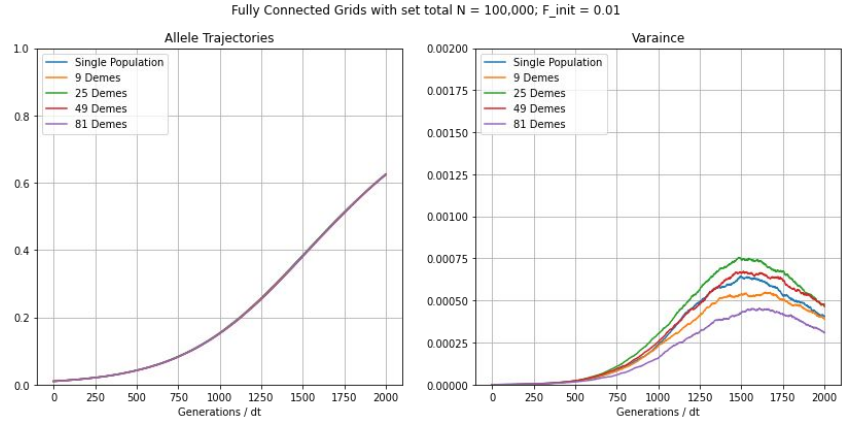
Rotation Week 8 Updates

Rowan Hart
Ecology and Evolution
University of Chicago

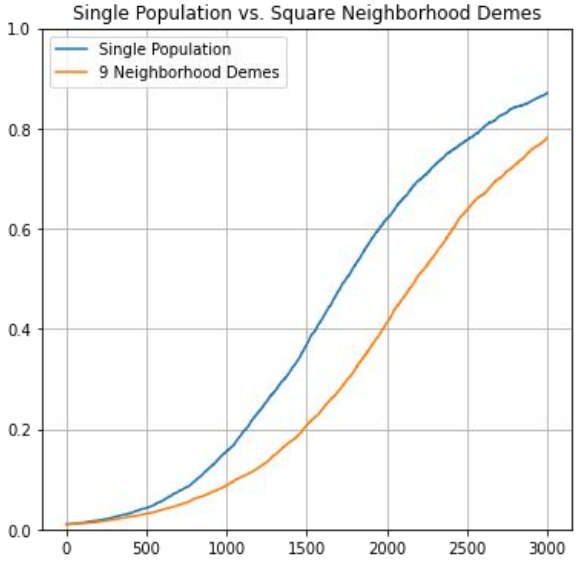


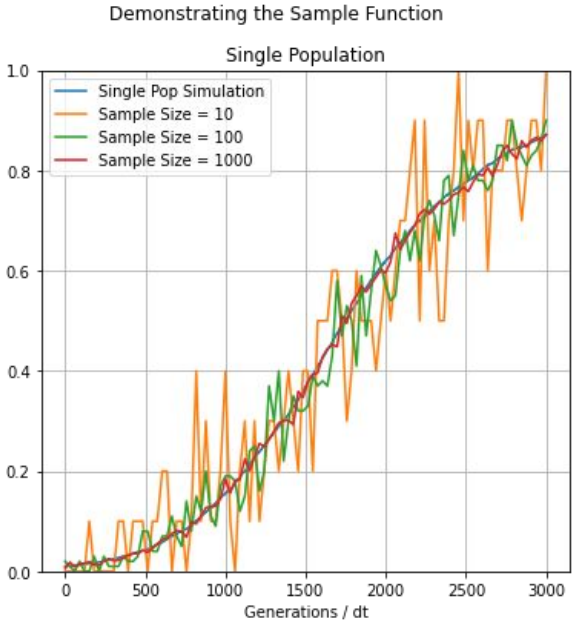
Sampling Data from Simulations

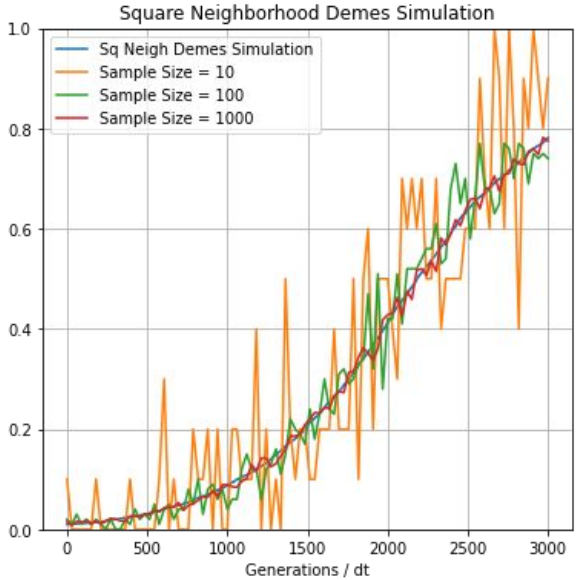
```
########## Random Sampling From Simulations ##########

# Function to Sample From Simulation
def sample_sim(sim, num_samples, samp_size):
  sample_times = np.linspace(0, len(sim) - 1, num_samples, dtype=int)
  freqs = np.zeros((num_samples, samp_size))
  inds = np.zeros((num_samples, samp_size))
  samps = np.zeros(num_samples)
  for i in np.arange(num_samples):
    for j in np.arange(samp_size):
      freqs[i,j] = np.random.choice(sim[sample_times[i]])
      inds[i,j] = np.random.binomial(1, freqs[i,j])
    samps[i] = sum(inds[i]) / samp_size
  return (sample_times, samps)
```

