Identifying genetic loci under selective pressure using a hierarchical Bayesian model with a posterior predictive p-value classifier

Toby Dylan Hocking

September 30, 2009

Contents

List of Figures

1 Introduction

The recent explosion of molecular marker data in animal populations from technologies such as Single Nucleotide Polymorphism (SNP) assays opens new avenues of research for population genetics. These data allow testing of many aspects of our current models of population genetics, such as the evolutionary status of these markers relative to selective pressures. These data are usually investigated by either examining summary statistics and empirical distributions, or by using model-based approaches. We are more concerned with model-based approaches, with emphasis on detecting departures from the model.

In this study, we are interested in is the estimation of genetic differentiation between populations, and establishing methods for determining which markers and genomic regions have been under selective pressure. Genomic areas under selection are areas with probable functional significance, thus the goal is to develop methods we can use to identify functional genes and augment the current state of functional annotations for domestic animal genomes.

In the last several decades, the statistical tools of biologists and geneticists have evolved considerably. In particular, modern computers and stochastic methods such as Markov Chain Monte Carlo (MCMC) allow for estimation of the posterior distribution of parameters of Bayesian models of evolutionary systems. In this work, we investigate one such Bayesian model used to describe evolution of domestic animal populations, and extend it with a classifier for markers under selection.

The model of domestic animal evolution that we consider in this work is the hierarchical Bayesian model of Nicholson *et al.* [6], hereafter referred to as the Nicholson model. This model assumes that a single population gives rise to several subpopulations, which branch off at the same time, and begin independently evolving. The Nicholson model assumes all loci are affected only by genetic drift (not selection), and attempts to measure population differentiation in a manner which is analogous to the classical F_{ST} of population genetics. In the process the model also yields estimates of ancestral allele frequency.

The new idea put forth in this work is a classifier for markers under selection, based on loci which do not fit well into the pure-drift Nicholson model. To quantify the probability that a locus fits the pure-drift model, we use Posterior Predictive P-values, or PPP-values [3, 2]. Essentially these PPP-values are the Bayesian analogue of the usual frequentist P-values, which indicate departures from the model hypotheses. The Nicholson model was designed for, and accurately models, independently evolving populations un-

der genetic drift. However, loci under selection in addition to genetic drift represent departures from the model hypotheses. Thus we use PPP-values estimated from the model to identify these aberrent loci.

Furthermore, to test the robustness of our classifier under different evolutionary conditions, we test it using extensive simulation of evolution by genetic drift and selection. The simulator has been implemented using the R programming language [7]. The model fitting has been implemented using compiled FORTRAN code dynamically linked to R. The simulations, analyses, and graphics discussed in this article can be reproduced by using the code published in the R package `nicholsonppp` on R-Forge [9]:

`http://nicholsonppp.r-forge.r-project.org/`

2 Methods

2.1 Simulating genetic drift and selection

To simulate selection and genetic drift in several independent populations, we use a modified version of the simulator in [1]. We assume L independent loci, and P independent populations, evolving over T generations.

To model SNP data, each locus has only 2 possible alleles, thus we denote the allele frequency for locus i in population j at generation t as $\alpha_{ij}(t)$, with $0 \leq \alpha_{ij}(t) \leq 1$. To assign ancestral allele frequencies

$$\pi_i = \alpha_{i1}(1) = ... = \alpha_{iP}(1)$$

we draw from a truncated $\beta(0.7, 0.7)$ distribution (Figure 1). That is, for all i,

$$P(\pi_i < 0.05) = P(\pi_i > 0.95) = 0$$

and if $Z \sim \beta(0.7, 0.7)$,

$$P(\pi_i = 0.05) = P(\pi_i = 0.95) = P(Z < 0.05)$$

We chose to truncate the distribution of initial allele frequencies so as to reduce the number of loci which are "fixed" with allele frequency 0 or 1 at the end of the simulation.

This choice is motivated by population genetics, which gives us the result that at equilibrium, under mutation and genetic drift, the distribution of allele frequencies approximately follow a $\beta(4N\mu, 4N\mu)$ distribution, where N is the effective population size and μ is the mutation rate [11]. We also considered using a truncated $\beta(1, 1)$ distribution, which is the same as the $U[0, 1]$ distribution, but no noticeable difference was observed.

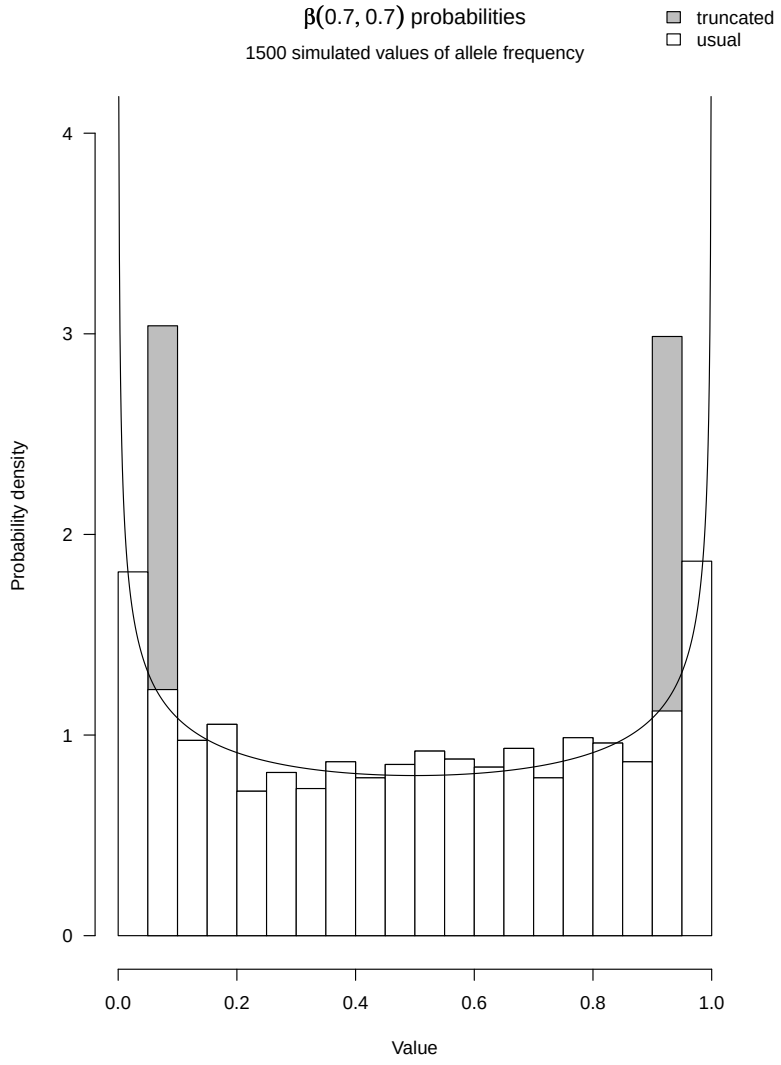


Figure 1: Distribution of ancestral allele frequencies in our simulations follow a truncated $\beta(0.7, 0.7)$ distribution.

If we assign the colors blue and red to the 2 alleles, and we define $\alpha_{ij}(t)$ as the blue allele frequency for locus i in population j at generation t, then $1 - \alpha_{ij}(t)$ is the red allele frequency. The color of the allele will determine its fate under selection in populations which each have a background color of either red or blue. Population color is chosen at random, independently for each locus, at the beginning of the simulation, with probability 0.4 for each of red or blue, and 0.2 for neutral populations where neither allele is favored.

The idea is to simulate the fact that some alleles are favored in some environments, while disfavored in others. Thus, under positive selection, red alleles will be favored in red populations, and disfavored in blue populations (vice versa for blue alleles). Under balancing selection, it is advantageous to have both a blue and red allele. This simulates the heterozygote advantage, a phenomenon classically observed in the gene that controls malaria resistence and sickle-cell anemia affectation.

The allele frequency changes in each generation via 2 mechanisms: drift and selection. Drift introduces some random variability up or down in allele frequency, independent of population color:

$$\alpha^*_{ij}(t) \sim \text{Bin}(N_{ij}, \alpha_{ij}(t-1))$$

The effect of drift grows more important relative to selection as population size N_{ij} diminishes.

Then to update the allele frequency for selection, we first calculate relative fitness of each diploid genotype. Relative fitness of a locus is based on the selection coefficient for that locus $s_i \in \mathbb{R}$, which is a parameter of the simulation, usually between 0 and 1 in empirical studies. Selection for a locus grows more important relative to genetic drift as s_i increases.

w_{ij}^{BB}	w_{ij}^{BR}	w_{ij}^{RR}	selection type	population color
1	$1 + s_i/2$	$1 + s_i$	positive	red
$1 + s_i$	$1 + s_i/2$	1	positive	blue
1	$1 + s_i$	1	balancing	
1	1	1	neutral	

Then we update blue allele frequency for selection based on Hardy-Weinberg equilibrium, which allows us to derive expressions for genotype frequencies in terms of allele frequency:

$$\alpha_{ij}(t) = \frac{w_{ij}^{\text{BB}}\alpha^*_{ij}(t)^2 + w_{ij}^{\text{BR}}\alpha^*_{ij}(t)[1-\alpha^*_{ij}(t)]/2}{w_{ij}^{\text{BB}}\alpha^*_{ij}(t)^2 + w_{ij}^{\text{BR}}\alpha^*_{ij}(t)[1-\alpha^*_{ij}(t)] + w_{ij}^{\text{RR}}[1-\alpha^*_{ij}(t)]^2}$$

We repeat the process for $t = 2, ..., T$, and we take values $\alpha_{ij}(T)$ as the output allele frequencies of the simulation.

