BIM 105
Fall 2024

Lab 6: Genetic Drift and the Wright-Fisher Model

Instructions: Read the prompts and use MATLAB to answer the questions. Submit your solution as a single script that outputs all answers to the Command Window. A template script is available on Canvas.

Part 1. Genetic Drift

Genetic drift is defined as changes in the proportion of genetic variants (alleles) within a certain population due to random sampling. For instance, consider a population of 10 people. Humans are diploid organisms, meaning that each individual has two copies of each gene. Thus, there are 20 copies of each gene within the population of 10 people. Imagine, for one particular gene, there are 8 copies of a specific allele in the population's gene pool. As the population evolves, there won't necessarily be exactly 8 copies of that allele in the next generation. This is because the alleles present in the offspring are a *sample* of those present in the parents, and as such, *chance* has a role in determining which alleles are passed to the next generation.

In the absence of natural selection, recombination, and mutation, genetic drift can be modeled using simple laws of combinations and probabilities.

Reconsider the example above with 20 copies of a gene in a population, 8 of which are a certain allele (which we'll call allele $\boldsymbol{A}$). Under the assumption that there are only two allelic variants, the remaining 12 copies must be the other allele (which we'll call $\boldsymbol{B}$). Assuming that generations do not overlap, the allelic composition of the next generation can be considered as a random, independent sampling of alleles from the current gene pool.

Question 1: If you sample one allele from the initial gene pool, what's the probability that it's allele $\boldsymbol{A}$?

Question 2: If you sample two alleles (with replacement) from the initial gene pool, what's the probability that both alleles are allele $\boldsymbol{B}$?

Assuming that successive populations are the same size, the general formula for the probability of sampling k copies of allele $\boldsymbol{A}$ in the next generation is given by the binomial distribution. This formula considers the probability of sampling exactly k copies of allele $\boldsymbol{A}$ as well as the total number of possible ways to sample k copies of allele $\boldsymbol{A}$ given the gene pool is of size $2N$, where N is the number of individuals. With $p = i/2N$, the formula is:

$$\Pr(k \text{ copies of allele } \boldsymbol{A} \text{ in next generation}) = \binom{2N}{k} p^k (1-p)^{2N-k}$$

Recall that $\binom{n}{k}$ is the binomial coefficient (the number of combinations of n things taken k at a time). The built-in function **`nchoosek(n, k)`** can be used to compute the number in MATLAB. Plugging in the numbers for our example, the formula becomes

$$\Pr(k \text{ copies of allele } \boldsymbol{A} \text{ in next generation}) = \binom{20}{k}\left(\frac{8}{20}\right)^{k}\left(\frac{12}{20}\right)^{20-k}$$

Question 3: Using this formula and our initial example, what is the probability of the next generation having *exactly* 8 copies of allele $\boldsymbol{A}$?

Question 4: What is the probability of the next generation not having exactly 8 copies of allele $\boldsymbol{A}$?

Suppose we wanted to calculate the probability of the number of copies of allele $\boldsymbol{A}$ *decreasing* in the next generation. To do this, we need to consider the probabilities of any outcome in which $k \leq 7$ and then sum these values. We can perform this computation in MATLAB by calling **`binocdf(7, 20, 0.4)`**, which computes the cumulative probability density of $k \leq 7$. This results in a probability of 0.416 for the number of copies of allele $\boldsymbol{A}$ to decrease.

Question 5: Compute the probability of the number of copies of $\boldsymbol{A}$ *increasing*. Is the number of copies more likely to increase or decrease in the next generation?

Part 2. Simulating Populations

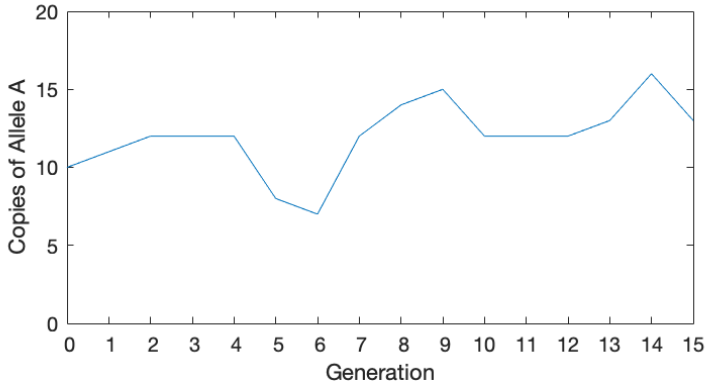
Simulations can be a powerful tool to confirm theory and intuition. In this section, we will use MATLAB to simulate the allelic resampling that underlies genetic drift.