Sampling Data from Simulations







What are the probabilities of **allele loss**?
Differences between Normal and Binomial Distributions

Binomial Distribution:

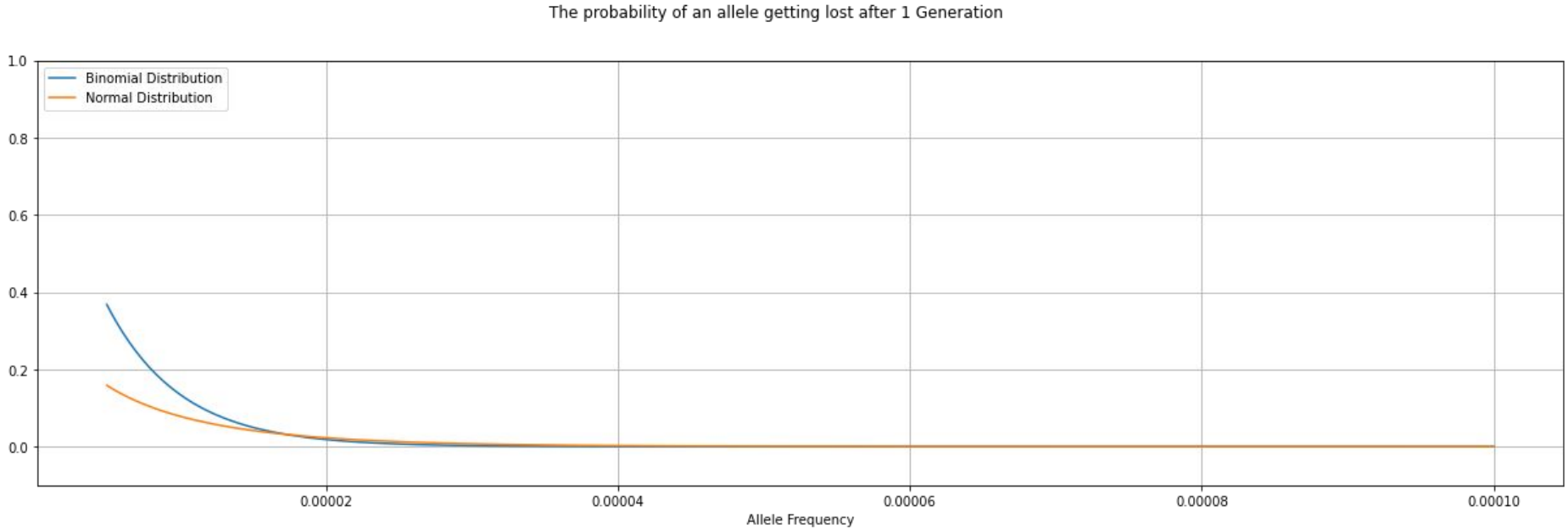
CDF(**X** = 0, 2*N, freq)

Normal Distribution:

CDF(**X** = 0, mean=freq,
std=(1/2N)*(1-freq)*freq)

Parameters:
N_total = 100,000; No Selection
- Test along a number of allele frequencies

What are the probabilities of **allele loss**?
Differences between Normal and Binomial Distributions



What are the probabilities of **allele loss**?
Differences between Normal and Binomial Distributions

Binomial Distribution:

