Report: Genomic Assisted Breeding

Pedro Bueso-Inchausti Garcia

2020-1-27

Contents

1 Study the genetic drift through simulation

1.1 Introduction

The allele frequency is the fraction of copies of one gene that share a particular form in a population. The genetic or allelic drift [1, 2, 3] refers to the changes in this allele frequency due to the random sampling of organisms. Alleles in the offspring are a sample of those in the parents. The widespread idea is that such parent are selected through natural selection; however, randomness is also relevant, which is why genetic drift can explain many of the genetic changes appearing in populations.

If a evolutionary process is extended long enough, there are two possible outcomes for an allele: dissapereance or fixation. Let's think about the following example. We have a population of 20 organisms, 10 of which have the allele A and 10 of which have the allele B. In each new generation, the organisms reproduce at random and produce a offspring with 20 new individuals. If this process is repeated a number of times, the composition of each generation will fluctuate until eventally reaching a scenario where only one of the alleles is present. This allele would have become fixed while the other would have dissapeared. As random sampling can remove, but not replace an allele, genetic drift drives a population towards genetic uniformity over time.

This genetic drift consequences are sometimes difficult to understand. You start with a population where $A = B$ and where all individuals are equally likely to survive and reproduce. Why is it then that the most likely future scenario is one where $A \neq B$?. This question can be answered through the binomial distribution, which represents the number of successes in a sequence of n independent experiments, each with a propability p of success and $1 - p$ of failure. If we want to calculate the probabilities for exactly k successes, we use the following formula, where n is the number of experiments, k the number of succesful experiments and p the probability of success.

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

If we want to calculate the probabilities for a specific number of allele copies to survive, we can use the same formula, where n is the number of individuals in the future generation, k the number of copies of an allele in the future generations and p the probability of a given allele being present. We want to prove that having the same number of A and B alleles is less probable than having them unequally distributed. Therefore, we compare the probabilities of $X = 10$ and $X \neq 10$. The results make clear that a genetic drift from one population to its offspring is highly expected.

$$P(X = 10) = \frac{20!}{10!10!} \frac{1}{2}^{20} \approx 0.176$$

$$P(X \neq 10) = 1 - P(X = 10) \approx 0.824$$

There are several mathematical models that try to describe genetic drift. The two most important ones are the Wright-Fisher model and the Moran model. The former assumes that the generations do not overlap and that each individual found in the new generation is copied at random from all the individuals in the old generation. The latter assumes that the populations overlap and that, at each time, one individual is chosen to reproduce and one is chosen to die. In terms of computational requirement, the Wright-Fisher model is prefered because it takes t time steps, while the Moran model takes tn timesteps.

1.2 Defining our genetic drift function

This is the drift function with which we will be working. It operates in the following way. It receives the population size and the allele frequency. It starts the simulation, which does not end until the allele frequency equals 0 or 1, which means that one allele has fixed. In each generation, the genotypes of the new population are defined from a binomial distribution sampling which considers the allele frequency of the previous generation as the probability of success. It follows, therefore, the Wright-Fisher model. The function also includes the possibility to simulate bottlenecks every certain number of generations. This consists on taking a smaller sample (25% of the normal population size) for conforming future generations.

```
simulate_drift = function(num_individuals,allele_freq,num_simulation,bottleneck=FALSE,each=NULL)
{
  init_allele_freq = allele_freq
  allele_freq_evolution = c(init_allele_freq)
  generation = 0
  generation_evolution = c(generation)

  while(allele_freq>0 & allele_freq<1)
  {
    generation = generation + 1
    generation_evolution = c(generation_evolution,generation)
    if (bottleneck==TRUE)
    {
      if  (generation%%each==0)
      {genotypes = rbinom(integer(0.25*num_individuals),1,allele_freq)}
      else
      {genotypes = rbinom(num_individuals,1,allele_freq)}
    }
    else
    {genotypes = rbinom(num_individuals,1,allele_freq)}
    allele_freq = mean(genotypes)
    allele_freq_evolution = c(allele_freq_evolution,allele_freq)
  }

  results = data.frame(Generation=generation_evolution,Allele_freq=allele_freq_evolution)
  results["Params"] = paste0("NumIndividuals:",num_individuals,", AlleleFreq:",init_allele_freq)
  if (bottleneck==TRUE) {results["Bottleneck"] = paste0("Every ",each," generations")}
  else {results["Bottleneck"] = "No bottleneck"}
  results["Simulation"] = num_simulation
  return(results)
}
```

