Exercises in coalescence theory and drift

Exercise 1A: Simulating a coalescence tree assuming a constant population size

The purpose of this first exercise is to make sure it is clear how a coalescence tree is simulated. To do this let us try to simulate a coalescence tree for five gene copies by hand (+ a little help from R):

1. Start by drawing a node for each of the five gene copies on an invisible horizontal line (with space for drawing a tree above them). Name these nodes 1,2,3,4,5

2. Also, make a list of the node names. You can either do this by hand or you can do it in R by simply (in R) writing

```
nodelist = c(1,2,3,4,5) # make the list and call it nodes
nodelist # print the list
```

3. Sample which two nodes will coalesce first (going back in time) by randomly picking two of the nodes. You can either do this by hand or you can do it in R by typing

```
nodecount = length(nodelist) # save the number of nodes in the variable nodecount
tocoalesce = sample(1:nodecount, size=2) # sample (the index of) 2 nodes in nodelist
nodelist[tocoalesce[1]] # print the name of the first sampled node from the nodelist
nodelist[tocoalesce[2]] # print the name of the second sampled node from the nodelist
```

   If you used R then make sure you understand what the R code does before moving on.

4. Sample the time it takes before these two nodes coalesce (measured from previous coalescence event in units of 2N) by sampling from an exponential distribution with rate equal to nodecount*(nodecount-1)/2 where nodecount is the number of nodes in your node list. Do this in R by typing:

```
nodecount = length(nodelist) # save the number of nodes in the variable nodecount
coalescencerate = nodecount*(nodecount-1)/2 # calculate the coalescent rate
coalescencetime = rexp(1, rate=coalescencerate) # sample from exponential w. that rate
coalescencetime
```

   Make sure you understand what the R code does before moving on.

5. Now draw a node that is the sampled amount of time further up in the tree than the currently highest node (so if the currently highest node is drawn at height T then draw the new one at height T plus the sampled coalescence time) and draw a branch from each of the nodes you sampled in step 3 to this new node indicating that these two nodes coalesce at this time.

6. Next, make an updated list of the nodes that are left by removing the two nodes that coalesced and instead adding the newly drawn node that represents their common ancestor. You can call the new node the next number not used as a name yet (e.g. if this is the first coalescence event you can call it 6, if it is the second coalescence event you can call it 7 etc.). You can either do this by hand or in R. If you want to do it R you can do it as follows:

```
nodelist <- nodelist[-tocoalesce] # remove the two nodes that coalesced
nodelist <- c(nodelist,2*5-length(nodelist)-1) # add the new node
nodelist # print the new list
```

   If you used R then make sure you understand what the R code does before moving on.

7. If you only have one node left in your list of remaining nodes you are done. If not, go back to step 3.

In the end, you should have a tree, which is a simulation of a coalescence tree ☺ Try to do this a couple times until you feel like you know how it is done and understand what is going on (if you after a drawing a few trees still don't understand then feel free to ask for help!)

Exercise 1B: Exploring the basic properties of a standard coalescence tree

Doing this by hand is obviously a bit tedious. So based on the R code snippets you already got I made a function that allows you to do this automatically (it even makes a drawing of the tree). You can use it by typing the following in R:

```
source("/home/moltke/exercisecode/simulatecoalescencetrees.R")
```

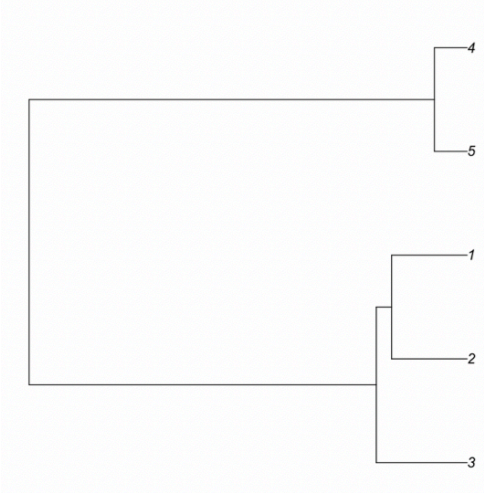
Once you have done this you can simulate and draw trees just like you just did by hand by typing:

```
yourtree <-simtree(5) # simulate tree with 5 nodes
ct<-read.tree(text=yourtree);plot(ct);add.scale.bar(cex = 0.7,col = "red")# draw tree
```