Ni = N / d
[CDF(**X** = 0, 2*Ni, freq)]^d

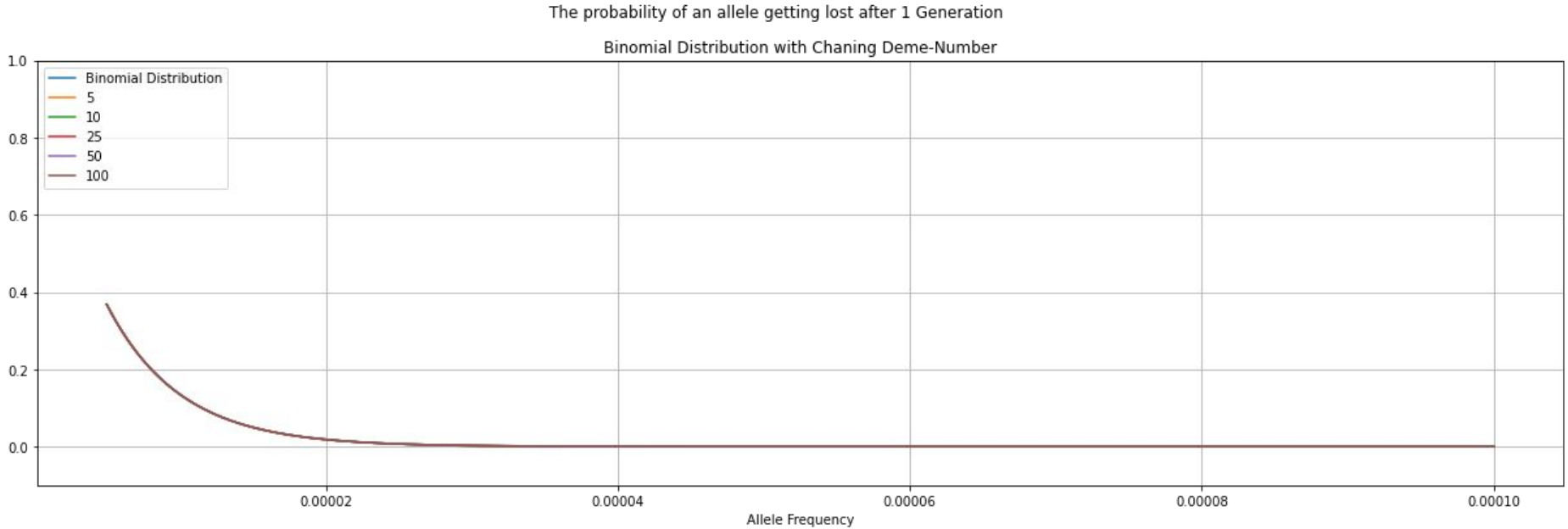
Normal Distribution:

Ni = N / d
[CDF(**X** = 0, mean=freq, std=(1/2Ni)*(1-freq)*freq)]^d

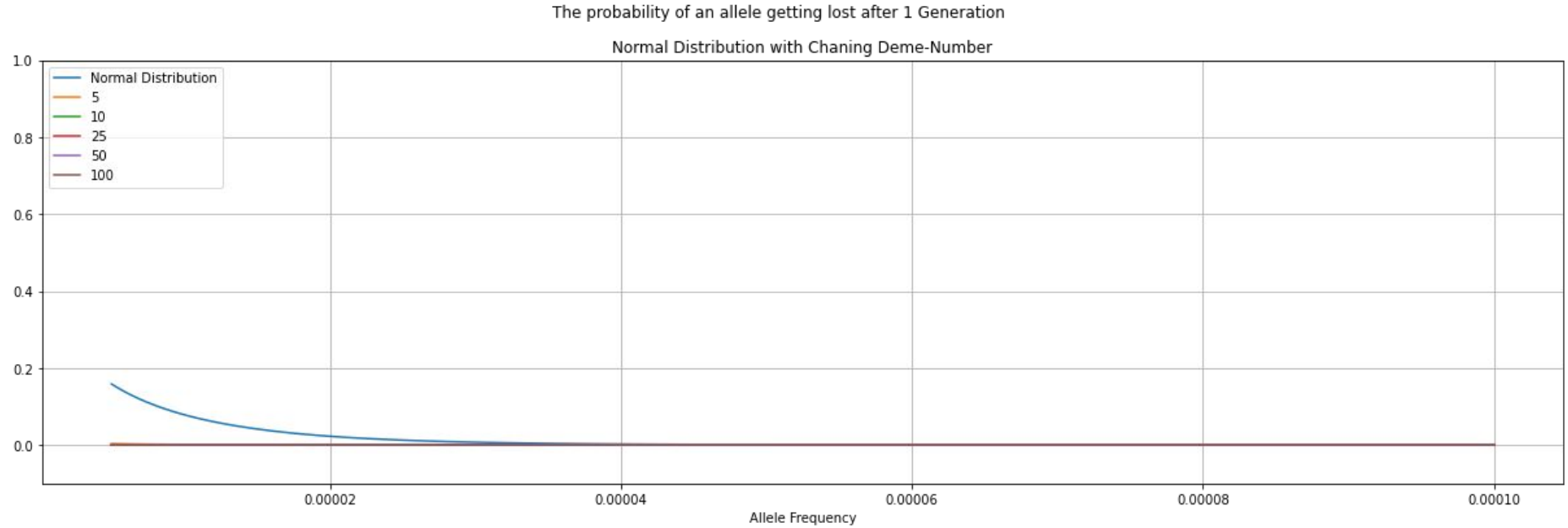
Parameters:
N_total = 100,000; No Selection

- Test along a number of allele frequencies
- Test along d = [1, 5, 10, 25, 50, 100]