2.2 The hierarchical bayesian Nicholson model

To model the variation between observed allele frequencies in different populations, the Nicholson model assigns a divergence parameter c_j to each population. The number of observed (blue) alleles for locus i in population j is modeled as

$$x_{ij} \sim \text{Binomial}(N_{ij}, \alpha_{ij})$$

where N_{ij} is the total number of alleles (red or blue), and α_{ij} is the population (blue) allele frequency.

This quantity is in turned modeled by

$$\alpha_{ij} \sim N(\pi_i, c_j \pi_i (1 - \pi_i))$$

a normal distribution truncated to the interval [0,1]. The differentiation parameter c_j is motivated by population genetics [6, section 2.2].

The distribution of the hyperparameter for ancestral allele frequency follows the prior distribution

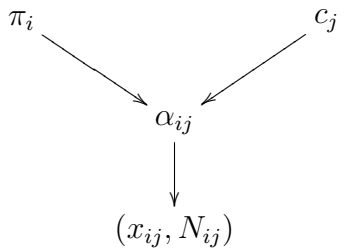
$$\pi_i \sim \beta(a, a)$$

Values $a \in \{0.7, 1\}$ were used, and showed similar results.

The population divergence hyperparameter follows the prior distribution

$$c_j \sim U[0, 1]$$

The relationship between model parameters is more clearly summarised in the following directed acyclic graph:



The point of Bayesian statistics is to exploit Bayes' Rule to obtain posterior distributions of the model parameters conditional on the data. Generally, if we use x to signify the observed data and θ to denote the model parameters, we can write the posterior distribution as

$$P(\theta|x) = \frac{P(x,\theta)}{P(x)} = \frac{P(x,\theta)}{\int P(x,\theta) d\theta} = \frac{f(x|\theta) g(\theta)}{\int f(x|\theta) g(\theta) d\theta} \tag{1}$$

where f is the density of the data x and g is the density of the model parameters θ.

Since it is often difficult to evaluate the integral in the denominator of equation 1, we instead turn to Markov Chain Monte Carlo techniques to sample from the posterior distribution. Essentially, we use a Metropolis-Hastings algorithm to draw samples from a Markov chain, thus giving us an approximation of the posterior distribution [5].

Thus, for each step in the chain t, we sample from the following posterior distributions:

$$\alpha^t = P(\alpha|c^{t-1}, \pi^{t-1}, x)$$
$$\pi^t = P(\pi|c^{t-1}, \alpha^t, a)$$
$$c^t = P(c|\pi^t, \alpha^t)$$

The model was first implemented using WinBUGS [4]. To provide speed optimizations for the model fitting, a faster model-fitting program was written in FORTRAN.

The posterior distribution for each model parameter was sampled 1000 times using MCMC, giving us an approximate posterior distribution. However, to simplify the analyses, each distribution was summarized using the mean. Thus when we refer to model parameter estimates, we mean the sample mean of the values drawn from the posterior distribution.

2.3 PPP-value calculation

The PPP-value for locus i is defined as

$$\text{PPP}_i = P\left[T(y_{ij}^{\text{rep}}, \theta_{ij}) > T(y_{ij}^{\text{obs}}, \theta_{ij})|y^{\text{obs}}\right]$$

where $y_{ij} = x_{ij}/N_{ij}$ is the allele frequency for locus i and population j, $\theta_{ij} = (\pi_i, c_j)$ is a vector of model parameters, and T is a discrepancy criterion applied to replicated (rep) and observed (obs) data sets.

We need to choose a discrepancy criterion which depends on both data and parameters. Here, we use a χ^2-type criterion:

$$T(y_{ij}, \theta_{ij}) = \sum_{j=1}^{P} \frac{\left[y_{ij} - E(y_{ij}|\theta_{ij})\right]^2}{\text{Var}(y_{ij}|\theta_{ij})}$$

The derivation of the expectation and variance of y_{ij} follows. First, we note that we can decompose the Binomial random variable

$$x_{ij} = \sum_{k=1}^{N_{ij}} x_{ijk}$$

into the sum of independent, identically distributed Bernoulli random variables $x_{ijk} \sim \text{Bernoulli}(\alpha_{ij})$. Then we can write

$$y_{ij}|\theta_{ij} = \frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} x_{ijk}$$

To find the expectation and variance of y_{ij} given the hyperparameters θ_{ij}, we will first characterize the moments of x_{ijk}, using the fact that $\alpha_{ij} \sim N(\pi_i, c_j \pi_i (1 - \pi_i))$:

$$E(x_{ijk}) = E_{\alpha_{ij}} \left[E(x_{ijk}|\alpha_{ij}) \right] = E(\alpha_{ij}) = \pi_i$$

$$\begin{aligned}
\text{Var}(x_{ijk}) &= \text{Var}_{\alpha_{ij}} \left[E(x_{ijk}|\alpha_{ij}) \right] + E_{\alpha_{ij}} \left[\text{Var}(x_{ijk}|\alpha_{ij}) \right] \\
&= \text{Var}_{\alpha_{ij}}(\alpha_{ij}) + E_{\alpha_{ij}}(\alpha_{ij}) - E_{\alpha_{ij}}({\alpha_{ij}}^2) \\
&= E_{\alpha_{ij}}(\alpha_{ij}^2) - E_{\alpha_{ij}}(\alpha_{ij})^2 + E_{\alpha_{ij}}(\alpha_{ij}) - E_{\alpha_{ij}}(\alpha_{ij}^2) \\
&= E_{\alpha_{ij}}(\alpha_{ij}) \left(1 - E_{\alpha_{ij}}(\alpha_{ij}) \right) + \\
&= \pi_i (1 - \pi_i)
\end{aligned}$$

Using a similar derivation, we find that

$$\begin{aligned}
\text{Cov}(x_{ijk}, x_{ijk'}) &= E_{\alpha_{ij}} \big[\underbrace{\text{Cov}(x_{ijk}, x_{ijk'}|\alpha_{ij})}_{0} \big] + \text{Cov} \big[\underbrace{E(x_{ijk}|\alpha_{ij})}_{\alpha_{ij}}, \underbrace{E(x_{ijk'}|\alpha_{ij})}_{\alpha_{ij}} \big] \\
&= 0 + \text{Cov}(\alpha_{ij}, \alpha_{ij}) \\
&= \text{Var}(\alpha_{ij}) \\
&= c_j \pi_i (1 - \pi_i)
\end{aligned}$$

Using the previous results, we can derive the expectation and variance of y_{ij}, which we need to calculate the PPP-values:

$$E(y_{ij}|\theta_{ij}) = \frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} E(x_{ijk}) = \frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} \pi_i = \pi_i$$

$$
\begin{aligned}
\mathrm{Var}(y_{ij}|\theta_{ij}) &= \mathrm{Var}\left(\sum_{k=1}^{N_{ij}} \frac{x_{ijk}}{N_{ij}}\right) \\
&= \frac{1}{N_{ij}^2}\left[\sum_{k=1}^{N_{ij}} \mathrm{Var}(x_{ijk}) + 2\sum_{k=1}^{N_{ij}}\sum_{k'=k+1}^{N_{ij}} \mathrm{Cov}(x_{ijk}, x_{ijk'})\right] \\
&= \frac{1}{N_{ij}^2}\left[\sum_{k=1}^{N_{ij}} \pi_i(1-\pi_i) + 2\sum_{k=1}^{N_{ij}}\sum_{k'=k+1}^{N_{ij}} c_j\pi_i(1-\pi_i)\right] \\
&= \frac{N_{ij}\pi_i(1-\pi_i) + N_{ij}(N_{ij}-1)c_j\pi_i(1-\pi_i)}{N_{ij}^2} \\
&= \frac{1}{N_{ij}}\pi_i(1-\pi_i)\left[1 + (N_{ij}-1)c_j\right]
\end{aligned}
$$

In practice, we will calculate the discrepancy criterion T for the replicated (rep) and observed (obs) data at each iteration in the Markov chain. Thus, using ^{t} to denote the value of a variable in the t-th iteration through the Markov chain,

$$
\begin{aligned}
T(y_{ij}^{\mathrm{rep},t}, \theta_{ij}^t) &= \frac{\left(\mathrm{Bin}(N_{ij}, \alpha_{ij}^t)/N_{ij} - \pi_i^t\right)^2}{\pi_i^t(1-\pi_i^t)\left[1 + (N_{ij}-1)c_j^t\right]/N_{ij}} \\
T(y_{ij}^{\mathrm{obs},t}, \theta_{ij}^t) &= \frac{\left(Y_{ij}/N_{ij} - \pi_i^t\right)^2}{\pi_i^t(1-\pi_i^t)\left[1 + (N_{ij}-1)c_j^t\right]/N_{ij}}
\end{aligned}
$$

where $\mathrm{Bin}(\cdot,\cdot)$ represents a randomly generated number from the binomial distribution.