And the code also prints the simulated coalescence times. Do this at least 10 times and based on the results you get answer the following questions (note if you want to keep all the results open at the same time then type `x11()` before each simulation):

1. Which coalescence event takes the longest on average: the first coalescence event, the second, ..., or the last (going back in time)? And which event takes the shortest on average?
2. Is that what you would expect (the mean of an exponential distribution with rate r is 1/r and the coalescence rate when there are x nodes left is x(x-1)/2. So the mean is 2/x(x-1), so for instance for when there are 5 nodes left the mean coalescent time is 2/5(5-1)=0.1)
3. Which coalescence event time seems to vary the most?
4. Is that what you would expect (the variance of an exponential is $1/r^2$)
5. Finally, imagine the following case: a researcher has estimated the structure of a tree for mtDNA from a species sampled in a single location. She obtains a tree looking as follows:

Past **Present time**



Based on the structure of the tree, i.e. two groups of related individuals separated by long branches down to the root of the tree, she concludes that there must be population

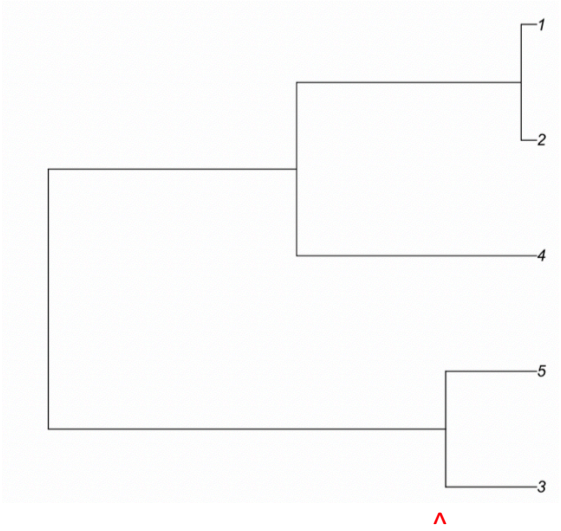
subdivision with two clearly differentiated groups. Based on what you have learned from the simulations, do you agree with this conclusion?

Exercise 1C: Effect of population size on coalescence trees

The time scale in the trees we have been looking at so far is 2N (assuming we are looking at a diploid organism), where N is the population size. With this in mind answer the following questions:

1. How would you expect a tree for five gene copies from a population of size 1000 to differ from a tree for five gene copies from population of size 2000? (if we drew them in real time scale)
2. The tree below is simulated assuming a constant population size:

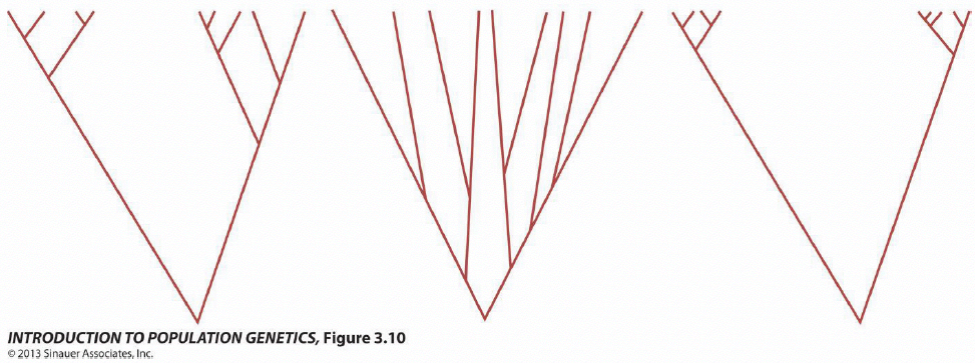
Past **Present time**



^

How would the shape of this tree be different if the population size had not been constant, but instead to population size had been 10 times bigger from the time point indicated with ^ *to* the present? (i.e. the population had increased dramatically in size recently)