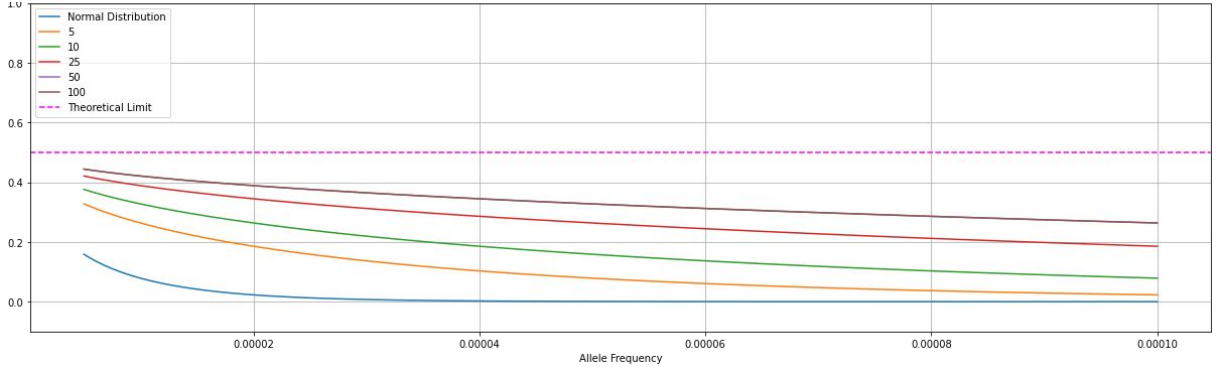
What are the probabilities of **allele loss**?
Differences between Normal and Binomial Distributions

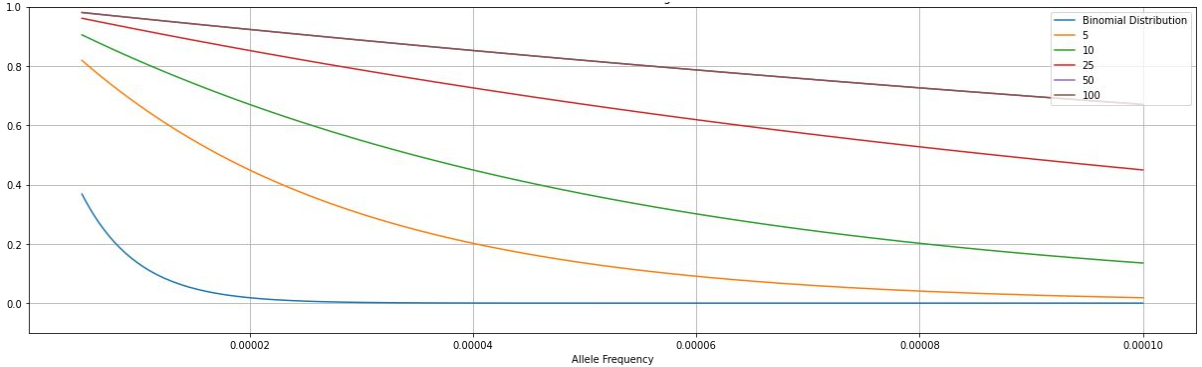


What are the probabilities of **allele loss**?
Differences between Normal and Binomial Distributions