Next, we define the indicator variable

$$
\mathrm{PPP}_i^t = \begin{cases} 1 & \text{if } \sum_{j=1}^{P}\left[T(y_{ij}^{\mathrm{rep},t}, \theta_{ij}^t) - T(y_{ij}^{\mathrm{obs},t}, \theta_{ij}^t)\right] > 0 \\ 0 & \text{otherwise} \end{cases}
$$

where P is the number of populations. We sum over populations since we need to have a criterion for each locus at each iteration.

Finally, the PPP-value for locus i is calculated at the end of the N_{iter} iterations of Markov chain sampling from the posterior distribution:

$$
\mathrm{PPP}_i = \frac{1}{N_{\mathrm{iter}}}\sum_{t=1}^{N_{\mathrm{iter}}} \mathrm{PPP}_i^t
$$

3 Results

3.1 Simulation verification

After obtaining allele frequencies from the simulator, we can do diagnostic plots to visually verify that the allele frequencies are evolving according to the theoretical evolution framework we had envisioned. R packages lattice and ggplot2 are used to visualize these multivariate data [8, 10].

From population genetics, the expectation and variance of allele frequency in a population under only genetic drift is given by:

$$E(\alpha_{ij}(t)) = \pi_i$$

$$\text{Var}(\alpha_{ij}(t)) = \pi_i(1-\pi_i)\left[1 - \left(1 - \frac{1}{2N_{ij}}\right)^{t-1}\right]$$

Thus we expect a good simulator of genetic drift to be unbiased for the starting ancestral allele frequency, and to have variance increasing with each generation of evolution t.

We visually checked how allele frequencies evolve over time by examining lineplots of allele frequency over time (Figure 2). This plot shows 2 loci under balancing selection, 2 loci not under selection, and 2 loci under positive selection.

For the loci not under selection, the values of allele frequency rest near the starting allele frequency, and the variance increases over time. Thus, the loci not under selection exhibit the expected characteristics and we can conclude that the simulator works well for these loci.

Similarly, for loci under selection, these plots reveal no signs of departure from the hypotheses of our evolution simulator. That is, high blue allele frequency is clearly favored for blue populations under positive selection. For balancing selection, colored populations evolve toward an allele frequency of 50%, accurately simulating the heterozygote advantage.

These time series plots of allele frequency were useful for visualizing a few loci. However, to verify that all loci behave according to expectations, we used dotplots of final allele frequency (Figure 3).

This dotplot clearly shows that all loci under selection display no signs of selective pressure when in neutral color populations. Inversely, all loci which are not under selection behave similarly, regardless of population color. Also, the symmetry between blue and red alleles is clearly visible. This dotplot efficiently shows that the allele frequencies evolved according to the simulator hypotheses.

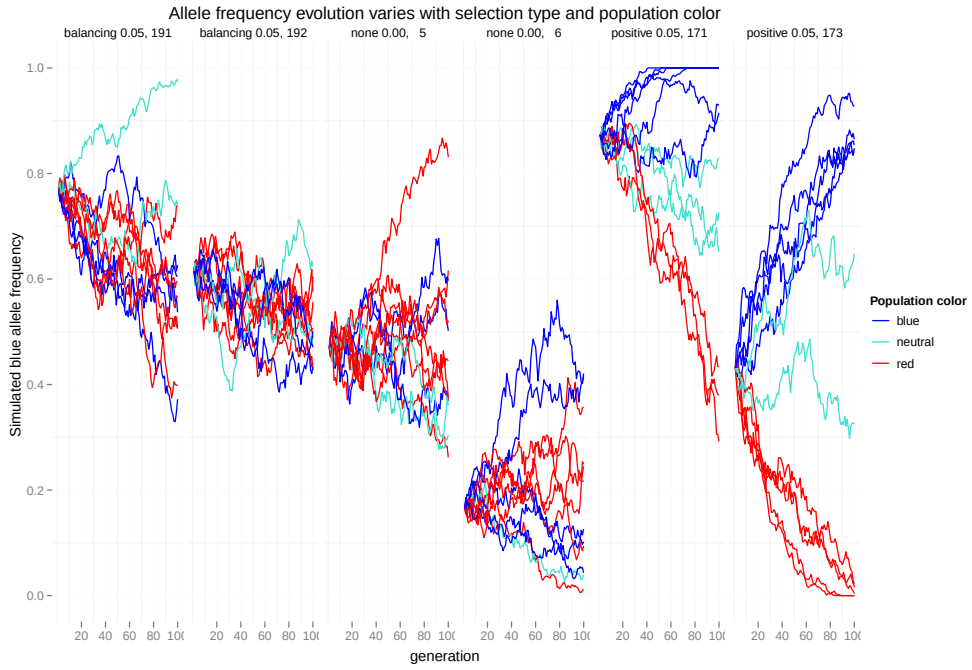


Figure 2: Allele frequency evolution of 6 loci in 12 populations over 100 generations. Each panel represents a different locus, and each line therein represents a different population. Shown are two loci for balancing selection (left), no selection (middle), and positive selection (right).

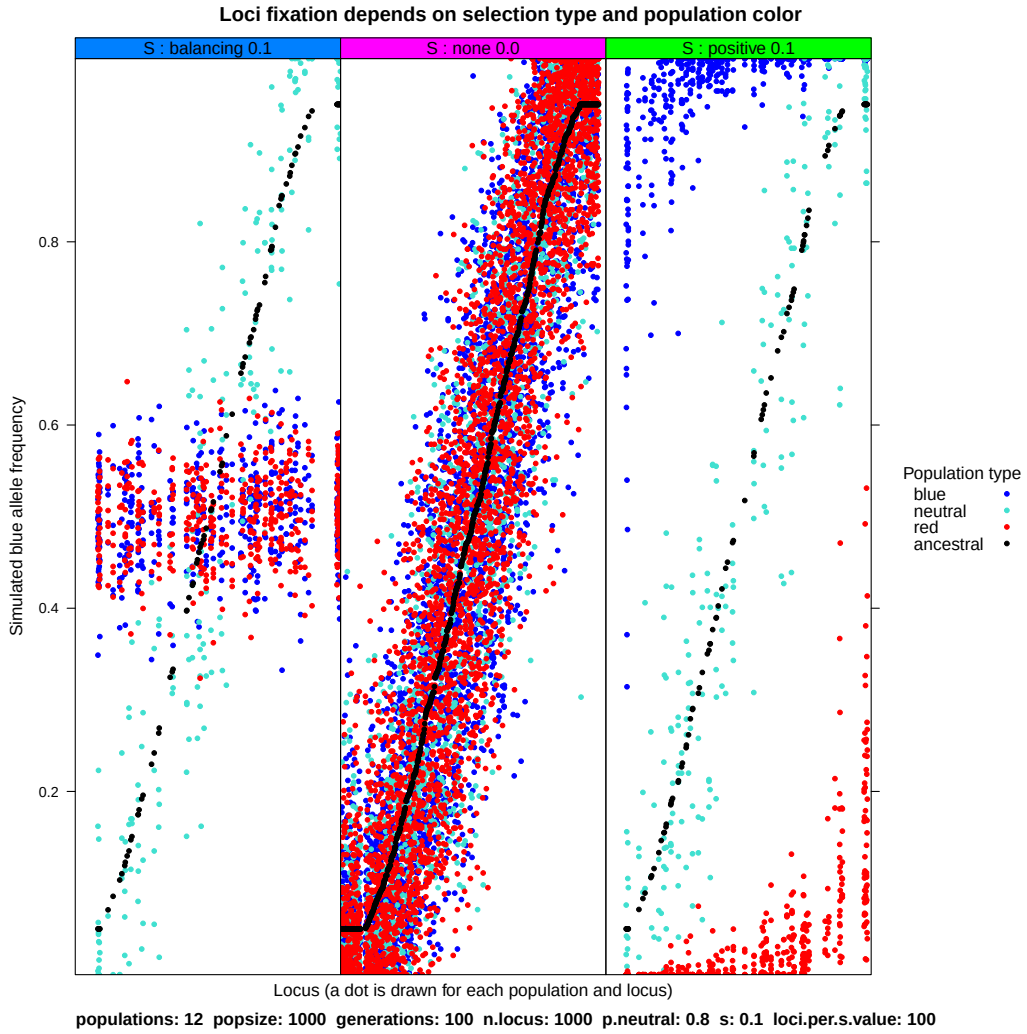


Figure 3: Final simulated blue allele frequency for 1000 loci and 12 populations is shown in a dotplot. Loci are ordered on the horizontal axis by ancestral allele frequency, and then divided into 3 panels by selection state. Note that loci under selection display no signs of selective pressure when in neutral color populations. Inversely, all loci which are not under selection behave similarly, regardless of population color. Also, the symmetry between blue and red alleles is clearly visible.

3.2 Model estimates

We are simulating loci under selection, and analyzing them using the pure-drift Nicholson model. Thus we expect that the loci under selection will not fit the model well.

To diagnose how selection state influences model fit, we plot ancestral allele frequency estimates for each loci versus actual values from the simulation (Figure 4).

