Drift versus Draft - Classifying the Dynamics of Neutral Evolution

Alison Feder*

December 13, 2013

1 Introduction

Evolutionary biologists agree that genomes change over time, although the exact mechanisms of how and why are still unclear. Mutations enter the population that introduce new traits (called alleles), but it is not well understood what causes these traits to rise or sink in frequency.

Some trait value changes can be expected. For example, if a trait allows individuals to digest the sugars in milk, milk digesters will have an advantage against other individuals in the population. They will have more offspring (also able to digest the sugars), and we can expect the trait to deterministically increase in frequency.

These sites under selection likely make up only a small fraction of the genome. Many other traits are thought to be selectively neutral - they neither cause the organisms that possess them to have more or fewer offspring. However, neutral sites still change in frequency over time, without being under selection. There are two competing hypotheses as to why this is:

Some biologists argue that it is driven by drift: Because the population of reproducing individuals is finite, there is some stochasticity in the system. If a certain trait is present in 500 out of 1000 individuals in one generation, it probably won't be present in exactly 500 individuals in the next generation. These fluctuations in trait frequencies are referred to as genetic drift. They are particularly acute in smaller populations, where the amount of stochasticity generated by reproduction is larger. Therefore, even if a trait does not confer any advantage or disadvantage, it will probably change in frequency over time.

Other biologists believe that these changes are driven by draft: although neutral sites do not have any selective advantage, they are nearby to sites that are under selection and frequency changes at nearby sites are correlated. A process called recombination breaks up these correlations. If the recombination rate is not too high, the movements of selected traits will drag the movements of neutral traits in a process called genetic draft. An illustration of this is given in figure 1.

Undoubtedly, both drift and draft both affect how neutral sites in the genome change over time. However, understanding if ones of these drives neutral changes over the other provides further insights into the importance of selection, recombination and population size. In this paper, I describe a machine learning approach to differentiate patterns of neutral traits that appear in the genome. Using forward genetic simulations and support vector machines, I set out to show that draft leaves a different signal in the genome than drift does.

*Early stages of this project were discussed with Dr. Philipp Messer

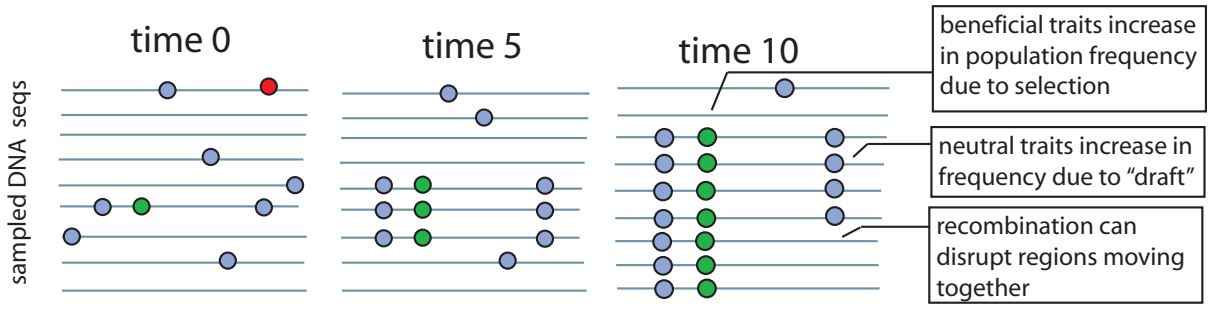

Figure 1: At time 0, a mutation with a selective advantage (green) enters the population by occurring in the DNA of a single individual. This individual already possesses some neutral traits. As time proceeds, offspring of that individual outcompete others strains, and so the trait rises in frequency within the population. This results in an increase in frequency of nearby neutral sites by genetic draft, even if those traits offer no advantages.

1.1 Why machine learning?

Machine learning is a particularly good approach to this problem because of the complicated dynamics surrounding changes in allele frequencies over time. While the dynamics of drift are tractable, and have been explored, they are require significant approximation, and can become particularly obfuscated when allele frequencies are close to boundaries (near 0% or 100%).

The dynamics of draft are even harder to model analytically. Every site is affected by every other site and the probability that there will be a recombination event between them. Further, this recombination rate is not constant throughout the genome. A machine learning approach that can can look for patterns in these complicated dynamics means that we can approach problems that would otherwise not be analytically tractable.

2 Data

The data are generated using a forward genetic simulator called SLiM.[1] SLiM simulates populations of individuals evolving over time under different parameter regimes. In particular, SLiM allows you to evolve populations with different selection and recombination coefficients and for different population sizes.