Genetic drift occurs when the example illustrated in Part 1 is repeated across successive generations. For instance, in a population of 10 individuals with 8 copies of allele $\boldsymbol{A}$, the next generation will have somewhere between 0 and 20 copies of the allele (as the alleles are randomly resampled). This process can be repeated for the next generation, where the outcomes are now dependent on how the number of allelic copies in the first generation are resampled. This resampling process can be coded in MATLAB to produce genetic drift simulations. The built-in function **`randsample`** Is useful for this, although for this lab we provide the necessary simulation code in a function **`sim_wright_fisher(N, N0, N_generations, N_simulations)`**.

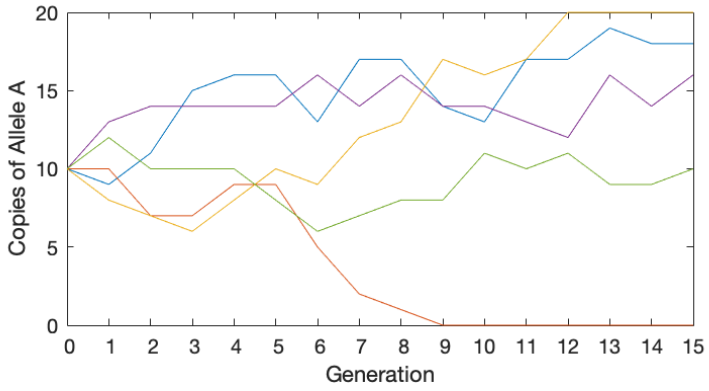
Download the function **`sim_wright_fisher.m`** from Canvas and add it to your active MATLAB directory. As inputs, the function takes in an allele pool size ($2N$ for haploid organisms), an initial number of alleles of interest, the number of generations to simulate, and the number of simulations to perform. For example, if you wanted to perform one simulation of 15 generations with the Wright-Fisher model in a population of 10 individuals where 50% of the alleles at the start are allele A, you would call:

```
chains = sim_wright_fisher(20, 10, 15, 1);
```

The trajectory of the allele counts over time could be visualized with the **`plot`** command, which should give something like this:



The results of one simulation don't tell us a lot about the behavior of the system, however. If we instead use 5 simulations by calling **`sim_wright_fisher(20, 10, 15, 5)`**, the plot will look something like:



We can see that each simulation results in a different trajectory. One trajectory reached 20 copies of the allele (12th generation) and stayed there; one reached 0 copies of the allele (9th generation) and stayed there; the other three simulations had not yet fixed by 15 generations.

Question 6: Run 100 simulations of genetic drift and plot the trajectories of allelic copies as is done above.

Question 7: Plot the distribution of allelic counts in the first generation using the **`histogram`** function. (*Hint: the allelic counts of the first generation are stored in* `chains(2,:)`).

Question 8: Plot the distribution of allelic counts in the 15th generation using the **`histogram`** function. About how many of the simulations reached a count or 0 or 20?

Part 3. Wright-Fisher Model of Genetic Drift

Consider the sequence of allelic frequencies for a gene in a given population as a Markov chain. Again, assume that only two allelic variants exist. The state of the system at each generation can be characterized by the number of copies of one of the alleles in the population.

Let's say that in a certain population of N individuals, there are i copies of an allele. The number of copies in the next generation (which can be denoted k) may subsequently assume any integer from $\{0 \dots 2N\}$.The probability of *transitioning* to each of those values in the next generation is given by the probability distribution in Part 1 (the binomial distribution). As such, we can construct the *transition matrix* of the system by computing for each value of i the probability of sampling k alleles. More formally, each element of the transition matrix is given by the formula (note that $i/2N = p$):

$$P_{ik} = \binom{2N}{k}\left(\frac{i}{2N}\right)^{k}\left(1-\frac{i}{2N}\right)^{2N-k}$$

Let's construct the transition matrix for a simple system of 4 individuals. In MATLAB, we construct the matrix by iterating through each row, as shown below:

```
N = 4;
P = zeros(2*N+1, 2*N+1);

for i = 0:2*N
    P(i+1,:) = binopdf(0:2*N, 2*N, i/(2*N));
end

display(P);
```

The resultant output is:

```
P =

    1.0000         0         0         0         0         0         0         0         0
    0.3436    0.3927    0.1963    0.0561    0.0100    0.0011    0.0001    0.0000    0.0000
    0.1001    0.2670    0.3115    0.2076    0.0865    0.0231    0.0038    0.0004    0.0000
    0.0233    0.1118    0.2347    0.2816    0.2112    0.1014    0.0304    0.0052    0.0004
    0.0039    0.0312    0.1094    0.2188    0.2734    0.2188    0.1094    0.0312    0.0039
    0.0004    0.0052    0.0304    0.1014    0.2112    0.2816    0.2347    0.1118    0.0233
    0.0000    0.0004    0.0038    0.0231    0.0865    0.2076    0.3115    0.2670    0.1001
    0.0000    0.0000    0.0001    0.0011    0.0100    0.0561    0.1963    0.3927    0.3436
         0         0         0         0         0         0         0         0    1.0000
```

