Week 10: Tree Simulations

Due April 13th (not a typo! Next Monday!)

I. General

Type up all of your work in a text editor.

Before you begin, make a new folder in Tasks called Task_08, and save an empty file named task08.r in that folder.

You must also make a separate "Project" folder within Tasks. Inside your Project folder, make "code", "data", and "text" subfolders. By Friday, you should have some data in the data folder, some code to read the data in to `R` in the code folder, and a document with a *single sentence* describing your hypothesis.

When you're **done** with this assignment, turn it in by (1) saving your text document, (2) opening your Terminal or GitBash, (3) navigating to the appropriate directory using `cd` and (4) typing:

`git add -A` (enter)
`git commit -m "Task 08"` (enter)
`git push -u origin master` (enter)

Exponential growth is a pain in the ass, as we're all learning first hand from this COVID19 business.

This R assignment is a lot less guided than the past few. The main goal this week is to understand the variation that can produced under purely neutral processes of diversification and trait evolution. By understanding the large range of expected outcomes of neutral evolutionary processes, hopefully you'll be better able to compare exceptional cases where purely neutral processes are insufficient, while also viewing "just-so" stories predicated on "look at al this diversity!" with a more critical eye. I care mostly about checking over the plots that you make this week, so make sure you include the proper pdf files!

II. Your Code!

Tree simulations, and trait simulations!

- Load the library phytools, which you installed last week.
- QUESTIONS 1-3: Now, we're going to simulate some trees. What I want you to do:
  1. make an empty LIST called "`trees`", an empty vector called "`births`", and an empty vector called "`Fractions`"
  2. open a `for` loop from 1 to 100
  3. draw a random uniform number, store it as the i'th element of "`births`"
  4. draw a second ranodm uniform number, store it as the i'th element of "`Fractions`"
  5. for the i'th element of "`trees`", use the function pbtree to simulate a tree. For pbtree, set: speciation rate = i'th of births, extinction rate = i'th of births multipled by the i'th of `Fractions`, number of tips = 100
  6. close the `for` loop

Hint: to store an object in a particular element of a list, you need to double the square brackets. So `exampleList[[i]] <- mean(...)`, for instance, would save the mean of whatever vector was inside of it in the i'th slot of exampleList.

Hint II: I refer to each round of the `for` loop as the i'th round. If you use a variable other than i for the loop (e.g., run, count, j, k), then you won't use the letter i, but instead use whatever you set to change each go-through the `for` loop

- QUESTION 4: How does the **net diversification** rate for each tree compare to the **log of the total number of tips** for each tree? Hint: make a plot, you don't need to send me a pdf of it
- QUESTION 5: How does the **speciation rate** for each tree compare to the **average branch length** for each tree? Hint: make another plot, you don't need to send me a pdf of it. Hint II: there are many possible ways to get the average branch length for each tree, but one straightforward route is to use a little for loop across the number of trees in your list.
- QUESTION 6: use the `cor()` function to find the exact numerical relationship between **speciation rate** and **average branch length**.
- QUESTION 7: Now we're going to simulate traits on a tree.
  1. choose the largest tree in the set you simulated before.
  2. store it as it's own object (e.g., Tree), then plot it!
  3. make an empty vector called "`rates`", and an empty list called "`traits`"
  4. Now, write another `for` loop from 1 to 100 again.
  5. draw a random uniform number again, store it in the i'th element of "`rates`"
  6. use `fastBM()` and set `tree = Tree`, and `sig2 =` the i'th element of rates, and store it in the i'th element of traits
  7. close the `for` loop

  Think very carefully about the answers to the next 3 questions, and make 100% certain that you're correct and that you nderstand *why* you're correct. Come talk to me if needed.
- QUESTION 8: What is the correlation between the **mean** of traits and the rates? Hint: use `cor()` but also make a plot
- QUESTION 9: What is the correlation between the **variance** of `traits` and `rates`?
- QUESTION 10: What is the correlation between the **first** element of `traits` and the **second** element of `traits`? Is it significant? Why? Make a 2-column matrix out of the first two elements of traits, like this: `traitMat <- cbind(traits[[1]], traits[[4]])`

Extra credit

A big advantage of learning `R` and getting skilled at understanding data, is that you are less reliant on others now! You know more about biology than the vast majority of Americans, and you know more about data analysis than the vast majority of Americans. So you can do (at least some) analyses of your own. Demonstrating the degree to which you can do such things on your own is the crux of the final project.

But in these crazy times, we can play with another dataset. Check out https://covidtracking.com/. Note the "Get the Data" tab. Note that you can find a link to a .csv! Treat that website exactly as you have treated jonsmitchell.com for downloading data in the past `R` assignments, and pull the real data from that real website and make a plot showing some analysis of the COVID-19 rates. Then save that code and make a reddit post. One week later, rerun the exact same code to produce a new, updated plot, and post that in the same thread, comparing them.