Effect of ‘N_i’ on Probability of Allele Loss

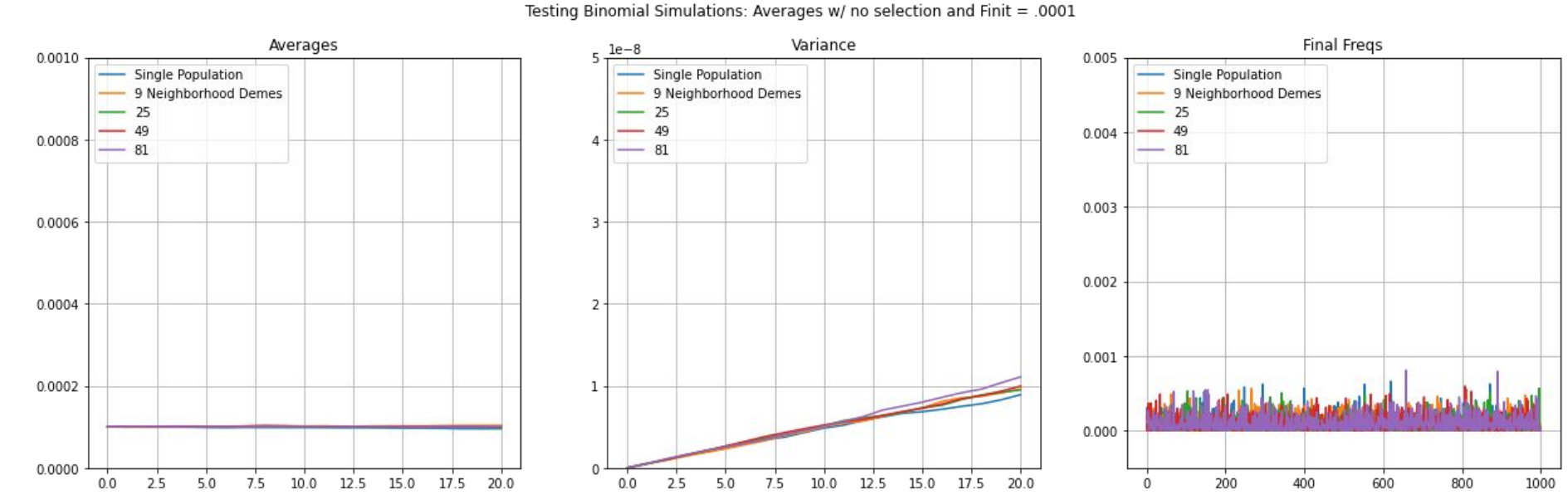




Comparing **Binomial Distribution** to Gaussian distribution

Parameters:
N_total = 100,000; F_init = 0.00005; dt = 1; t = 20 generations; 1,000 iterations

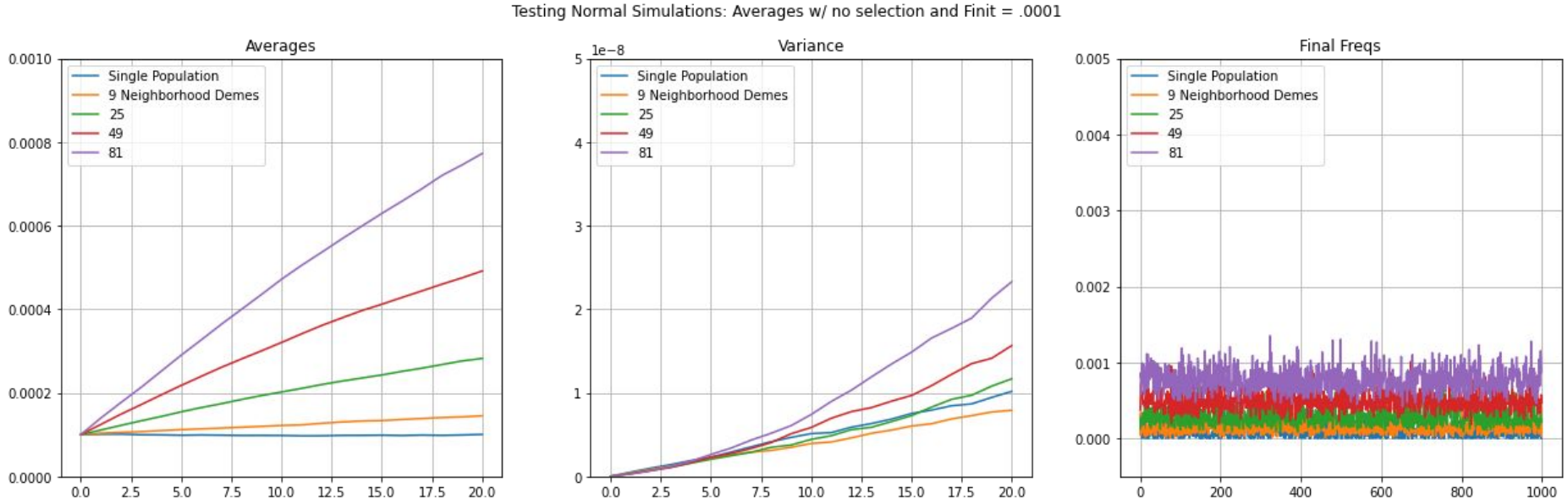
- no selection
- fully connected demes



Comparing Binomial Distribution to **Gaussian distribution**

Parameters:
N_total = 100,000; F_init = 0.0001; dt = 1; t = 20 generations; 1,000 iterations

- no selection
- fully connected demes



Comparing **Gaussian distribution** to Binomial Distribution

Parameters:
N_total = 100,000; F_init = 0.00005; dt = 1; t = 20 generations

- no selection
- fully connected demes

Binomial Alleles Lost
(of 1,000 iterations to 20 gen)

```
print(sum(results_bin[0][-1] < (10 ** -15)))
print(sum(results_bin[1][-1] < (10 ** -15)))
print(sum(results_bin[2][-1] < (10 ** -15)))
print(sum(results_bin[3][-1] < (10 ** -15)))
print(sum(results_bin[4][-1] < (10 ** -15)))
```

```
634
615
595
643
638
```

Normal Alleles Lost
(of 1,000 iterations to 20 gen)

