Assignment 10: Wright-Fisher Model

You are a graduate student working on a gene to make “superyeast” in Professor Sue Dohyphal’s lab. You have cloned a gene from a Martian strain with a mutation that you believe dramatically increases fitness. You decide to call this allele Bio5488. You insert Bio5488 into a chromosome of a lab strain. When you take one cell of the Bio5488 lab strain and place it in a culture of 99 wild-type cells, the Bio5488 allele fixes 90% of the time after 1000 generations. Being an expert in population genetics (because of your 1st year genomics class), you decide to estimate the approximate fitness of the Bio5488 allele. To do this, you’re going to use the Wright-Fisher model.

Question 1
What assumptions does the Wright-Fisher model make?

Part 1

In this assignment, you will calculate the lowest integer fitness value that gives 90% fixation rate for a mutant allele. Before you start to code, WRITE OUT what you want to do (including if you want to use dictionaries, lists or whatever) and comment out these lines. Here is an outline of the Wright-Fisher model:

Step 1: Initialize a population where each individual gets 2 alleles. In our case, we’ll start with everyone homozygous for the non-Bio5488/wild-type allele except one heterozygous individual .

Step 2:
Start at the first individual in the next generation, randomly pick an individual from the population based on fitness. So if the fitness for individual A is 3 and individual B is 1, then it should be 3 times more likely to pick A. Once you pick an individual, you need to randomly pick only one of its alleles. Next, pick the first individual in the next generation’s second allele separately (using different random numbers). To do these steps, you’re going to need to make random numbers. Hint: use the `random` module. Repeat this process for the remaining individuals in the next generation.

Step 3: Check for fixation of either allele. If there is fixation by the max generation time, record which allele fixed. Then start over with the original starting population.

Step 4: Repeat steps 1-3 100 times and give the percent of the time that the Bio5488 allele fixed.

Write a script called `wrightfisher.py` to perform the steps outlined above. The input parameters to the script will be population size, maximum # of generations, Bio5488 fitness, and model, i.e., dominant or recessive. For this part of the assignment, the model will be “dominant.” In your script, assume the Bio5488 allele is completely dominant, meaning that fitness(Het) =

fitness(Homozygous Bio5488) and fitness(Homozygous non-Bio5488)=1. For the recessive model, fitness(Het) = fitness(Homozygous non-Bio5488)=1. Output the percent of simulations where the Bio5488 allele fixed or were lost.

The usage of your script should be:

```
$ python3 wrightfisher.py <population size> <max # generations> <fitness value> <model>
```

Question 2

Given a population size of 100, a max # of generations of 1000 use your `wrightfisher.py` script to model Bio5488 as a dominant allele. What is the minimum fitness value that gives 90% fixation of the Bio5488 allele? Is this what you expected?

Part 2

Now modify your `wrightfisher.py` script to model Bio5488 as a recessive allele when the model parameter is “recessive”.

Question 3

Given a population size of 100, a max # of generations of 1000, and a“recessive” model, what is the minimum fitness value that gives 90% fixation of the Bio5488 allele? How does this behavior differ from the dominant model and why?

Part 3

Now it’s time to visualize your data. Modify `wrightfisher.py` to plot the allele and genotype frequencies (Y-axis) against the generation number (X-axis). Only create the plot for the first simulation in which Bio5488 goes to fixation. Include a title, x/y labels, a legend, and also do different line styles to differentiate between the genotype and allele frequencies.

Rerun `wrightfisher.py` with a population size of 100, a max # of generations of 1000, and Bio5488 modeled as a **dominant** allele to generate the figure. Save your output to `allelic_frequency_vs_generations_dominant.png`.

Finally, run `wrightfisher.py` with the same parameters except for model, which will be **recessive.** Name the new figure `allelic_frequency_vs_generations_recessive.png`.

What to turn in

- A completed `README.txt`
- A commented `wrightfisher.py`
- Figures appropriately scaled with labelled axes and informative titles:
  - `allelic_frequency_vs_generations_dominant.png`
  - `allelic_frequency_vs_generations_recessive.png`

Note from TAs: After you have placed your script in the submission folder, ensure that it will still run correctly with either model.