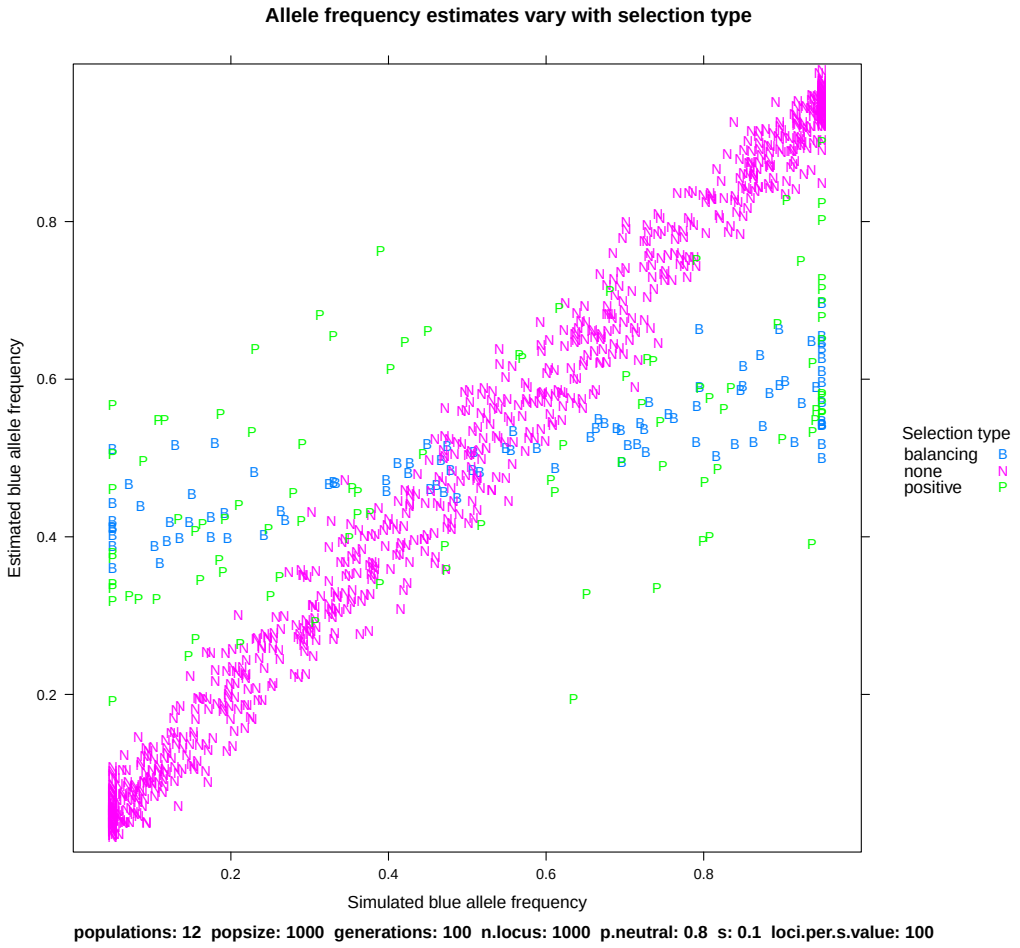


Figure 4: This grouped scatterplot illustrates that model estimates of ancestral allele frequency are not robust to selection.

This scatterplot clearly shows that neutral loci are well estimated by the

model, but loci under balancing and positive selection are not well estimated. This result is sensible in view of the fact that the Nicholson model was designed with only genetic drift in mind. Thus it is expected that model parameters for loci under selection are not estimated well.

To examine the robustness of these ancestral allele frequency estimates to changes in number of populations and number of generations, we did exhaustive simulations of several parameter values. We did 9 simulations, fitting the model for each of them, with 25, 50, or 100 generations, and 4, 8, or 12 populations.

To display the results of the ancestral allele frequency estimates, we used trellised scatterplots of estimates versus actual values (Figure 5).

From what we know about genetic drift, we expect that augmenting the number of generations will increase the variance of the allele frequencies, and thus increase the variance of the ancestral allele frequency estimate. Similarly, we expect that increasing the number of populations will give us more information about the ancestral allele frequency, leading to more accurate estimates.

This series of scatterplots clearly shows that the estimates behave as expected. Less accurate estimates are clearly seen with more generations and fewer populations.

To evaluate if the model estimates of our FORTRAN program agree with model estimates given by WinBUGS, we used both programs to fit the model to a single simulation. Furthermore, Nicholson *et al.* note that the model estimates are robust to changes in the prior distribution of the π_i. To verify this claim, we fit the Nicholson model to a single data set using prior distributions of $\beta(0.7, 0.7)$ and $\beta(1, 1) = U[0, 1]$.

The results of all these model fits are summarized in a scatter plot matrix of ancestral allele frequency estimate versus actual value (Figure 6).

The scatter plot matrix clearly shows that there are no significant differences between models that use the same program. That is, the choice of prior distribution of the ancestral allele frequencies has little influence on model estimates.

However, there appears to be a difference between estimates from WinBUGS and our FORTRAN program, uniquely for loci under positive selection. Our FORTRAN parameter estimates are systematically higher than the corresponding WinBUGS estimates for these loci. However, since these loci are precisely the ones which do not fit the model, it is expected that their estimates may differ.

To investigate the estimates of the population differentiation parameters

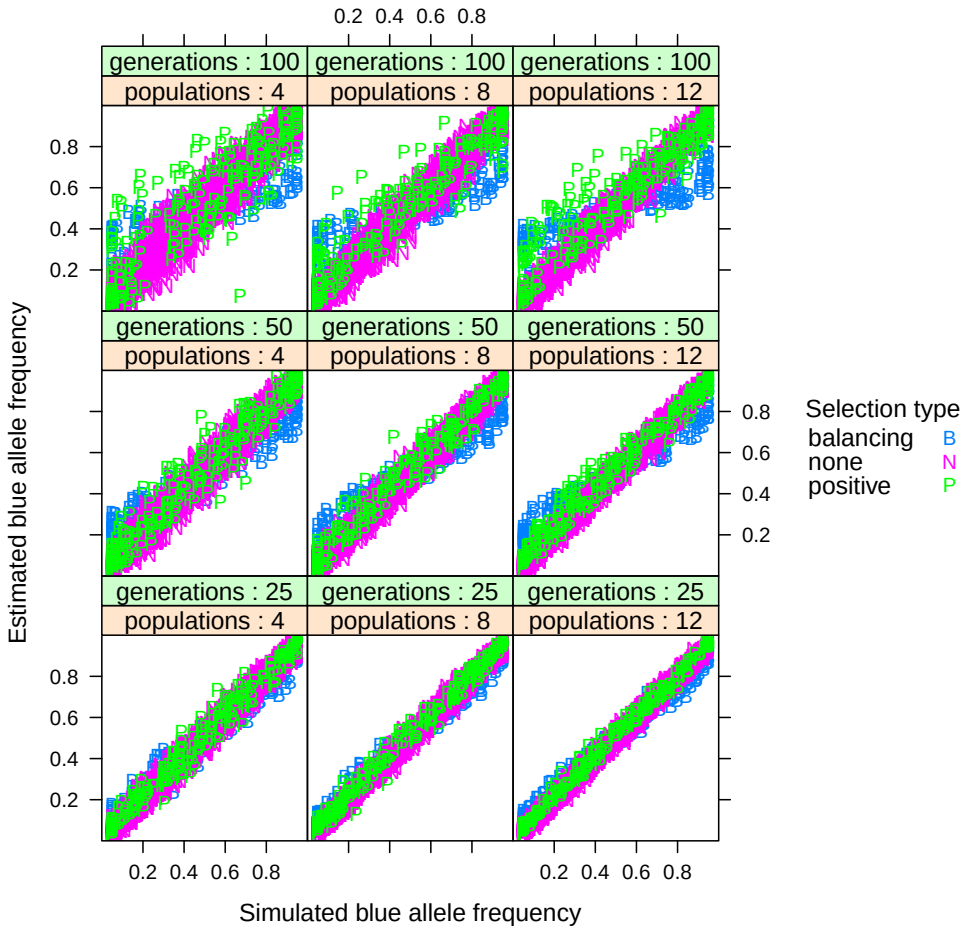


Figure 5: Scatterplots of estimated versus actual ancestral allele frequency, trellised by number of generations and number of populations.