1.3 Exploring generations to fixation (GTF) for different population sizes

According to theory, small populations achieve fixation faster, whereas big population achieve it slower. In other words, the expected number of generations for fixation to occur is proportional to the population size [4]. In this section, we will prove this through simulation.

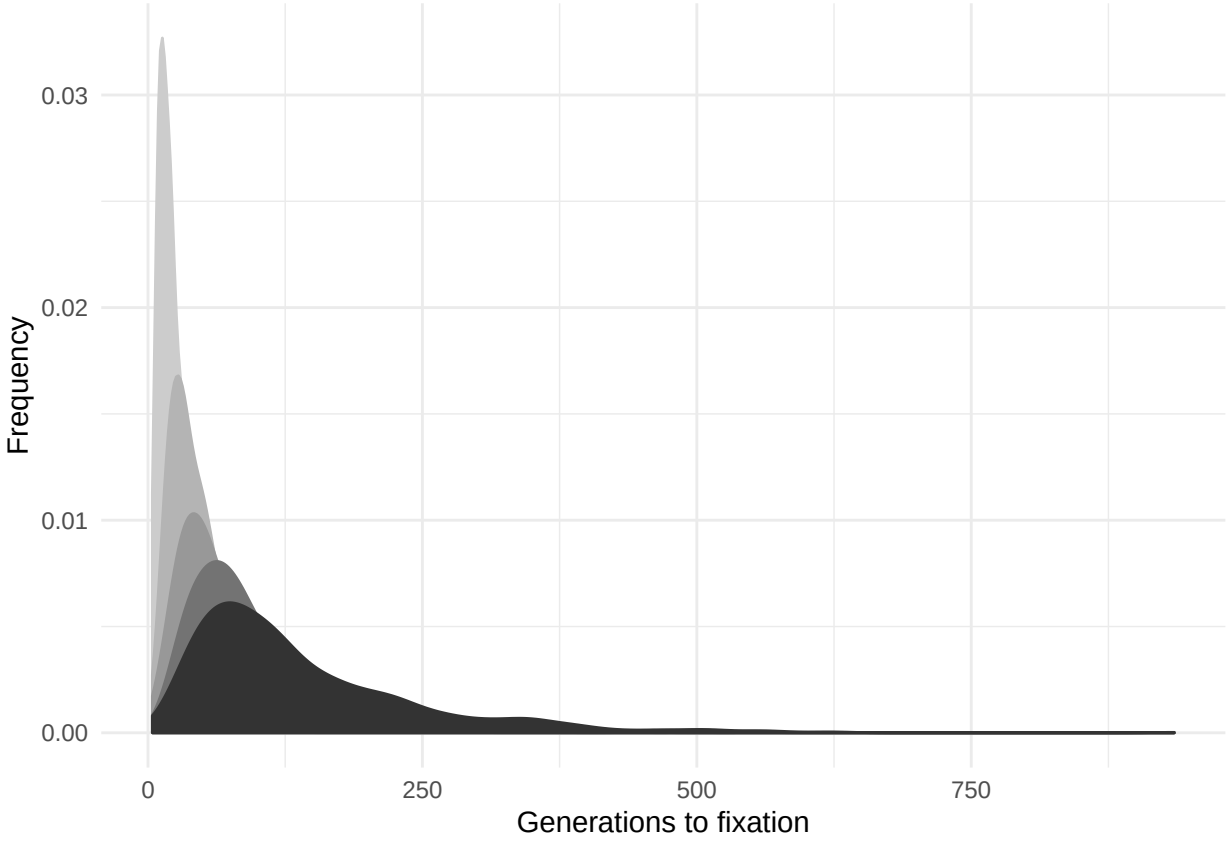
This function allows plotting the distribution of generations to fixation for multiple drift simulations.

```
plot_GTF_drift = function(df)
{
  plot = ggplot(df,aes(x=Values,fill=Factors,color=Factors)) +
    geom_density() +
    xlab("Generations to fixation") +
    ylab("Frequency") +
    scale_color_grey(start=0.8,end=0.2) +
    scale_fill_grey(start=0.8,end=0.2) +
    theme_minimal() +
    theme(legend.position="none")
  return(plot)
}
```