Now consider what happens if the number of copies for an allele reaches zero. To which states can this system now transition to? Because there are no copies of the allele to sample into the next generation, the system is now *fixed* at this *recurrent* state. No matter how many times you sample the next generation, the number of copies of this allele will always remain at zero. This feature of our system is captured by the first and last rows of the transition matrix, which show the only possible outcome for resampling zero and 8 copies of an allele is zero and eight copies, respectively.

Having the transition matrix makes it easy to characterize how a given system will evolve over generations. In order to do that, we need to define a vector, $\Pi(0)$, that tells us the state of the system at the first generation. Using our example where $N = 4$ from above, we may want to study systems that start with 4 copies of the allele ($p = 0.5$). In this case, $\Pi(0)$ would be:

$$\Pi(0) = [0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0]$$

Then, to compute the probabilities of the system assuming each of the possible values k in the next generation, we use matrix-multiplication:

$$\Pi(1) = \Pi(0)P$$

Having already computed the transition matrix, P, from above, we get:

```
pi0 = zeros(1, 2*N+1);
pi0(N+1) = 1; % Assign the 5th element to be one (corresponding to four copies)
pi1 = pi0*P; % Matrix multiplication
display(pi1);

pi1 =

    0.0039    0.0312    0.1094    0.2188    0.2734    0.2188    0.1094    0.0312    0.0039
```

$\Pi(1)$ gives us the probabilities of our system transitioning to k copies of the allele in the next generation. The first element of $\Pi(1)$ corresponds to the probability of transitioning to zero copies; the second element of $\Pi(1)$ corresponds to the probability of transitioning to one copy; and so on. Note that the probabilities display symmetry about the point $k = 4$.

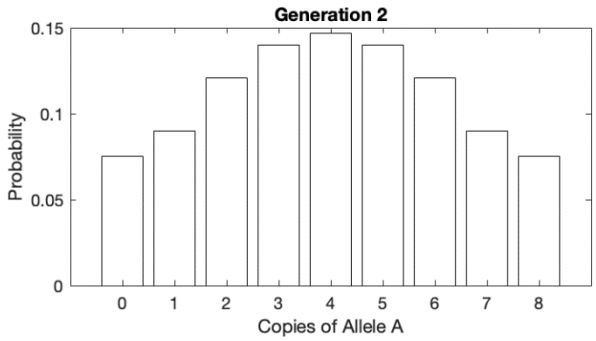
The matrix multiplication can then be repeated to compute the probability distribution over the range of allele copies at any generation t. This gives us the **Chapman-Kolmogorov Equation**:

$$\Pi(t) = \Pi(0)P^t$$

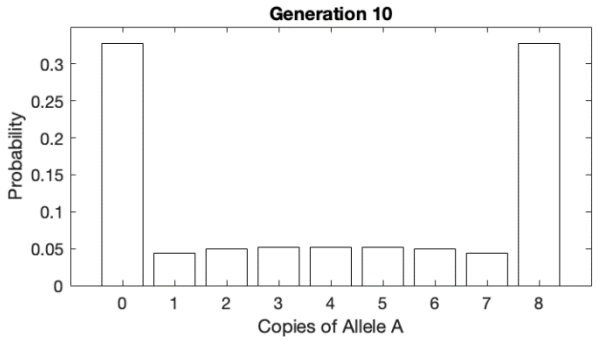
In MATLAB, we can use this equation to compute the ensemble properties of trajectories in our system, at any generation. For example, we could compute the distribution of allelic counts at the *second* generation, $\Pi(2)$, and plot the values using the **`bar`** function.

```
pi2 = pi0*P^2;
bar(pi2)
```

Adding some labels then produces the plot:



We can repeat the process for a later time point, say, the 10th generation. The distribution then looks like:



We can see that in the second generation it's unlikely to have transitioned to zero copies. Most of the probability is rather concentrated nearer to the initial number of copies, which is four. As time goes on, however, the proportion of the probability density captured by recurrent states (zero and eight copies) gradually increases to a large proportion by the 10th generation.

Question 9: In the example above, we started with 4 copies of allele $\boldsymbol{A}$ in the gene pool of size 8. Now consider a population of 10 individuals (gene pool of size 20) starting with 10 copies of allele $\boldsymbol{A}$ (i.e., the same parameters that we used in the simulations from Part 2).