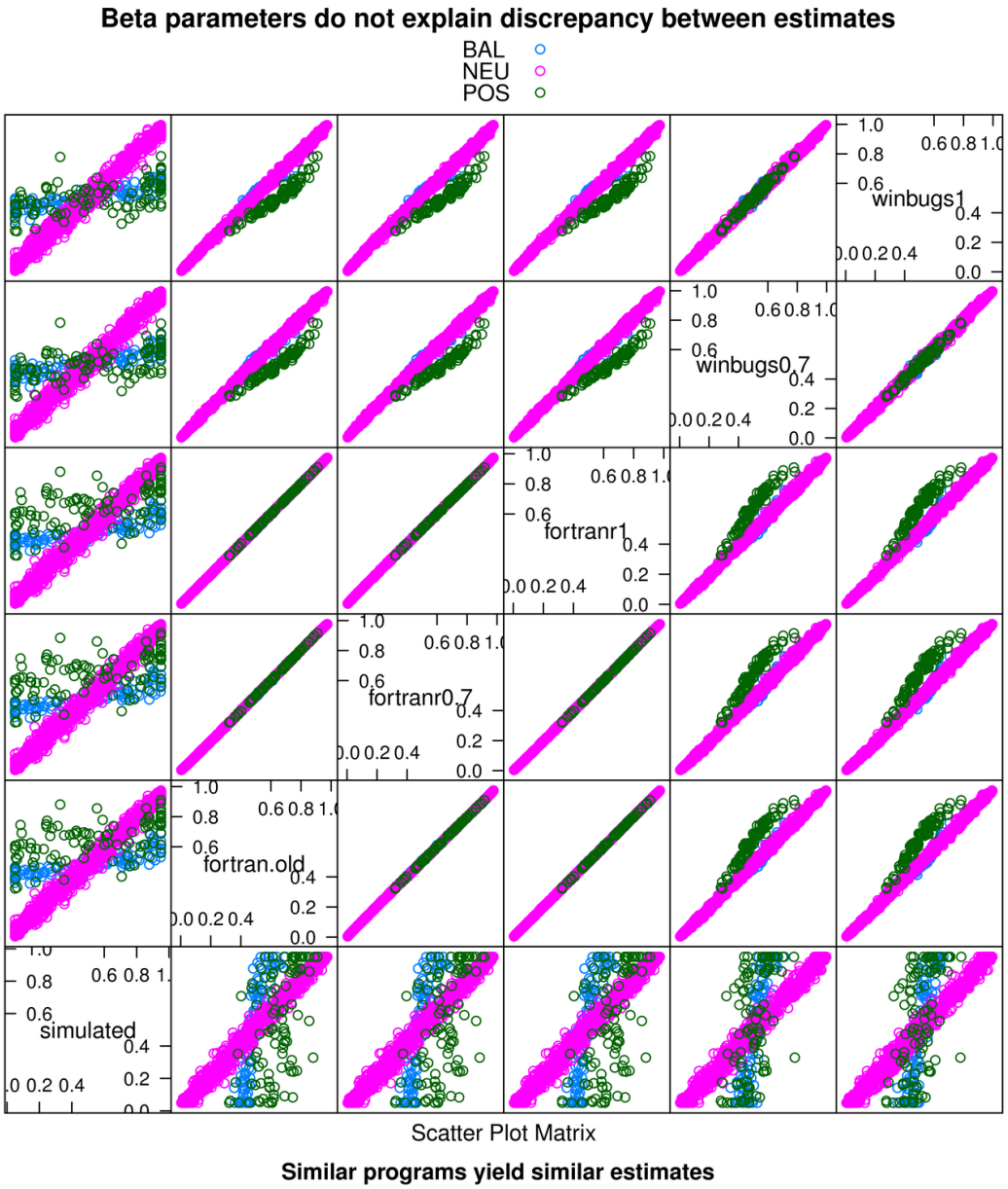


Figure 6: Scatter plot matrix for various values of ancestral allele frequency π_i for a simulated data set (actual simulated values indicated by the row/column labeled simulated). Models using priors that follow a $\beta(1,1)$ (indicated by fortranr1 and winbugs1) and $\beta(0.7,0.7)$ (fortran.old, fortranr0.7, and winbugs0.7) were fit using WinBUGS and our FORTRAN program. For alleles under positive selection, there are small discrepancies between the FORTRAN and WinBUGS programs.

c_j, we first note that from population genetics, we expect that

$$c_j = 1 - (1 - \frac{1}{N_j})^t \approx t/N_j \tag{2}$$

where t is number of generations and N_j is effective population size of population j.

Note that this approximation implies that we expect estimates of c_j to increase linearly over time. To visualize this linear trend, we used lineplots of the estimate differentiation parameter c_j over time t (Figure 7).

Note the linear behavior of the model estimates, as expected. However, the slopes of the lines do not always match the expected theoretical slopes, which can be attributed to approximation errors in Equation 2.

We did simulations with uniform and variable population sizes to determine if c_j estimates were robust to population size fluctuations. By comparing the panels in Figure 7, it is evident that the model fits for population size 1000 fall in the same range for both simulations. Thus we can conclude the model is robust against population size variations.

3.3 Characterization of loci fixation on model fit

The simulation diagnostic dotplots also clearly show that the loci under selection tend to get fixed at frequencies of 0 or 1 by the end of the simulation (Figure 3). These data seem to violate the initial hypothesis that we are dealing with SNP data. That is, SNP discovery is generally performed on small (< 10) numbers of individuals, so data from SNP microarrays is necessarily biased to favor loci which are polymorphic in these individuals. This phenomenon is called the "ascertainment bias" in the literature.

To characterize if the model estimates are sensitive to the ascertainment bias, we fit several models using non-fixed subsets of the loci. The criteria used for calling a locus "fixed" are as follows:

not.all.fixed Throw out the locus if all subpopulations fixed (more stringent criterion; less loci will be "fixed").

none.fixed Throw out the locus if one or more subpopulations fixed (less stringent criterion; more loci will be "fixed").

To evaluate the effect of throwing out these loci on the total number of loci left for input to the model, we made scatterplots of percent of loci "not fixed" versus selection strength s_i (Figure 8). Essentially this told us that loci under strong positive selection tend to be the ones which get called "fixed" and excluded from the model.

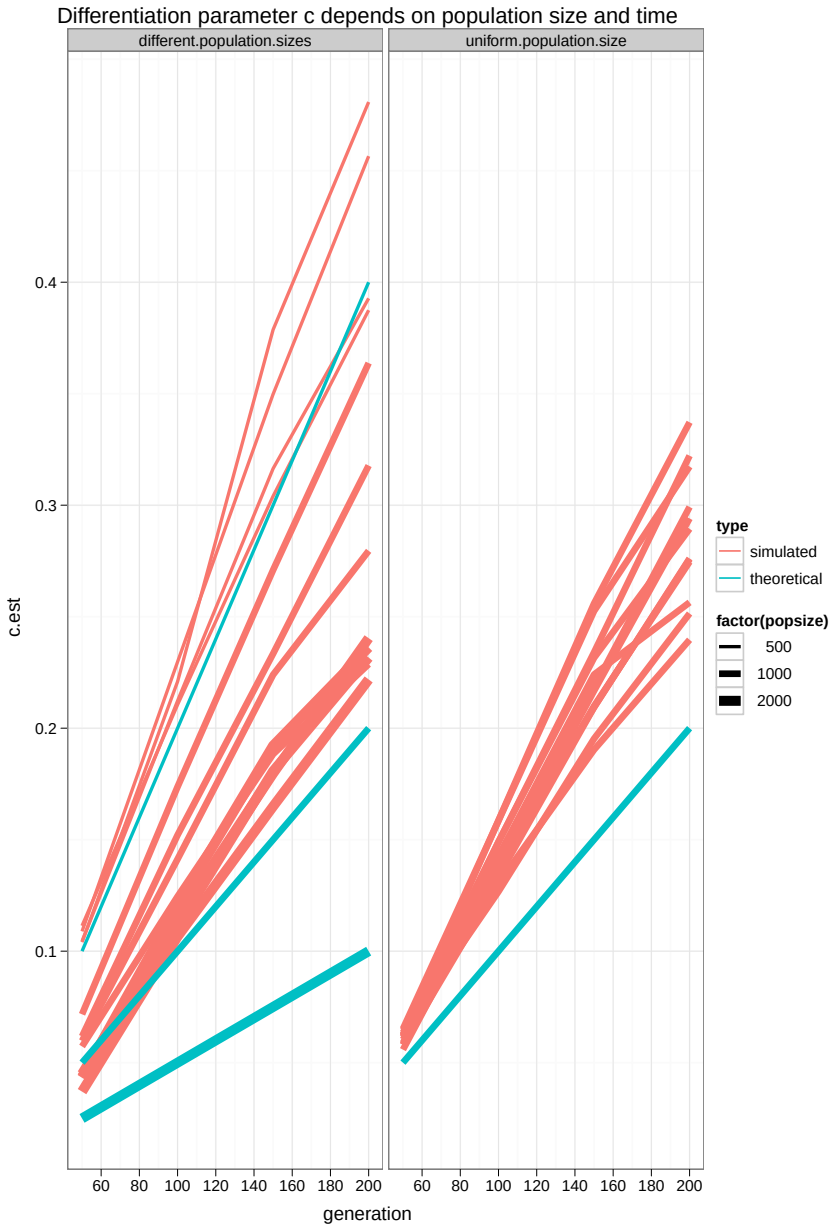


Figure 7: Lineplots of differentiation parameter estimates c_j evolving over time. The model was fit for 4 generations (50,100,150,200). Theoretical line shown in green is t/N_j.