This code allows running multiple simulations with different population sizes.

```
GTF_all=c(); GTF_means=c(); GTF_standard_deviations=c()
num_individuals=c(20,40,60,80,100); num_simulations=1000; count_simulations=0
for (individuals in num_individuals)
{
  GTFs = c()
  for (simulation in 1:num_simulations)
  {
    count_simulations = count_simulations + 1
    df = simulate_drift(individuals,0.5,count_simulations)
    GTF = nrow(df)
    GTFs = c(GTFs,GTF)
  }
  GTF_all = c(GTF_all,GTFs)
  GTF_means = c(GTF_means,mean(GTFs))
  GTF_standard_deviations = c(GTF_standard_deviations,sd(GTFs))
}
df1 = data.frame(Factors=as.factor(rep(num_individuals,each=num_simulations)), Values=GTF_all)
df2 = data.frame(NumIndividuals=num_individuals, GTFMean=GTF_means, GTFStdDev=GTF_standard_deviations)
```

From the results, the relationship between the population size and the number of generations to fixation is clear; the higher the population size, the more generations are needed. This comes to show that genetic drift has a greater impact when populations are small.



NumIndividuals	GTFMean	GTFStdDev
20	26.742	19.63698
40	53.034	39.53031
60	85.466	64.49527
80	109.008	76.20012
100	141.044	105.29405

1.4 Exploring final allele frequencies (AF) for different initial frequencies

According to theory, the probability for an allele to fixate equals its initial relative abundance. In this section, we will prove this through simulation.