Forward simulation offers many advantages over using real data, especially during the development phase of an approach like this. First, simulations are much cheaper and faster than evolving populations of fruit flies or bacteria. Second, in simulations, the true state of a population evolution is known. While tools exist to find non-neutrally evolving regions, this adds an additional layer of complication. Finally, simulations allow the opportunity to tune parameters and test a large variety of drifting and drafting regimes. In particular, higher selection and a lower recombination rate will lead to the signals of draft being stronger, whereas a smaller population size will increase the magnitude of drift. These are patterns we would like to be able to specify.

I simulated the evolution of fragments of DNA of length 10^6 base pairs (bp) long in a population of size $N = 1000$. The sequences acquired mutations with a rate of $\mu = 10^{-7}$ mutations/bp/generation. Populations evolved for a 5000 generation burn-in to reach an equilibrium of standing genetic variation. After the burn-in, each DNA sequence from the population was sampled the next five generations and trait frequencies were recorded. I also included recombination (a process that breaks up tracks of nearby DNA all evolving together) and, in the cases of draft, a certain proportion of sites under selection (P_s) across varying selective strengths (s). This will be discussed more extensively in Section 2.2. Thus, each run of a simulation produced population-wide trait frequencies of one instantiation of a 5000 generation evolution at every genomic position. A visualization can be seen in Figure 2.

[1]SLiM: Simulating Evolution with Selection and Linkage. Genetics. 194:1037

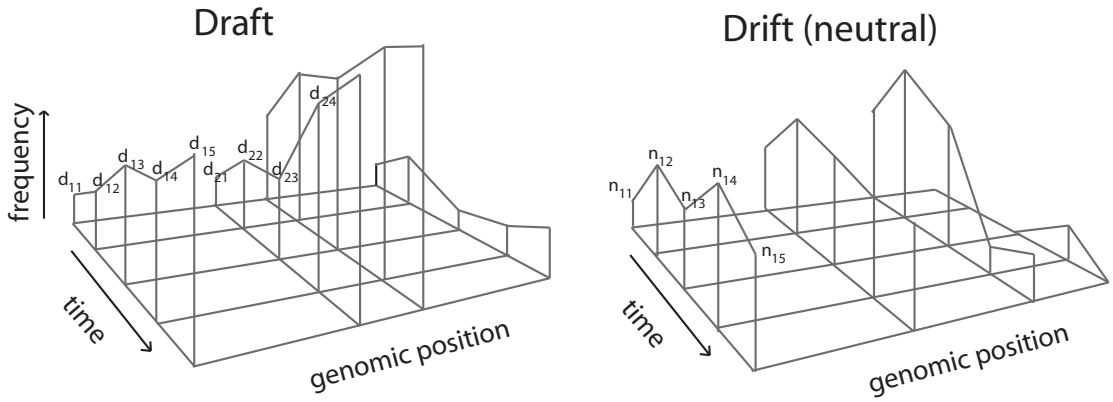

Figure 2: Parsed SLiM output example. For each polymorphic trait (at least two variants present in the population), the frequency of one of the traits was tracked across five time points, and we have information about the position of each trait relative to all other traits. Points were labeled such that the i^{th} polymorphic trait at the j^{th} time point for a purely drifting simulation was labeled n_{ij} and for a simulation with draft was labeled d_{ij}.

2.1 Data processing

From this state, I included only positions that had trait frequencies between 0 and 1 throughout the duration of the five sampled time points. At these positions, I recorded the 5-tuple encoding the five trait frequencies. In the trials with draft, I discarded the 5-tuple for any site that was under selection, so that only neutral sites were included in my final implementation. I flattened all 5-tuples from a given population into a vector.

$$\begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{21} & d_{22} & ... \\ d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{21} & d_{22} & ... \\ ... & & & & & & & ... \\ n_{11} & n_{12} & n_{13} & n_{14} & n_{15} & n_{21} & n_{22} & ... \\ ... & & & & & & & ... \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ ... \\ -1 \\ ... \end{bmatrix}$$

where labels are given as in Figure 2, and different rows of d_{ij} represent different independent runs of SLiM. Because not every simulation had the same number of positions with mutants, each feature vector was increased to be the size of the longest feature vector, with 0s to fill the end positions. I determined that this was a reasonable thing to do, because information about the number of sites is encoded in the 0s added (i.e., fewer mutated sites as represented by more 0s may be indicative of drift or draft). Each evolution of a population formed an input vector which was then classified as 1 or -1, depending on if selection was present or not.

2.2 Parameter Tuning

In order for draft to show up in our data, the recombination rate ρ and the amount of selection have to be carefully tuned to create dynamics differentiable from drift.

