The Coalescent: Task 06

Due February 21st

I. General

Type up all of your work in a text editor. Basically, you should NEVER type things directly into the `R` terminal. Type them into a text editor, then either run them or copy/paste them into `R`.

Before you begin, make a new folder in Tasks called Task_06, and save an empty file named task03.r in that folder.

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 "yourname Task 06"` (enter)

`git push -u origin master` (enter)

II. learnPopGen

I want you to use the `R` package `learnPopGen` to simulate coalescence in populations. The function `coalescent.plot()` will be most helpful. Review the help pages for that package to do this.

I would like for you to save three separate PDFs. Each PDF should be a single page long and contain only the final plot from one of three coalescent simulations. In addition, please answer the following questions:

1. How many alleles does each simulation begin with? How do you modify that?
2. On average, how many generations does it take for one allele to go to fixation?
3. What's the average number of offspring each haploid individual has? What's the variance in that number?
4. What role does fitness play in these simulations?
5. Is the most recent common ancestor for the focal locus typically alive in generation 0?

III. coala

`learnPopGen` is very useful for visualizing straightforward processes. But I want you to use the package `coala` now, as you can simulate a truly staggering array of different scenarios and examine how changes in those scenarios impact things like genetic diversity.

You'll need to install and load up the libraries `coala` and `phytools`. `coala` uses a strange syntax where you add functions (features, "`feat_`") to one another to build a model, and tell it what kind of data you want to have output (summary statistics, "`sumstat_`"). Then use the `simulate` function on that built model to get your output. Use the help function & google to see the feature & sumstat options.

```
# Setting up a model.
# We'll use a sample of 5 individuals from 1 population.
#   Each indivdual will have 10 loci.
#   Each locus will be 500 base pairs long.
#   And there will be two copies per individual (diploid individuals).
#  We'll also add some features.
#   Specifically, mutation and recombination, at fixed rates.
# Finally, we'll summarize the output by showing pedigrees for each
   locus, and the overall diversity
model <- coal_model(sample_size = 5, loci_number = 10, loci_length =
   500, ploidy = 2) +
    feat_mutation(10) +
    feat_recombination(10) +
    sumstat_trees() +
    sumstat_nucleotide_div()

# Actually *run* the simulation. We can run more than one simualtion
   for these same parameters, if we want, by chaning nsim.
stats <- simulate(model, nsim = 1)

# Each locus has a measure of genetic diversity called ''pi''. pi is a
   standard measure. It's the average number of differences at a locus
   between any two individuals.
Diversity <- stats$pi

# Looking at the Diversity object, are all the numbers the same? What
   causes the differences?

# Each SNP in each locus has its own ancestry tree. We'll look at these
Nloci <- length(stats$trees)

# First, let's just look at the first SNP for the first locus.
t1 <- read.tree(text=stats$trees[[1]][1])
plot(t1)
axisPhylo()

# Each copy of a given locus around today is given a number. The tree
   pattern shows the pedigree connecting the copies of this locus
   present today (t = 0).
# Question 6. Why does the number of tips NOT match the number of
   individuals you simulated?
# We can find out the age the most recent ancestor for this SNP on this
    locus for these individuals lived by looking at how deep in time
   the tree goes.
Age1 <- max(nodeHeights(t1))

# Now let's look at the first SNP of the second locus
t2 <- read.tree(text=stats$trees[[2]][1])
```

```
plot(t2)
axisPhylo()

# How far back is the most recent common ancestor for this SNP? Is it
    the same age as for the first SNP?

# Question 7. Do they match? Let's plot them next to each other.
par(mfrow=c(1,2))
plot(t1)
axisPhylo()
plot(t2)
axisPhylo()

# We can also compare the trees quite explictly, and see how the
    patterns of descent and timing differ between these two SNPs from
    two different loci.
compare.chronograms(t1, t2)

# Now, make more comparisons! You should understand WHY you see the
    patterns/results that you do when you make graphs. Here's an example
    :
t1_1 <- read.tree(text=stats$trees[[1]][1])
t1_2 <- read.tree(text=stats$trees[[1]][2])
compare.chronograms(t1_1, t1_2)

# There is a lot of benefit to comparing individual SNPs to one another
    . I do not recommend continuing on until you fully understand the
    plots you just made, and I encourage you to come talk to me about
    them if you aren't sure if you know why you see what you get when
    you change values around.
# But we can also leverage the power of R to compare all of the SNPs
    from all of the loci all at once.
for (locus in 1:Nloci) {
  ntrees <- length(stats$trees[[locus]])
  for (n in 1:ntrees) {
    if (locus == 1 && n == 1) {
      outPhy <- read.tree(text=stats$trees[[locus]][n])
    }
    else {
      outPhy <- ape:::c.phylo(outPhy, read.tree(text=stats$trees[[locus
          ]][n]))
    }
  }
}

# Now, we'll plot all of the trees all at once.
par(mfrow=c(1,1))
densityTree(outPhy)
```

```
131
132 # This is an excellent study opportunity.
133 # 1. Look at this set of trees.
134 # 2. Go up and change ONE thing about the model (e.g., recombination
135    rate).
136 # 3. Predict, based on what you changed in the model, how this final
137    plot will be different.
138 # 4. Rerun the model & this code. Is it different in the way you
139    predicted?
140
141 # finally, for your own awareness in studying, you can specify very
142    complicated models. Here, mutation rate varies in each of 40
143    simulations
144 model3 <- coal_model(10, 50) +
145     feat_mutation(par_prior("theta", sample.int(100, 1))) +
146     sumstat_nucleotide_div()
147 stats <- simulate(model3, nsim = 40)
148
149 mean_pi <- sapply(stats, function(x) mean(x$pi))
150 theta <- sapply(stats, function(x) x$pars[["theta"]])
151
152 # Plot mean_pi and theta against one another and fit a regression line.
153 # With coala, you can simulate multiple populations, selection,
154    population size changes over time, etc., and look at the outcome of
155    those processes on genetic diversity.
```

IV. Extra Credit

Use `coala` to simulate a set of two populations of different sizes, with different degrees of selection, that all undergo population size changes of different magnitudes at different times, with asymmetric migration, and do this many times so that you can map out, on average, how the two populations compare in nucleotide diversity (pi, π).