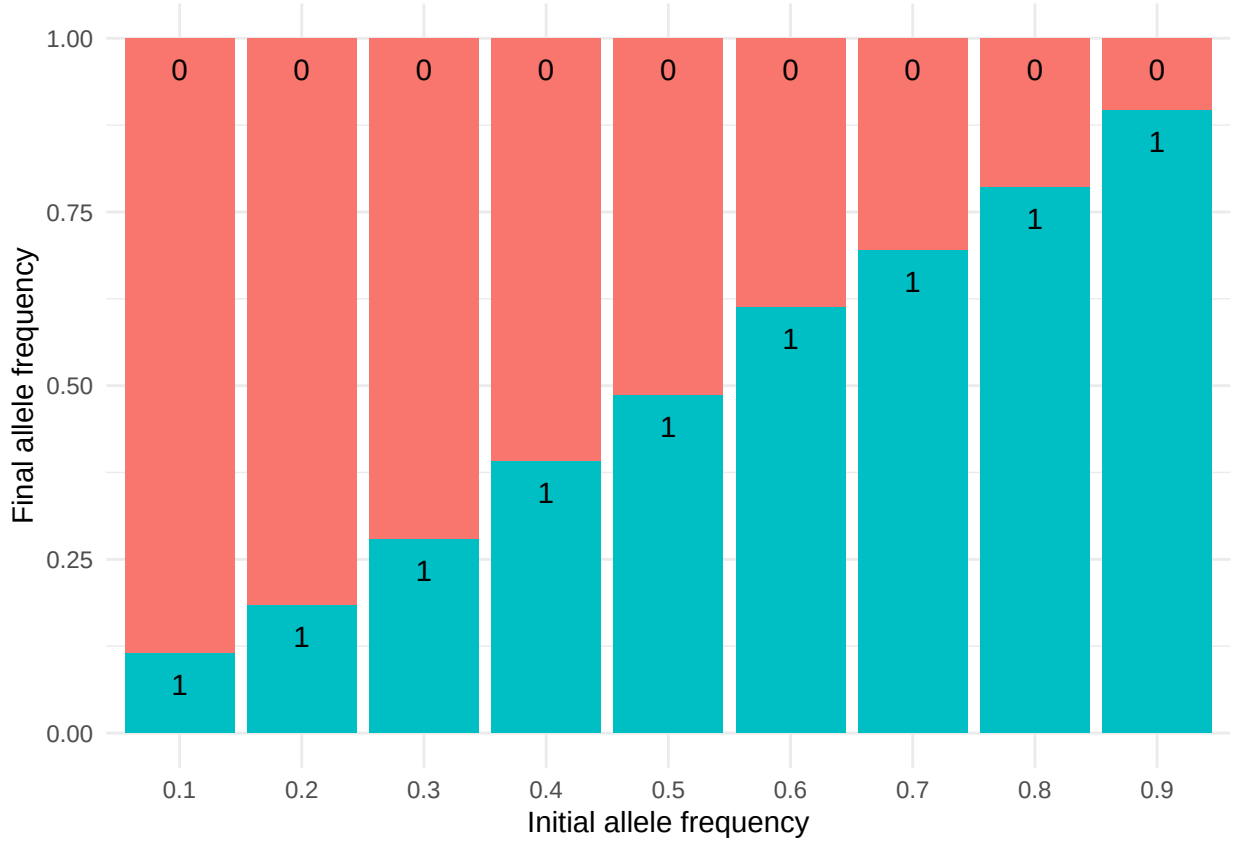
This function allows plotting the distribution of final allele frequencies for multiple drift simulations.

```
plot_AF_drift = function(df,xlab)
{
  df = ddply(df,"Priori",transform, Position=cumsum(Values))
  plot = ggplot(df,aes(x=Priori,fill=Posteriori,y=Values)) +
    geom_bar(stat = "identity") +
    xlab(xlab) +
    ylab("Final allele frequency") +
    geom_text(aes(y=Position, label=Posteriori), vjust=2) +
    theme_minimal() +
    theme(legend.position="none")
  return(plot)
}
```

This code allows running multiple simulations with different allele frequencies.

```
AF_1s=c(); AF_0s=c(); AF_10s=c()
allele_freq=seq(0.1,0.9,0.1); num_simulations=1000; count_simulations=0
for (allele in allele_freq)
{
  AFs = c()
  for (simulation in 1:num_simulations)
  {
    count_simulations = count_simulations + 1
    df = simulate_drift(50,allele,count_simulations)
    AF = df[nrow(df),]$Allele_freq
    AFs = c(AFs,AF)
  }
  AF_1s = c(AF_1s,mean(AFs))
  AF_0s = c(AF_0s,1-mean(AFs))
  AF_10s = c(AF_10s,mean(AFs),1-mean(AFs))
}
df1 = data.frame(Priori=as.factor(rep(allele_freq,each=2)),
                 Posteriori=as.factor(rep(c(1,0),rep=length(allele_freq))),Values=AF_10s)
df2 = data.frame(AlleleFreq=allele_freq, Frequency1=AF_1s, Frequency0=AF_0s)
```

From the results, it is clear that the probability for an allele to fixate equals its initial relative abundance.