The recombination rate (ρ) controls the size of the DNA blocks (linkage blocks) that move together. If ρ is set too low, linkage blocks are long, and many selective beneficial mutations occur on each block. These competing mutations obfuscate selective dynamics in a process called interference. If ρ is too big, the linkage blocks are very small, and any single site under selection does not influence its surrounding sites.

The percentage of selected sites (P_s) and the selection strength (s) control the power of the selective dynamics. If too few sites are under selection, much of the genome will evolve without draft. If there are too many selected mutations, interference will make the patterns of draft harder to discern.

I choose ρ/P_s combinations that would allow for, on average, one mutation per linkage block. If $\rho = 10^{-8}$, then there will be 10^6 bp $*10^{-8}$ recombination events/bp/gen * 5000 generations = 50 recombination events. If the overall mutation rate is 10^{-7} mutations/bp/generation, then if 1% of mutations are positively selected, then, 10^6 bp $*10^{-7}$ mutations/bp/gen * 5000 generations * 1% mutations positively selected = 50 positively selected mutations. Similarly, if $\rho = 10^{-9}$, then 0.5%

of mutations should be positively selected, in order to achieve an expected one selected mutation per linkage block.

It is also important how strongly selection acts on these mutations. If selection is very strong (s large), sweeps go too fast and we cannot observe draft dynamics in action. If selection is weak (s small), sweeps are not strong enough to create draft. In order to determine which s to test thoroughly, small amounts of data (500 drift and 500 draft trials per regime) were generated for values $s = 0.1, 1, 10$ for both regimes of recombination described above ($\rho = 10^{-8}, 10^{-9}$) . Support vector machines (SVMs) were fit to these data using the LIBLINEAR[2] package in Matlab.

For each value of ρ and of s, I performed 30/70 cross validations in order to test the model. In each one, I took the 150 drifting samples and combined it with 150 trials from one of the drafting population regimes. Results are plotted in Figure 3.

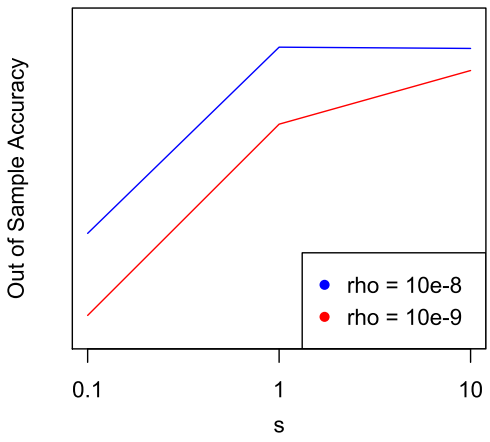

Figure 3: Different selection coefficients were tested across the $\rho = 10^{-8}$ (red) and $\rho = 10^{-9}$ (blue) parameter regimes described above.

Although fits were similar between $\rho = 10^{-9}$ and 10^{-8}, these results indicate that the strength of selection plays an important role in the ability of the model to differentiate drift and draft scenarios. Initially increasing s has larger returns that diminish as s becomes too large. This may correspond to the dip in diversity that accompanies stronger selective sweeps. Resultantly I chose to proceed with $\rho = 10^{-9}, s = 1$. Although this sample did not have the highest out of sample accuracy in the validation shown in Figure 3, I determined that a selection coefficient of $s = 10$ was unrealistically high, and that more useful information might be yielded from choosing the more moderate value.

2.3 Further SVM Evaluations

To delve deeper into the performance of one of the clearer parameter regimes from the first validation ($\rho = 10^{-9}, s = 1, P_s = 1\%$), I simulated 3000 drafting trials and 3000 drifting trials.

With these samples, I removed 500 of them to be a test set (randomized 250 drift, 250 draft). I then fit SVMs using different sized training sets in order to create a learning curve and tested the 500 removed samples. The out of sample accuracy increases as the training set size grows, and appears to flatten out at 73% (Figure 4).