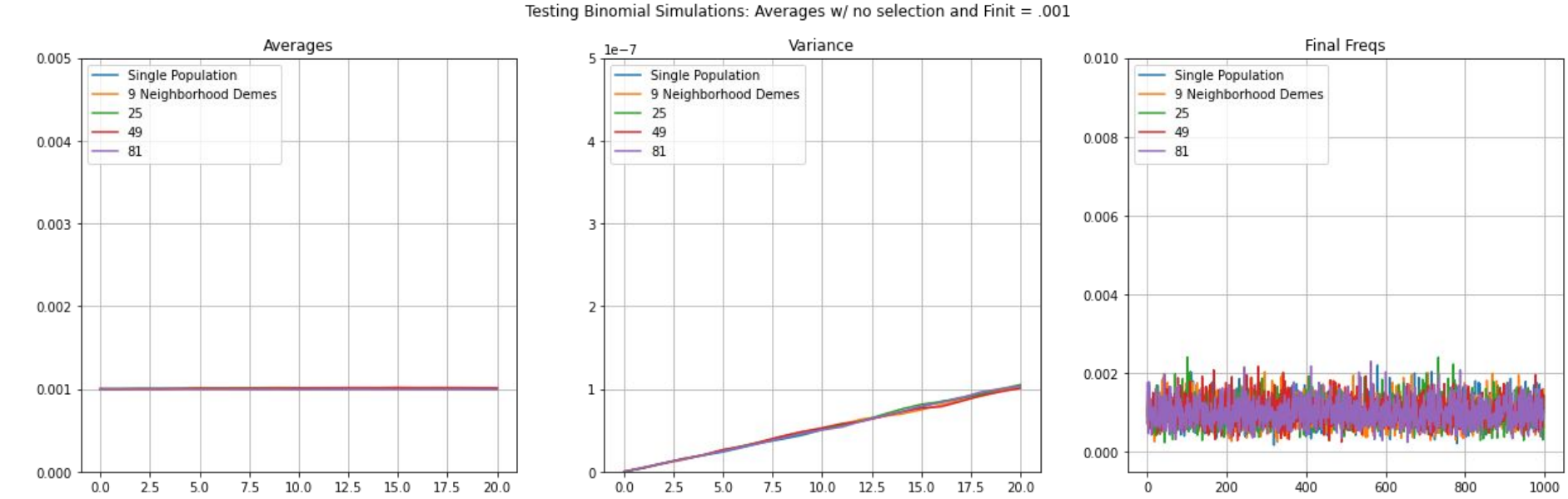
```
print(sum(results[0][-1] < (10 ** -15)))
print(sum(results[1][-1] < (10 ** -15)))
print(sum(results[2][-1] < (10 ** -15)))
print(sum(results[3][-1] < (10 ** -15)))
print(sum(results[4][-1] < (10 ** -15)))
```

```
349
0
0
0
0
```

Comparing **Binomial Distribution** to Gaussian distribution

Parameters:
N_total = 100,000; F_init = 0.001; dt = 1; t = 20 generations; 1,000 iterations

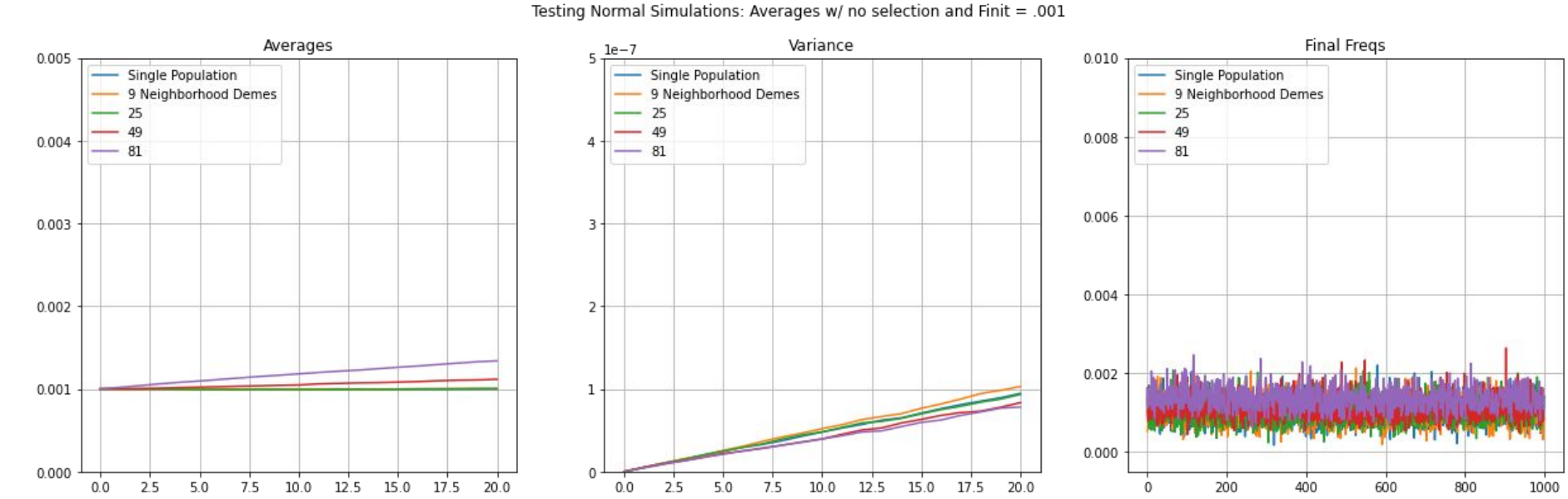
- no selection
- fully connected demes



Comparing Binomial Distribution to **Gaussian distribution**

Parameters:
N_total = 100,000; F_init = 0.001; dt = 1; t = 20 generations; 1,000 iterations