AlleleFreq	Frequency1	Frequency0
0.1	0.115	0.885
0.2	0.184	0.816
0.3	0.280	0.720
0.4	0.392	0.608
0.5	0.487	0.513
0.6	0.613	0.387
0.7	0.696	0.304
0.8	0.786	0.214
0.9	0.897	0.103

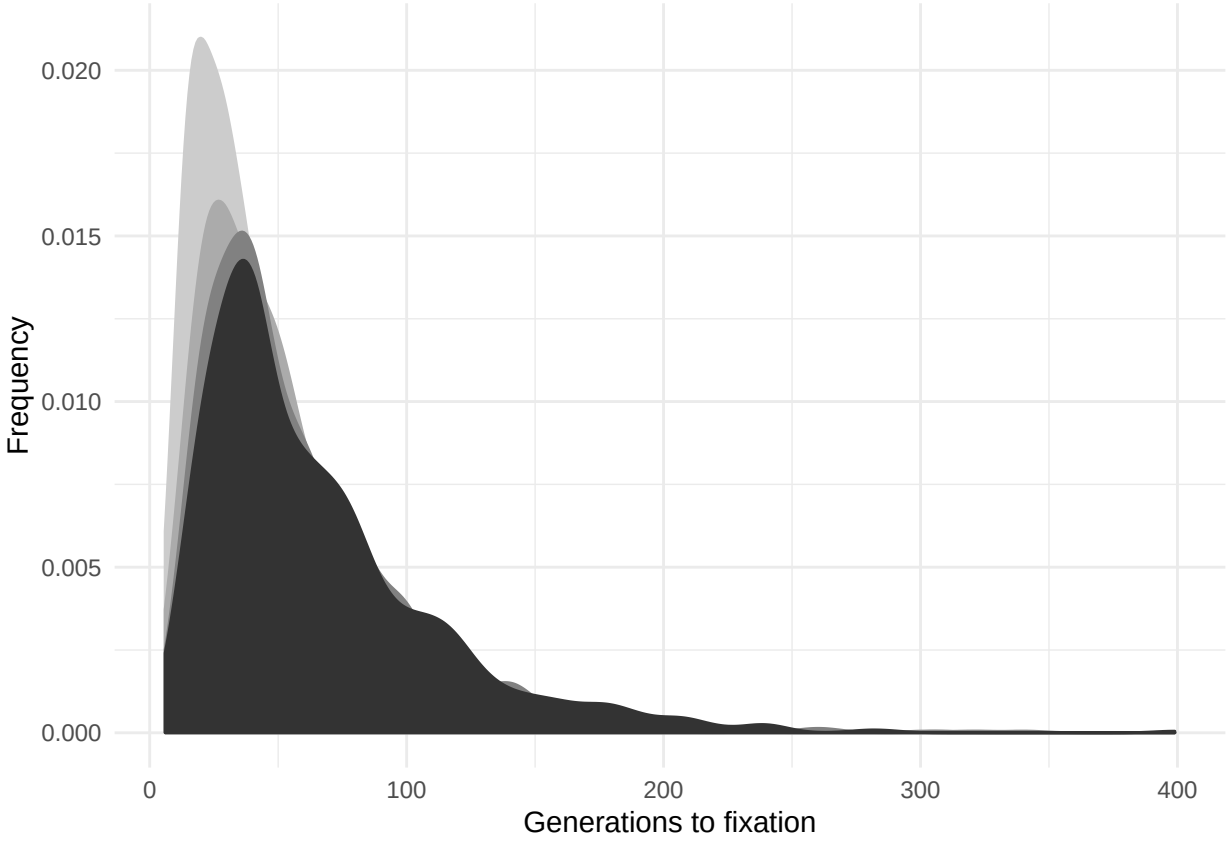
1.5 Exploring the bottlenecks

There are two events that could be implemented in a genetic drift simulation for studying 'what if' scenarios. These are the bottleneck effect and the founder effect. A bottleneck refers to the event where a population contracts to a significantly smaller size over a short period of time due to some random environmental event. This can generate radical changes in the allele frequencies, tipically leading a loss of genetic diversity and to great fluctuations in the allele frequencies. Since adaptation to changes requires from diversity, the loss of variation leaves the surviving population vulnerable to any new selection pressures. The founder effect is a special case of bottleneck, ocurring when a small group segregates from its original population and forms a new one. The random sample of alleles in the new colony is expected to grossly misrepresent the original population; for instance, if the number of alleles is higher than the number of founders, complete representation is impossible. The differences between the original population and the colony may also trigger a divergence that could grow through generations and eventually lead to a new species.