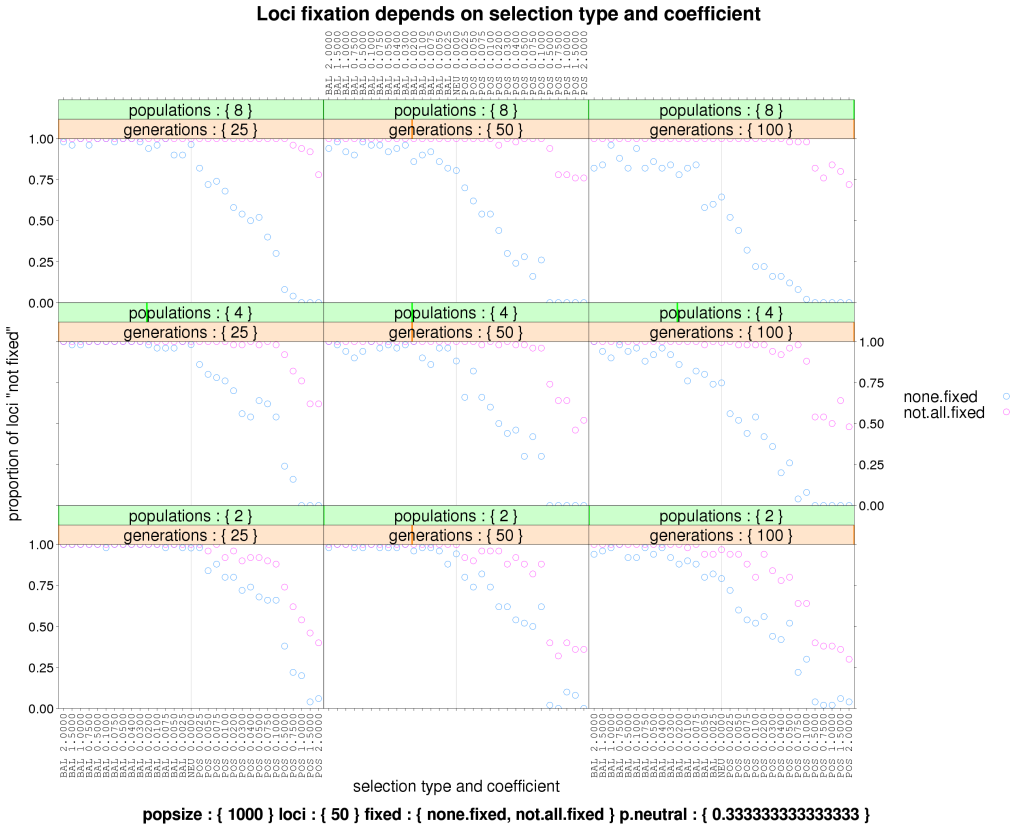


Figure 8: Percent of loci left after throwing out "fixed" loci, according to 2 criteria outlined in the text. Note how loci under strong positive selection are the loci which get excluded.

To examine if there are any large differences between model estimates when fixed data are not included, we made scatterplots of estimated versus simulated ancestral allele frequency π_i values (Figure 9).

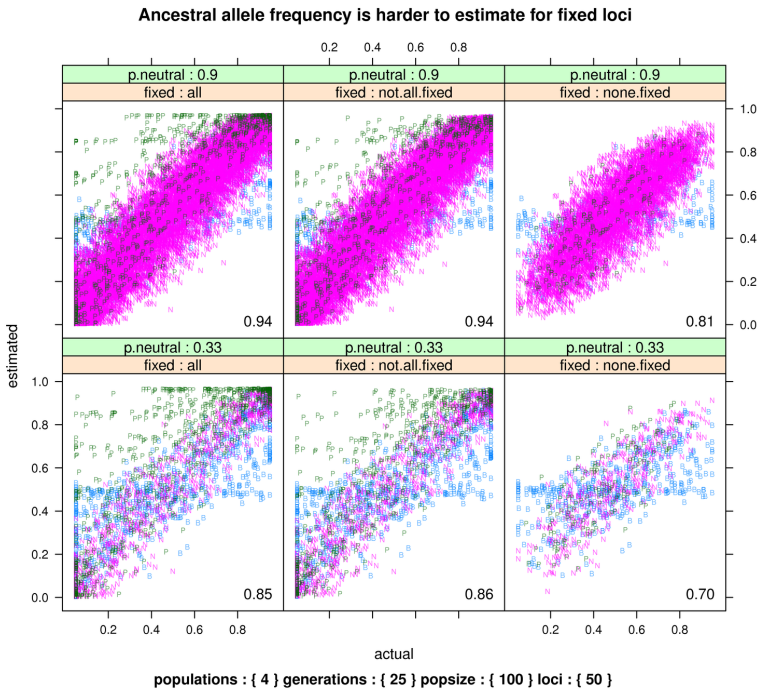


Figure 9: Scatterplots of estimated and simulated ancestral allele frequency. The Nicholson model was fit for all loci, and 2 subsets of loci which were "not fixed" (see text).

This plot indicates that the model fits are similar, regardless of the number of loci included in the dataset. We also compare simulations with large and small proportions of neutral loci. Note that the model estimates behave similarly regardless of the number of loci included in the model. Thus we conclude that the model estimates are robust against the exclusion of fixed loci and the amount of neutral loci.

Thus we can conclude that no harm is done by leaving in the "fixed" loci, and we proceed with the rest of our analyses using all of the simulated loci.

3.4 Simulation summaries using animations

To visualize 3 of the above simulation diagnostics at once, we made combined plots of allele frequency time series, ancestral estimates, and dotplots (Figure 10). We link the plots by showing one locus over time in the upper left, and then highlight this same locus in the other 2 plots.

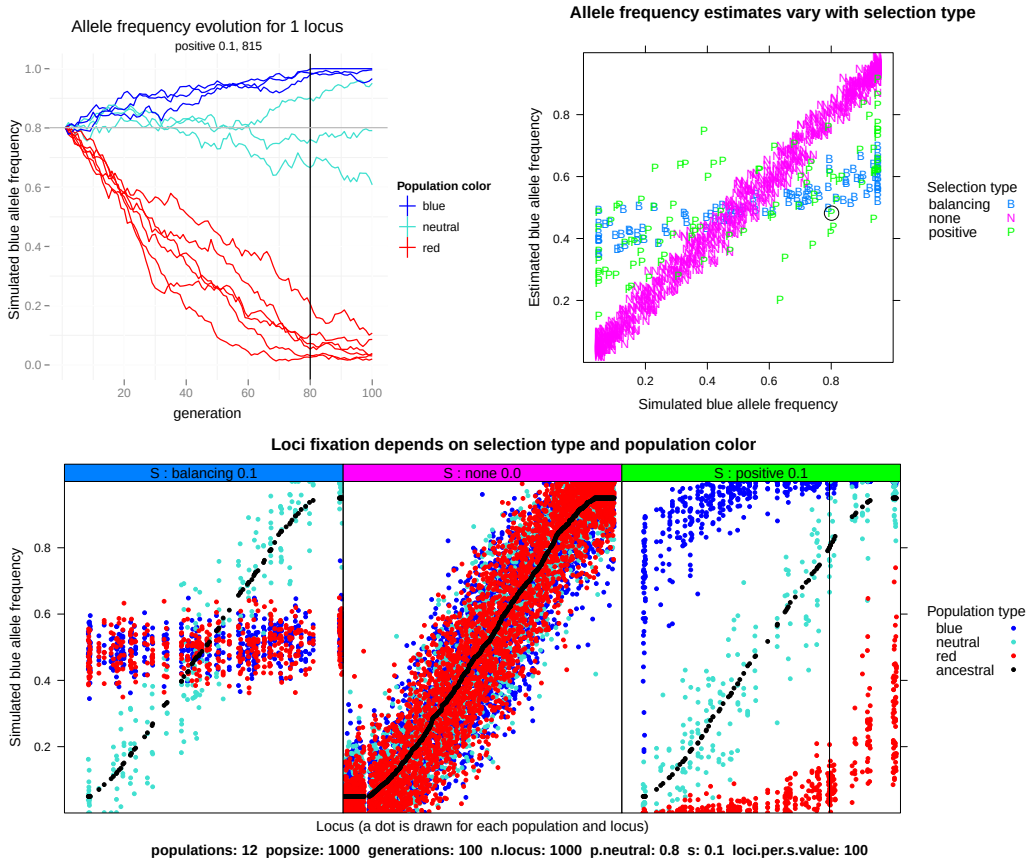


Figure 10: Frame 80/100 of a statistical animation that summarizes the evolution simulation. Note the time series plot for a single locus in the upper left. That same locus is highlighted with a circle in the upper right ancestral estimate plot, and with a vertical line in the bottom dotplot.

To visualize the influence of the number of generations on each of these diagnostic plots, we used to the animation package [12] to create a series of plots, one for each generation. These images are put together and viewed in sequence to form a statistical animation that reveals the dependence on the number of generations. The animations can be viewed on the accompanying

website:

`http://nicholsonppp.r-forge.r-project.org/`

The link between the plots, combined with the animation over several generations, has proven to be a powerful pedagogical device that encourages rapid understanding of the simulator and model hypotheses. Multivariate statistical animations such as this can be useful as teaching tools for students of statistical population genetics.

3.5 Prediction rates of the PPP-value classifier

To evaluate the sensitivity and specificity of the PPP-value classifier, we fit the model on 3 sets of 5 simulations with different parameter values:

Set	Populations	Loci
usual	12	1000
few populations	4	1000
many neutral loci	12	19999

For each of the above parameter sets, we fixed constant parameter values of population size 1000 and 100 generations of evolution. Then we did 5 different simulations with 100 loci each of

$$s_i \in \{0.001, 0.01, 0.032, 0.1, 1\} = \{10^{-3}, 10^{-2}, 10^{-1.5}, 10^{-1}, 10^{0}\}$$

These values were chosen since 1 is a value higher than usually observed in nature, and 0.001 is as weak as if there was no selection at all.

For each of these sets we first made density plots of PPP-value conditional on selection state for each s_i value (Figure 11, Figure 12, Figure 13).

These density plots clearly visualize the different distributions of PPP-values for different selection types. These plots suggest that low PPP-values can be used to indicate positive selection, and high PPP-values can be used to indicate balancing selection.