- no selection
- fully connected demes



Comparing **Gaussian distribution** to Binomial Distribution

Parameters:
N_total = 100,000; F_init = 0.001; dt = 1; t = 20 generations

- no selection
- fully connected demes

Binomial Alleles Lost
(of 1,000 iterations to 20 gen)

```
print(sum(results_bin[0][-1] < (10 ** -15)))
print(sum(results_bin[1][-1] < (10 ** -15)))
print(sum(results_bin[2][-1] < (10 ** -15)))
print(sum(results_bin[3][-1] < (10 ** -15)))
print(sum(results_bin[4][-1] < (10 ** -15)))
```

```
0
0
0
0
0
```

Normal Alleles Lost
(of 1,000 iterations to 20 gen)

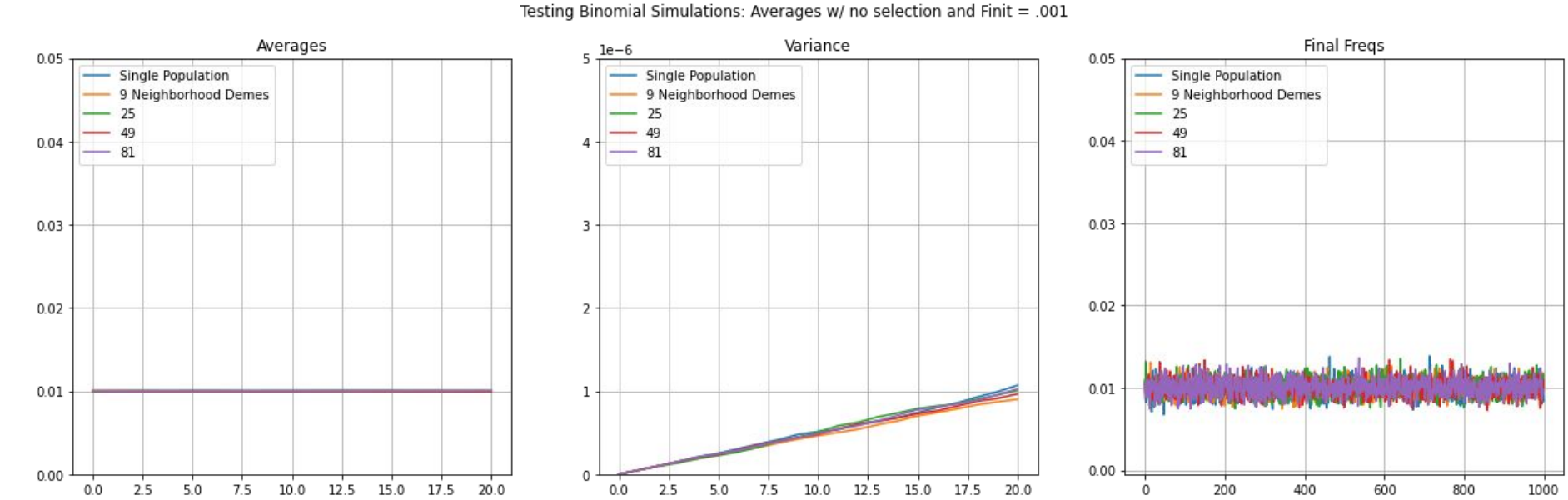
```
print(sum(results[0][-1] < (10 ** -15)))
print(sum(results[1][-1] < (10 ** -15)))
print(sum(results[2][-1] < (10 ** -15)))
print(sum(results[3][-1] < (10 ** -15)))
print(sum(results[4][-1] < (10 ** -15)))
```

```
0
0
0
0
0
```

Comparing **Binomial Distribution** to Gaussian distribution

Parameters:
N_total = 100,000; F_init = 0.01; dt = 1; t = 20 generations; 1,000 iterations