This code allows running multiple simulations with different bottleneck frequencies.

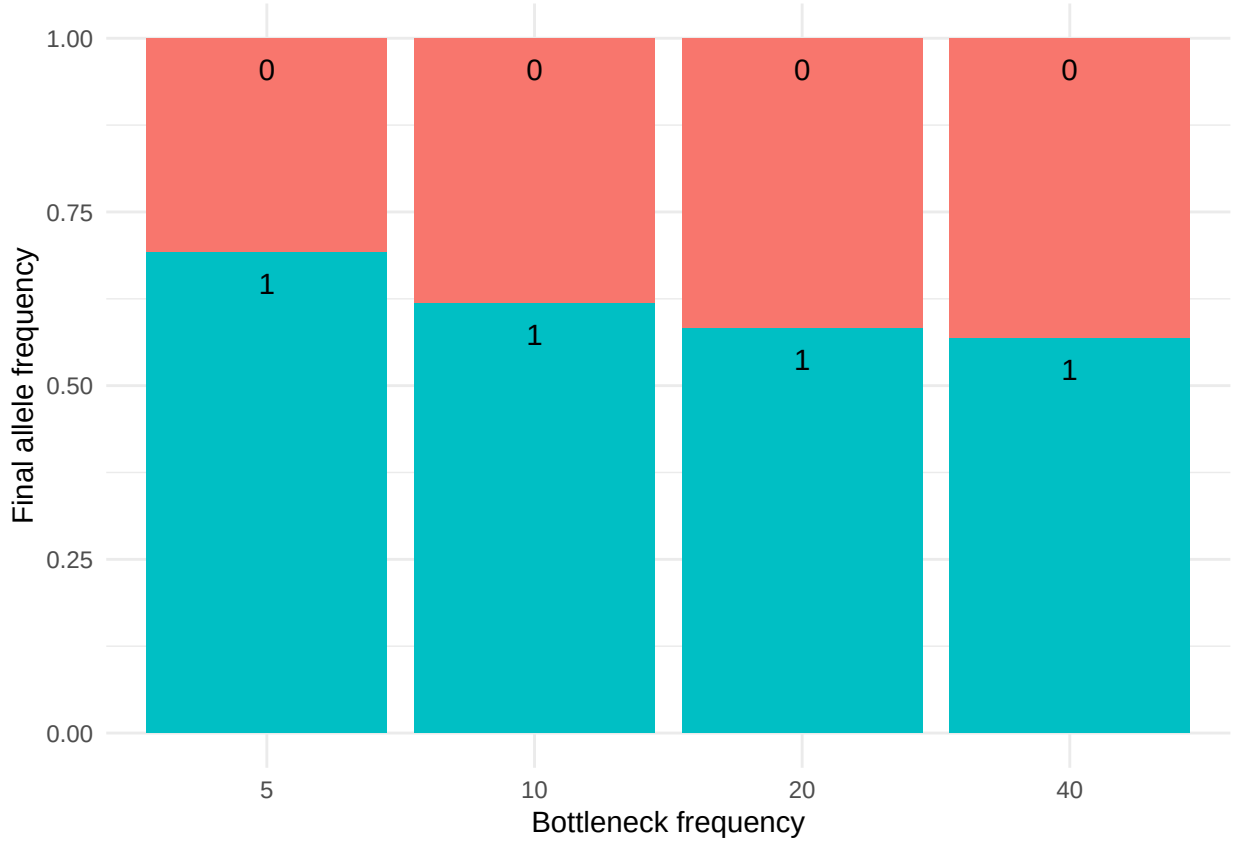
```
GTF_all=c(); GTF_means=c(); GTF_standard_deviations=c()
AF_1s=c(); AF_0s=c(); AF_10s=c()
bottleneck_freq=c(5,10,20,40); num_simulations=1000; count_simulations=0
for (bottleneck in bottleneck_freq)
{
  GTFs = c()
  for (simulation in 1:num_simulations)
  {
    count_simulations = count_simulations + 1
    df = simulate_drift(50,0.5,count_simulations,TRUE,bottleneck)
    GTF = nrow(df)
    GTFs = c(GTFs,GTF)
    AF = df[nrow(df),]$Allele_freq
    AFs = c(AFs,AF)
  }
  GTF_all = c(GTF_all,GTFs)
  GTF_means = c(GTF_means,mean(GTFs))
  GTF_standard_deviations = c(GTF_standard_deviations,sd(GTFs))
  AF_1s = c(AF_1s,mean(AFs))
  AF_0s = c(AF_0s,1-mean(AFs))
  AF_10s = c(AF_10s,mean(AFs),1-mean(AFs))
}
df1 = data.frame(Factors=as.factor(rep(bottleneck_freq,each=num_simulations)), Values=GTF_all)
df2 = data.frame(BottleneckFreq=bottleneck_freq, GTFMean=GTF_means, GTFStdDev=GTF_standard_deviations)
df3 = data.frame(Priori=as.factor(rep(bottleneck_freq,each=2)),
                 Posteriori=as.factor(rep(c(1,0),rep=length(bottleneck_freq))),Values=AF_10s)
df4 = data.frame(BottleneckFreq=bottleneck_freq, Frequency1=AF_1s, Frequency0=AF_0s)
```