9a: Generate the transition matrix for this system according to the Wright-Fisher model.

9b: What is the initial distribution vector, $\Pi(0)$, for this system?

9c: Use the Chapman-Kolmogorov equation to obtain the distribution of allelic copies in the second generation, and plot the distribution using `bar`.

9d: Use the Chapman-Kolmogorov equation to obtain the distribution of allelic copies in the 15th generation, and plot the distribution using `bar`.

9e: Compare the results of Questions 9c and 9d to the results from your simulations in Questions 7 and 8, respectively. How well do the simulations match the theoretical distribution?

Part 4. Comparing the model to real data

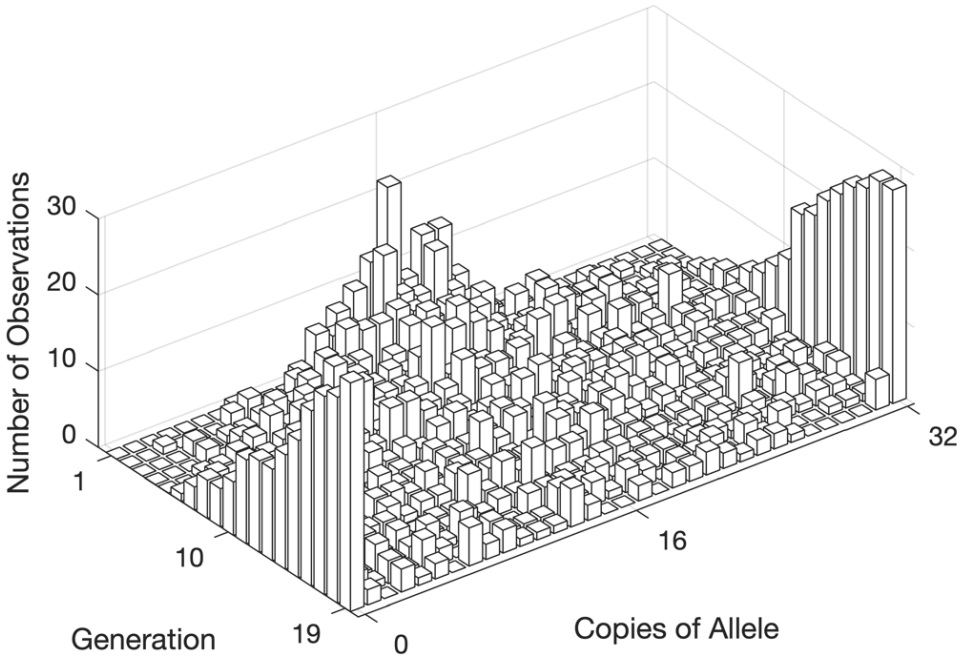
In 1956, Peter Buri used fruit flies (*Drosophila melanogaster*) to test the effects of genetic drift in the lab. He created 107 populations which contained 16 flies each. Each population started with exactly half of the gene pool being comprised by a certain allele (in this case, eye color was the phenotype). The populations were then allowed to reproduce over a course of 19 generations. At each generation, the allelic frequencies were counted.

The final counts of allelic frequency at each generation across all 107 populations in the experiment are contained in a **`.csv`** file called **`buri_data.csv`**. Download this file and save it to your working MATLAB directory.

To open the data, we can use **`csvread`**. This reads the file into a matrix format. Once the data is read into MATLAB, it can be visualized a number of different ways. The code below, for instance, reads the data and plots the allelic counts as a 2D bar graph over generations and allelic copies, similarly to our plots earlier in the lab.

```
buri_data = csvread('buri_data.csv');
bar3(buri_data(2:end,:), 'w')
xlabel('Copies of Allele')
yticks(1:length(buri_data(:,1))-1)
xticks(1:33)
xticklabels(cellstr(num2str((0:32)')))
ylabel('Generation')
zlabel('Number of Observations')
```

The output looks like this:



Question 10: Download and plot Peter Buri's data as shown above. Are the results of the experiment consistent with the theory of genetic drift?

Question 11: In Part 2, we used simulations to model genetic drift computationally. Perform a reproduction of Buri's experiment (107 simulations; 32 allelic copies; 16 allelic copies "of interest" at start; 19 generations) and plot the results. How similar are the simulations to the real data? What do you notice is different?

(*Hint 1: The function* **`sim_wright_fisher`** *can return a second output variable which contains the counts of each possible number of copies at each generation. Use the* **`bar3`** *command with these counts as its input*.)

(*Hint 2: Only plot the first generation onwards* (`counts(2:end,:)`*, as the "zero-th" generation will just be a large spike at k = 16.)*