For the purposes of this work, we will limit ourselves to the classification of a locus as positive or not positive, ignoring balancing selection.

The density plots suggest the following classifier for positive loci:

$$\hat{S}_i(h) = \text{state of locus } i \text{ given threshhold } h = \begin{cases} \text{positive} & \text{if PPP}_i < h \\ \text{none} & \text{if PPP}_i \geq h \end{cases}$$

where h is a PPP-value threshhold that will determine the false positive/false negative rates.

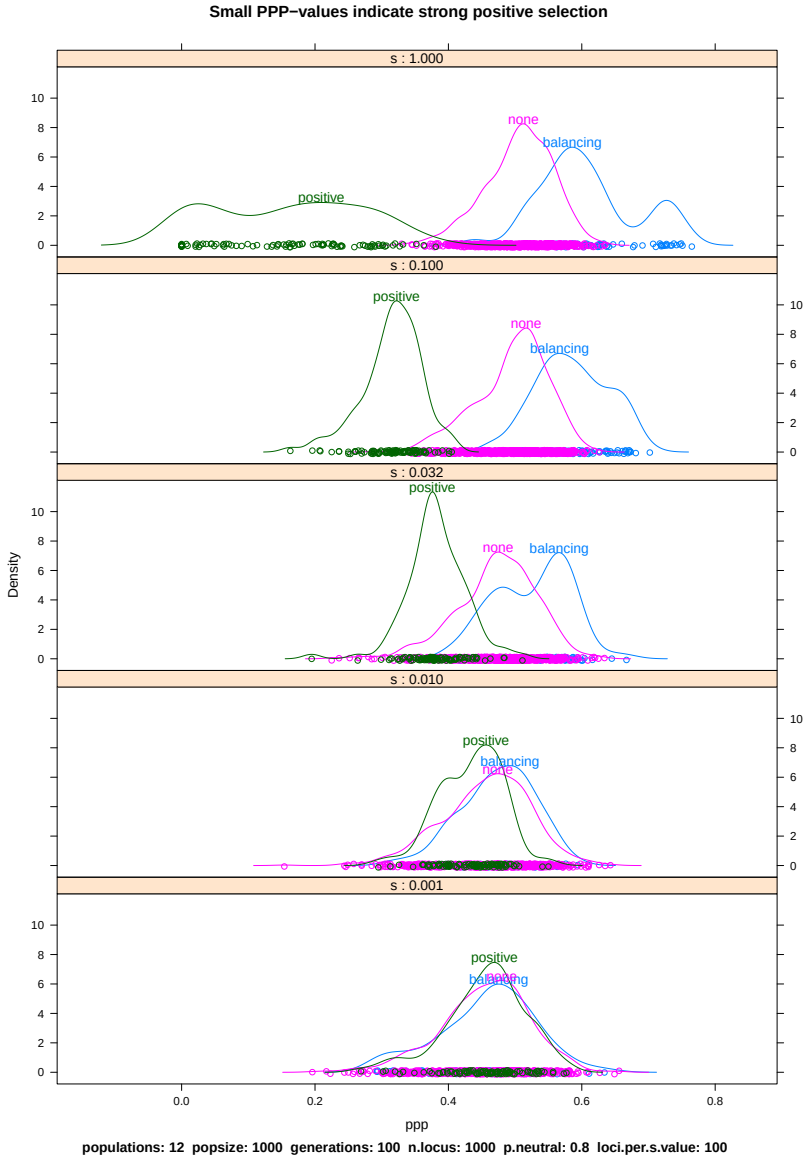


Figure 11: Density estimates for PPP-values of each selection state, given data sets simulated with different selection strengths s_i. Note how it gets easier to distinguish selection as the selection strength parameter increases.

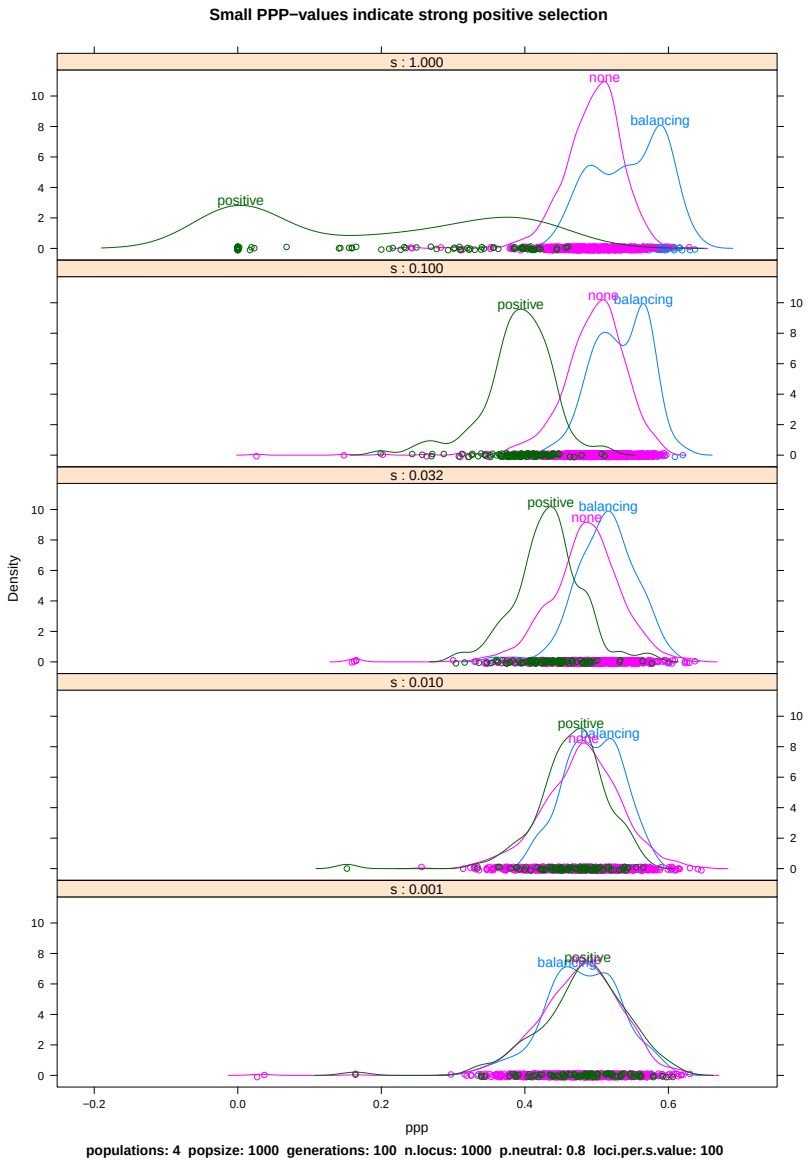


Figure 12: Density estimates for PPP-values, for only 4 populations. With fewer populations it is more difficult to distinguish the behavior of loci under selection.

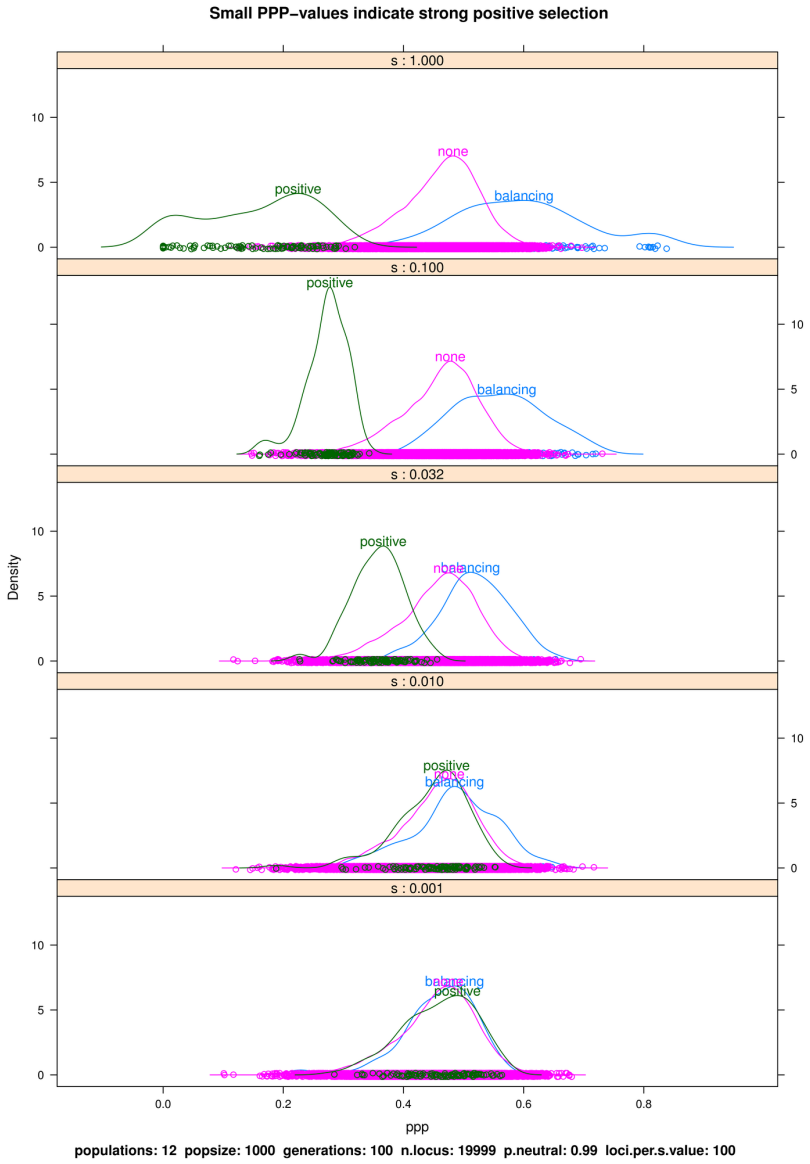


Figure 13: Density estimates for PPP-values, when there is an abundance of neutral loci. In this case the densities are clearly distinguishable, but the sheer number of neutral loci makes a linear cutoff rule suboptimal.