From the results, the relationship between the frequency of bottlenecks and the number of generations to fixation is clear; the higher the frequency of bottlenecks, the less generations are needed. This comes to show that the bottlenecks push towards genetic homogeneity.



BottleneckFreq	GTFMean	GTFStdDev
5	42.436	32.54196
10	52.648	37.07777
20	59.812	42.33610
40	64.996	45.67227

From the results, we can see something pretty interesting. The higher the frequency of bottlenecks, the further the final allele frequency is from its initial relative abundance. This makes sense because bottlenecks drive the outcome away from what it would be expected. What I am unable to explain is why bottleneck might be pushing, as they do, towards the fixation of allele 1 (note that we start with a relative abundance of 0.5 but the final allele frequency are all over such threshold).



BottleneckFreq	Frequency1	Frequency0
5	0.6930000	0.3070000
10	0.6186667	0.3813333
20	0.5837500	0.4162500
40	0.5692000	0.4308000

2 Relationship between GWAS and recombination

2.1 Introduction to GWAS

A genome-wide association study (GWAS) [5, 6, 7] is an observational study of a genome-wide set of variants in different individuals to see if any variant is associated with a trait. Although it can be applied to any genetic variant and to any organism, it is tipically focused on association between single-nucleotide polymorphisms (SNPs) and traits like human major diseases. It is important to remark that GWAS is a phenotype-first approach (it classifies individuals according to their manifestations) and a non-candidate driven approach (it investigates the entire genome and not a predefined region).

Prior to GWAS introduction, the main methods for studying the outcome of a given trait was through inheritance studies of genetic linkage in families. This approach,though very useful for single gene disorders, was innadecuate for more complex diseases. GWAS showed that an alternative approach were possible. This consisted on asking whether the allele of a genetic variant was found more often in individiduals with the phenotype of interest. If so, the variant is said to be associated with the disease.

