping/Desktop/RB_bisse_tutorial/anolisDataAppended.csv",
row.names = 1)
```

For the data I used I needed to turn the states into 0s and 1s

```
#Make ecomorphs into a state
ecomorph <- data[, "ecomorph"]
names(ecomorph) <- rownames(data)
# Create binary states: 1 for "TG", 0 for all others
states <- ifelse(ecomorph == "TG", 1, 0)
# Name the states based on species
names(states) <- rownames(data)
# Filter states to only include species present in the tree
states <- states[names(states) %in% tree$tip.label]
# Check the result
print(states)
```

Before moving onto the next step check that you have at least three objects: tree, data, states

Step 2: Define your rates and create a transition matrix

```
lambda1 <- 0.10 # Speciation rate for state 1
lambda0 <- 0.05 # Speciation rate for state 0
mu1 <- 0.01     # Extinction rate for state 1
mu0 <- 0.02     # Extinction rate for state 0

#Create a transition matrix for the BiSSE model. Basically the Q matrix
trans_matrix <- matrix( c( lambda1, lambda0, mu1, mu0), nrow = 2, byrow = TRUE)
```

#Assign names to the matrix

```
rownames(trans_matrix) <- c("State1", "State0")
colnames(trans_matrix) <- c("Speciation", "Extinction")
```

Before moving onto the next step write your Q matrix down to help visualize:

Example:

$$Q = \begin{Bmatrix} 0.1 & 0.05 \\ 0.01 & 0.02 \end{Bmatrix}$$

Step 3: Likelihood function

```
# Define a function to calculate the likelihood of the tree given the transition rates
calculate_likelihood <- function(tree, states, trans_matrix) {
  likelihood <- 1.0
  #We start with a likelihood of 1 that gets multiplied by probabilities along the edges
  (connection between two nodes) of the tree.
  # Start the for loop that moves through each edge in the tree
  for (i in 1:nrow(tree$edge)) {
    # ID the nodes
    child <- tree$edge[i, 2] # Child node
    parent <- tree$edge[i, 1] # Parent node

    # Check if parent and child states exist
    if (parent <= length(states) && child <= length(states)) {
      parent_state <- states[parent]
      child_state <- states[child]

      #Checks that both nodes have valid states. Retrieves the states for both the parent
      and child nodes.

      # Calculate the transition probabilities
      if (parent_state == child_state) {
        rate <- trans_matrix[parent_state + 1, 1] # Speciation
      } else {
        rate <- trans_matrix[parent_state + 1, 2] # Extinction
      }

      #Determines whether the transition is a speciation (same state) or extinction
      (different state) and selects the appropriate rate from the transition matrix.

      # Update the likelihood
      #####Updates the overall likelihood by multiplying it by the probability of the
      observed transition along the edge. The expression exp(-rate * tree$edge.length[i])
      calculates the likelihood of surviving through the edge length given the rate.
      likelihood <- likelihood * exp(-rate * tree$edge.length[i])
    }
  }

  return(likelihood)
}
```

Before moving onto the next step write your Equation down to help visualize:

Example:

$$\text{Log } L = \sum_{i=1}^N (-r_i * t_i)$$

N= Sum of edges

i= individual edge

r_i= state given ith edge

T_i = edge length given ith edge

Step 4: Prepare data for optimization and visualization.

4.1 determine negative log-likelihood

#Create a vector to store the four parameters in a single object. Allows you to pass all four parameters together as a single object

```
initial_params <- c(lambda1, lambda0, mu1, mu0)
```

Define a function to minimize (negative log-likelihood)

```
neg_log_likelihood <- function(params) {
```

```
  lambda1 <- params[1]
```

```
  lambda0 <- params[2]
```

```
  mu1 <- params[3]
```

```
  mu0 <- params[4]
```

Parameter Preparation: This part of the function is setting up the specific parameters that will be used later in the likelihood calculation. The function is likely intended to calculate the negative log-likelihood of a model given these parameters. After extracting these parameters, the function would typically proceed to calculate the negative log-likelihood based on the model, the tree structure, and the observed data (e.g., states). A lower value of NLL indicates a better fit of the model to the data.

4.2 Update the transition matrix and get new estimated parameters

Update transition matrix

```
trans_matrix <- matrix(c(lambda1, lambda0, mu1, mu0), nrow = 2)
```

```
return(-calculate_likelihood(tree, states, trans_matrix))
```

```
}
```

#Code calculates the likelihood of observing the states given the phylogenetic tree and the transition matrix. The negative sign in front of the likelihood indicates that you are returning the negative log-likelihood, which is a common approach when performing optimization (as mentioned previously). This code creates a transition matrix based on speciation and extinction rates, calculates the likelihood of the data given the model, and returns the negative of that likelihood for optimization purposes.

Optimize parameters

```
result <- optim(initial_params, neg_log_likelihood)
```

#The primary goal of this line of code is to fit the BiSSE model to your data by finding the best estimates for the speciation and extinction rates (represented by the parameters in initial_params).

Step 5: Results and plotting

```
#Results
estimated_params <- result$par
print(estimated_params)
#Should output your new parameters. Compare these to your original.

#Plot
library(ggplot2)
# Parameter names
param_names <- c("lambda1", "lambda0", "mu1", "mu0")
# Create a bar plot for estimated parameters
barplot(estimated_params, names.arg = param_names, col = "skyblue",
        main = "Estimated Parameters from BiSSE Model",
        ylab = "Rate")
#Now you should have a pretty graphy of your new estimated paramters
```

Practice Questions:

#QUESTION 1

#Write a function in R that implements a simple BiSSE model. The function should take the following inputs:

```
## A phylogenetic tree (tree)
## A vector of states for each tip (states)
## A transition matrix (trans_matrix)
```

#The function should calculate the likelihood of observing the given states on the tree.

```
calculate_bisse_likelihood <- function(tree, states, trans_matrix) {
  # Your code here
}
```

#QUESTION 2

#Define the transition matrix for a BiSSE model where:

```
## The speciation rate for state 1 is 0.2
## The speciation rate for state 0 is 0.5
## The extinction rate for state 1 is 0.03
## The extinction rate for state 0 is 0.06
```

#QUESTION 3

#Describe the likelihood equation used in the BiSSE model. Include the key components and explain how they contribute to the overall likelihood of observing the states on the tree.