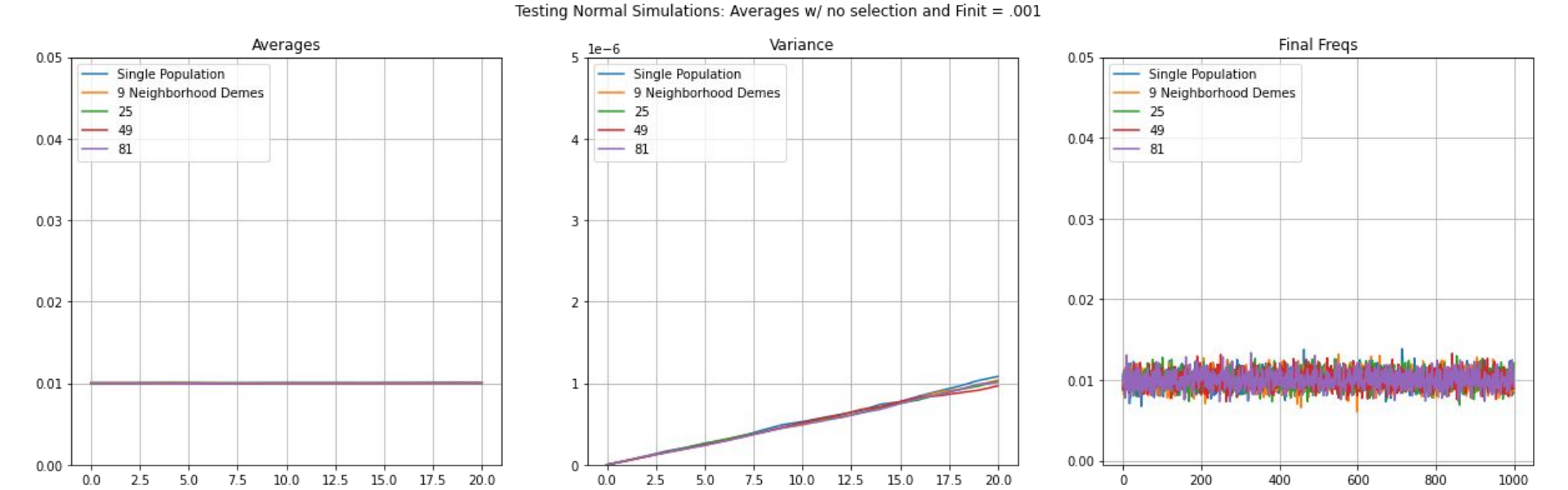
- no selection
- fully connected demes



Comparing Binomial Distribution to **Gaussian distribution**

Parameters:
N_total = 100,000; F_init = 0.01; dt = 1; t = 20 generations; 1,000 iterations

- no selection
- fully connected demes



Article

Inference of Population Structure from Time-Series Genotype Data

Tyler A. Joseph [1], Itsik Pe'er [1 2 3]

We assume each sampled individual is a mixture of K unobserved populations. Let the vector $\boldsymbol{\theta_d} = (\theta_{d1}, \theta_{d2}, \ldots, \theta_{dK})$ give the proportion of the genome individual d inherits from each population k; hence $\sum_{k=1}^{K} \theta_{dk} = 1$. Denote the allele frequency of the reference allele at locus l in population k at time point t by β_{kl}^{t}. With this notation, we assume the following generative model:

$$\boldsymbol{\theta_d} \sim \text{Dirichlet}\,(\alpha_1, \alpha_2, \ldots, \alpha_K) \qquad \text{for} \quad d = 1, \ldots, D. \tag{Equation 1}$$

$$\beta_{kl}^{t} | \beta_{kl}^{t-1} \sim \text{Normal}\left(\beta_{kl}^{t-1}, \tfrac{g_t - g_{t-1}}{12N_k}\right) \qquad \text{for } l = 1 \ldots L \text{ and } K = 1 \ldots K \tag{Equation 2}$$

$$x_{dl} | \beta_{1:K,l}^{t_d}, \boldsymbol{\theta}_d \sim \text{Binomial}\left(n_d, \textstyle\sum_{k=1}^{K} \theta_{dk} \beta_{kl}^{t_d}\right) \quad \text{for } d = 1 \ldots D \text{ and } l = 1 \ldots L \tag{Equation 3}$$

with initial allele frequencies β_{kl}^{0} and effective population sizes N_k treated as parameters. The β_{kl}^{t} are estimated from data, while N_k are treated as known and fixed. Ancient DNA samples are typically pseudo-haploid, as low sequencing depth makes it difficult to call full diploid genotypes. We explicitly model pseudo-haploid individuals by setting the sample size parameter of the binomial as either $n_d = 2$ or $n_d = 1$ depending on whether an individual is diploid or pseudo-haploid.

Note that the variance for drift is different than the variance obtained under Wright-Fisher or using a diffusion approximation. Traditionally, the variance of $\beta_{kl}^{t} | \beta_{kl}^{t-1}$ is

$$\text{Var}\left(\beta_{kl}^{t} | \beta_{kl}^{t-1}\right) = \tfrac{\beta_{kl}^{t-1}(1-\beta_{kl}^{t-1})(g_t - g_{t-1})}{2N_k}\,.$$

However, the appearance of allele frequencies in the variance leads to difficulties in deriving an inference algorithm. We approximate the variance by taking the average variance over possibly allele frequencies:

$$\text{Var}\left(\beta_{kl}^{t} | \beta_{kl}^{t-1}\right) \approx \tfrac{g_t - g_{t-1}}{12N_k} = \int_0^1 \tfrac{\beta_{kl}^{t-1}(1-\beta_{kl}^{t-1})(g_t - g_{t-1})}{2N_k} d\beta_{kl}^{t-1}\,.$$

Intuitively, this is similar to assuming a uniform prior over β_{kl}^{t-1} in the variance and taking the expectation.