3. What if the population size change had not been an increase but instead a decrease?
4. And by the same logic, which of the following trees do you think are simulated under decreasing population size, constant population size and increasing population size?

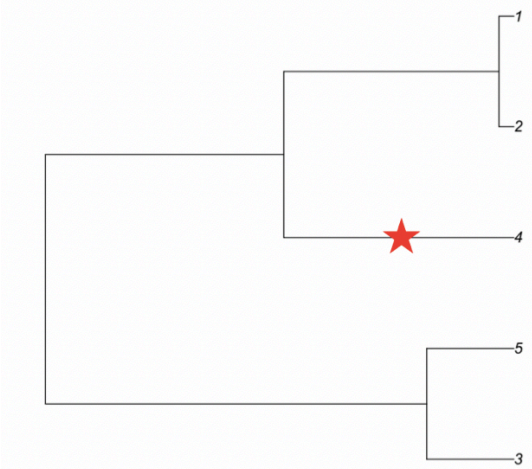


INTRODUCTION TO POPULATION GENETICS, Figure 3.10

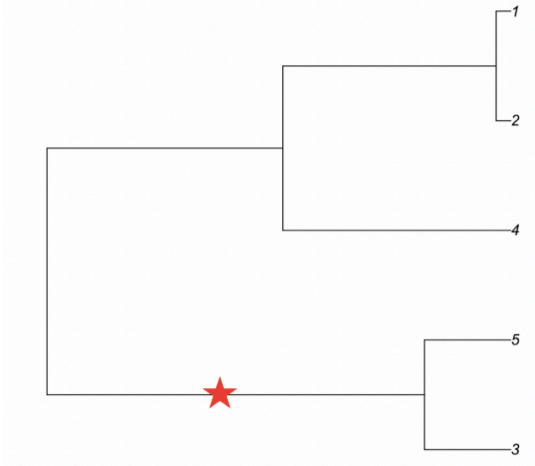
Exercise 1D: The connection between a mutation's location on a tree and the SFS

1. Which categories would a Site Frequency Spectrum (SFS) for five gene copies have? (e.g. singletons, doubletons,...?)

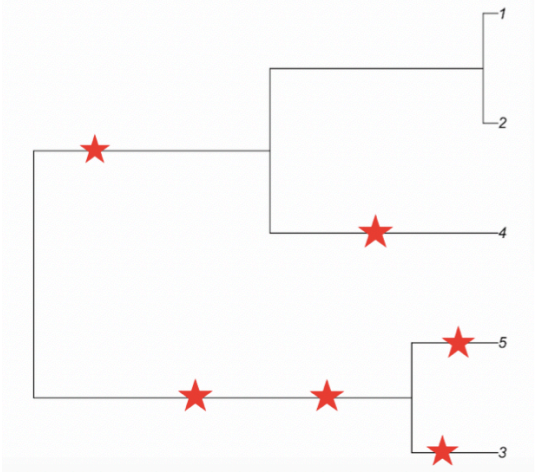
2. Which of these categories would the mutation in this tree (marked by a red star) contribute to:



3. Which category of the SFS would this mutation contribute to in an SFS:



4. Draw a SFS based on this tree:



Exercise 1E: Effect of population size changes on the SFS

Usually the number of mutations that happen on a branch is proportional to the length of the branch. With this and your answers in exercise 1C and 1D in mind answer the following questions:

1. How do you think population growth affects the frequency spectrum?
2. How do you think population size decrease affects the frequency spectrum?