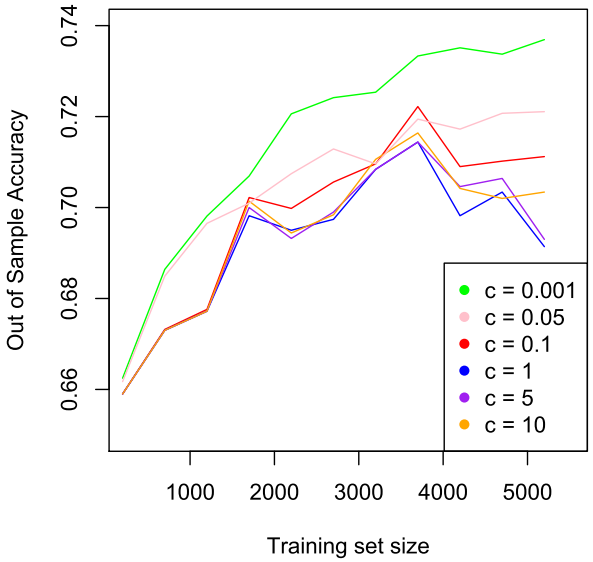

Figure 4: Out of sample accuracy across different training set sizes for different values of the slack variable c. While all models improve out of sample accuracy with increased training set size, the most highest out of sample accuracies at each training set size were achieved by the lowest slack variable ($c = 10^{-3}$). The dataset generated used the parameters $\rho = 10^{-8}, P_s = 1\%, s = 1$ and all other parameters as in Section 2.

[2]R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9(2008), 1871-1874. Software available at http://www.csie.ntu.edu.tw/ cjlin/liblinear

In order to achieve a better fit, I tuned the slack parameter c, which controls how misclassifications are weighted. It appears that marginal gains can be made by decreasing the slack variable, as the lowest value ($c = 10^{-3}$) achieved the highest out of sample accuracy. In addition, unlike higher c values, the training curve for $c = 10^{-3}$ may not have completely leveled out at the largest training set size, indicating that even further gains may be possible with a larger training set size.

The LIBLINEAR documentation suggested that in some cases, instructing the program to solve the primal problem as opposed to the dual will yield better results. Since the KKT conditions imply that solving the primal and dual problems yield the same optimization, this suggests that if one performs better than the other, there are computational problems associated with the maximization procedure. I fit learning curves separately using the same procedure as described above for both dual and primal optimizations. Although there is some evidence that decreasing the slack variable c can influence learning curves, I used the LIBLINEAR default value of $c = 1$ for these curves.

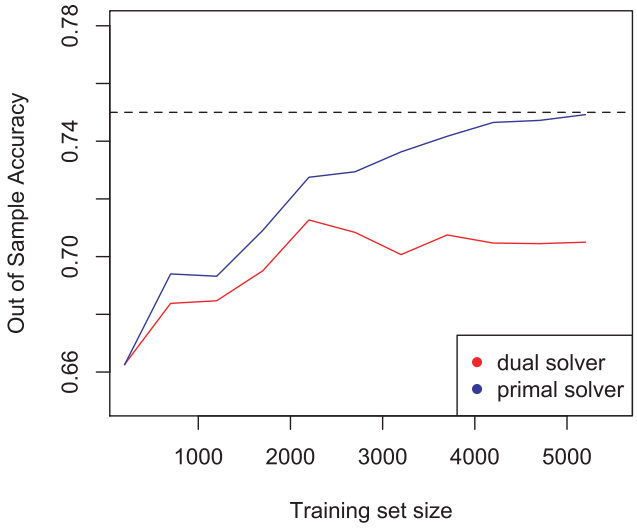

Figure 5: Learning curves were fit to the dataset using two different methods of solving the optimization problem. The primal solver (blue) produced higher out of sample accuracies than the dual solver (red) across all training set sizes when tested on a removed data portion. The dataset generated used the parameters $\rho = 10^{-8}, P_s = 1\%, s = 1$ and all other parameters as in Section 2.

I found that the solving the primal problem as opposed to the dual problem yielded higher out of sample accuracy values.

3 Future Directions

Much work must still be done before the this approach to differentiating drift and draft could be applied to real data. While a great deal of work has already gone into testing particular parameter regimes, many other variables have yet to be explored.

One of these variables is the time points at which sampling occurs. It may be that sampling every generation does not represent a sufficient amount of time for drafting patterns to become apparent. It is more probable that sampling every generation is a useful timescale for very strong selective sweeps, but holds little information for weaker selection. It may be that we must sample 10 or 100 time points before we can make very good classifications.

Another way to make potential improvements would be to try different ways of parametrizing the feature space. The approach I used simply listed the allele frequency across sites and across time and allowed the SVM to do the majority of the work. A more explicit approach might also incorporate several other pieces of information such as a dip in diversity and genome structural information.

A more direct approach might also use more of the spatial information between different trait values on the genome. In my parametrization of the feature vectors, traits that were tens of thousands of base pairs apart (with no other traits in between) and those occurring as direct neighbors were treated the same.

This is a complex problem with many different parameters controlling the extent to which draft is visible in the genome, but preliminary results from simulations suggest that SVMs are a promising way forward in differentiating the dynamics of drift and draft. While much remains to be done to get a to a point where this approach can be implemented on real data, simulation based studies are a good first step in the right direction.