Let's now study GWAS methodology more into depth. The most common approach is the case-control setup, which compares two large groups of individuals (control -healthy- and case -affected by a disease-). All individuals in each group are genotyped for the majority of common known SNPs. For each of these SNPs, it is then investigated if the allele frequency is significantly altered between the case and the control groups. This is done with an odd ratio, shown down below.

$$\text{Odd ratio} = \frac{\text{Odds of case for individuals having the allele}}{\text{Odds of case for individuals not having the allele}}$$

When the allele frequency in the case group is much higher than in the control group, the odd ratio is higher than 1. When the allele frequency in the case group is much lower than in the control groups, the odd ratio is lower than 1. The objective is, consequently, to find odd ratios different to 1.

What is actually tested, however, is not the odd ratios but the associated p-values. Because so many variant are considered, there is a risk to have multiple false positives. For instance, if we test 1,000,000 SNPs and assume a level of significance of 0.05 for each independent test, then 5% of the comparisons (50,000 SNPs) are expected to be significant by pure chance. Therefore, p-values are adjusted (it is standard to consider variants as significant if their p-value is lower than $5x10^{-8}$).

After the p-values have been calculated, a common approach is to create a Manhattan plot, which plots $-log_{10}pvalue$ against the position in the genome. This results in the SNPs with the most significant association to stand out on the plot.

2.2 Linkage disequilibrium and recombination

Two loci are said to be in linkage disequilibrium when the frequency of association of their different alleles is higher or lower than what would be expected if the loci were independent and associated randomly [8].

Let's use the following example. Suppose that, in a given cell, allele A occurs with a frequency P_A at one locus and allele B occurs with a frequency P_B at a different locus. The probability for both A and B ocurring together is P_{AB}. When the ocurrence of one allele does not affect the other, in which case $P_AP_B = P_{AB}$, these alleles are said to be in linkage equilibrium. Otherwise, they are said to be in linkage disequilibrium. The level of linkage disequilibrium is defined as $D_{AB} = P_{AB} - P_AP_B$.

Recombination has an important effect in linkage disequilibrium as it produces its dissapearence. Given sufficient evolutionary time, the ocurrence of random recombination events result in an equilibrium distribution of alleles at each locus. In other words, the frequency of a particular allele at a given locus will be independent of alleles at other linked loci [9]. In what sense is recombination relevant to GWAS? Well, GWAS uses linkage disequilibrium mapping to cover the entire genome by genotyping a subset of variants. From what has previously been explained, this mapping is only possible with the existance of recombination.

References

[1] J. Masel, "Genetic drift", 2011.

[2] B. Star and H. G. Spencer, "Effects of Genetic Drift and Gene Flow on the Selective Maintenance of Genetic Variation", 2013.

[3] D. Hartl and A. Clark, "Principles of population genetics". Sianauer associates Inc. Canada. Sinauer Associates, 1997.

[4] S. P. Otto and M. C. Whitlock, "The Probability of Fixation in Populations of Changing Size", 1997.

[5] "Genome-Wide Association Studies Fact Sheet", Online, Available: https://www.genome.gov/about-genomics/fact-sheets/Genome-Wide-Association-Studies-Fact-Sheet, Accessed: 23-Jan-2020.

[6] "What are genome wide association studies (GWAS)?", Online, Available: https://www.ebi.ac.uk/training/online/course/gwas-catalog-exploring-snp-trait-associations-2019/what-gwas-catalog/what-are-genome-wide, Accessed: 23-Jan-2020.

[7] W. S. Bush and J. H. Moore, "Chapter 11: Genome-Wide Association Studies".

[8] M. Slatkin, "Linkage disequilibrium-understanding the evolutionary past and mapping the medical future HHS Public Access", Nat Rev Genet, vol. 9, no. 6, pp. 477-485, 2008.

[9] R. R. Hudson, "Linkage Disequilibrium and Recombination", Handbook of Statistical Genetics, Chichester: John Wiley & Sons, Ltd, 2004.