These density plots also clearly show that increasing the value of the selection coefficient s_i tends to increase the separation of distributions of PPP-values.

Examining Figure 12, we see that with fewer populations it is more difficult to distinguish the behavior of loci under selection.

In Figure 13, we see that when there are very many neutral loci the densities of the different selection states are clearly distinguishable, but the sheer number of neutral loci makes a linear cutoff rule suboptimal. For every possible threshhold, there will be a large false positive rate.

Then we evaluated false positive and false negative rates for each possible decision rule; that is, each possible cutoff for the PPP-value. We used lineplots to trace the true positive, false positive, false negative, and incorrect rates as a function of classifier cutoff value (Figure 14, Figure 15, Figure 16).

These plots are used to investigate the best choice of threshhold for the classifier in the simulations. We define the best threshhold h^* as the one which minimizes the empirical risk:

$$h^* = \underset{h \in \{0,1\}}{\operatorname{argmin}} \; P(\hat{S}_i(h) \neq S_i)$$

where S_i is the actual selection state for locus i.

To identify the threshhold values of empirical risk minimization, we traced a horizontal grey line on the minimum value of the incorrect curve, and identified the corresponding threshhold value with a vertical grey line.

Receiver operating characteristics (ROCs) were also traced, to compare all 15 simulations at the same time (Figure 17, Figure 18). In ROCs, we plot lines of sensitivity against $1 -$ specificity for every possible threshhold value of the classifier, where sensitivity is the true positive rate and specificity is the false positive rate.

The ROCs in Figure 17 show that increasing selection strengths s_i tend to increase the area under the curve. This indicates that loci under stronger selection are easier to distinguish from loci not under selection.

In Figure 18, note how the data sets with 4 populations generate decision rules which are noticeably less powerful than those in the other 2 simulations.

Additionally, ROCs were traced for 9 simulations comprising a cross of 3x3 parameter values: 25, 50, and 100 generations; 4, 8, 12 populations (Figure 19).

These ROCs suggest that it is more difficult to accurately predict selection state for a smaller number of generations and populations.

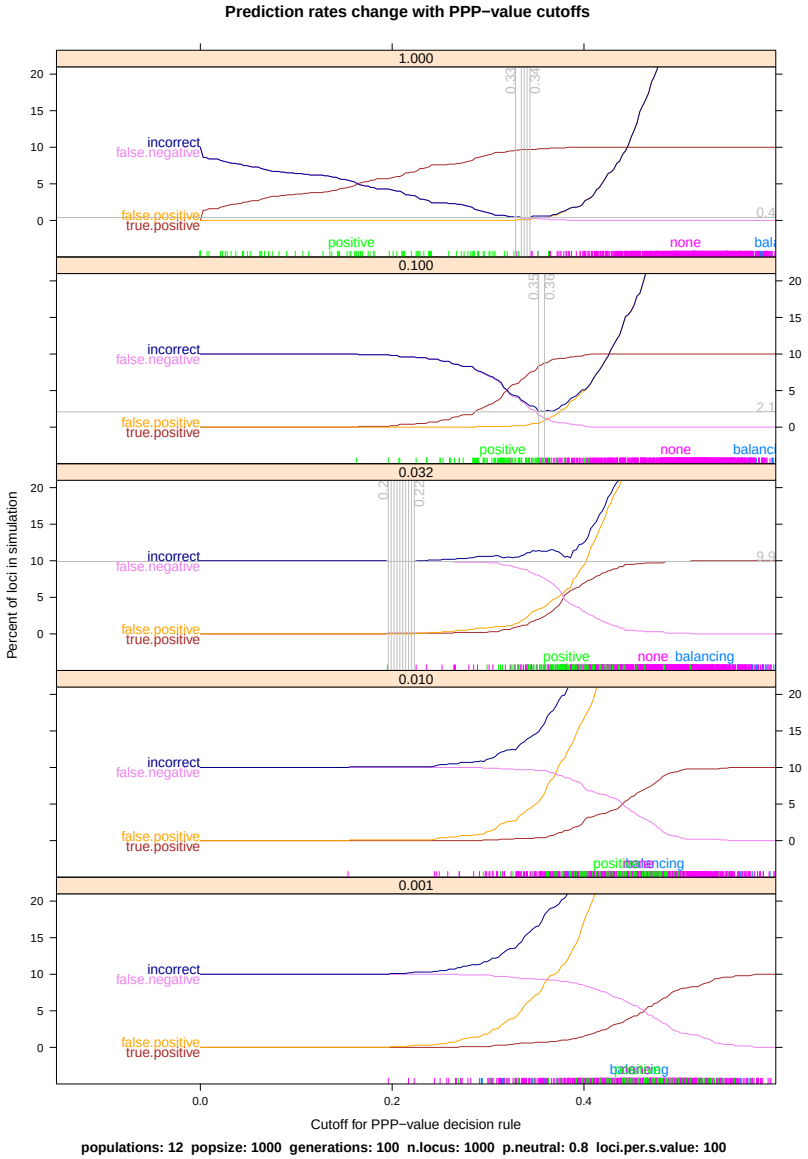


Figure 14: Lineplots of true positive, false positive, and false negative rates using the PPP-value classifier. Note the optimal cutoffs are near 0.35, according to empirical risk minimzation.

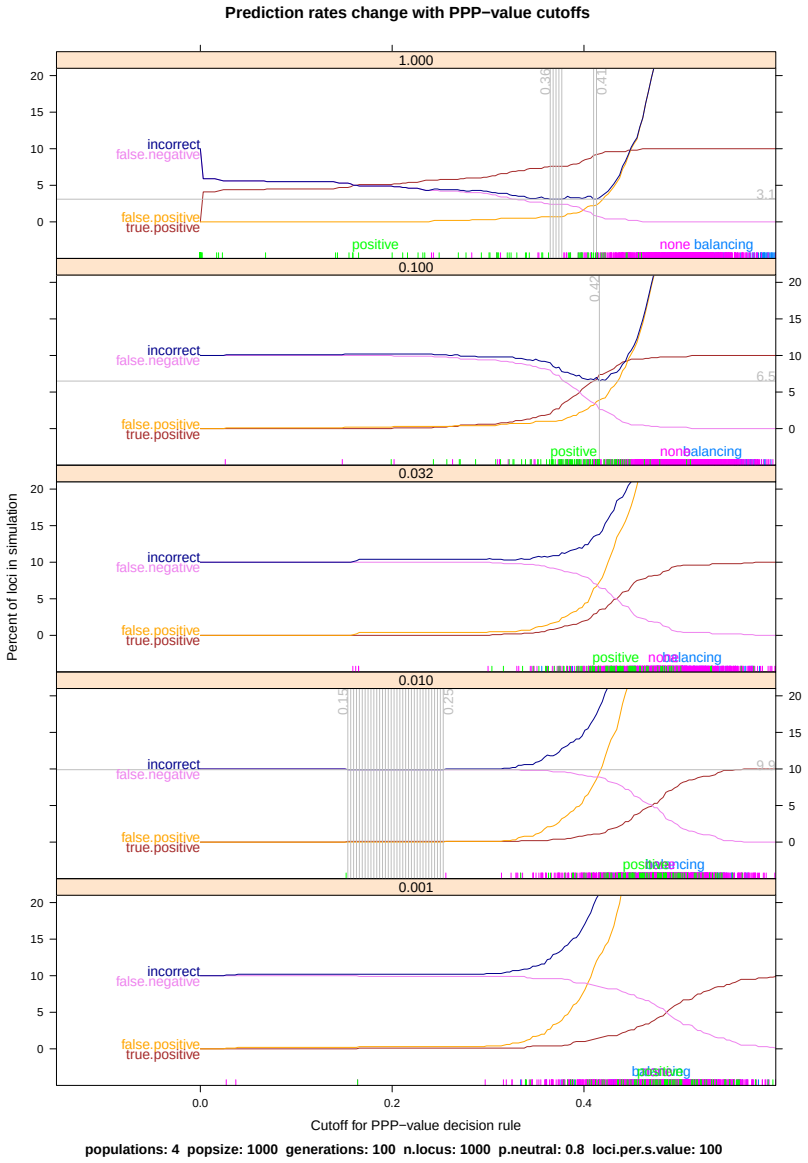


Figure 15: Lineplots of true positive, false positive, and false negative rates using the PPP-value classifier on a data set with few populations. Note that optimal cutoffs are near 0.4, according to empirical risk minimization, but that only high selection values s_i are detected. Best values for incorrect prediction are not as low as in the case where there are 12 populations.

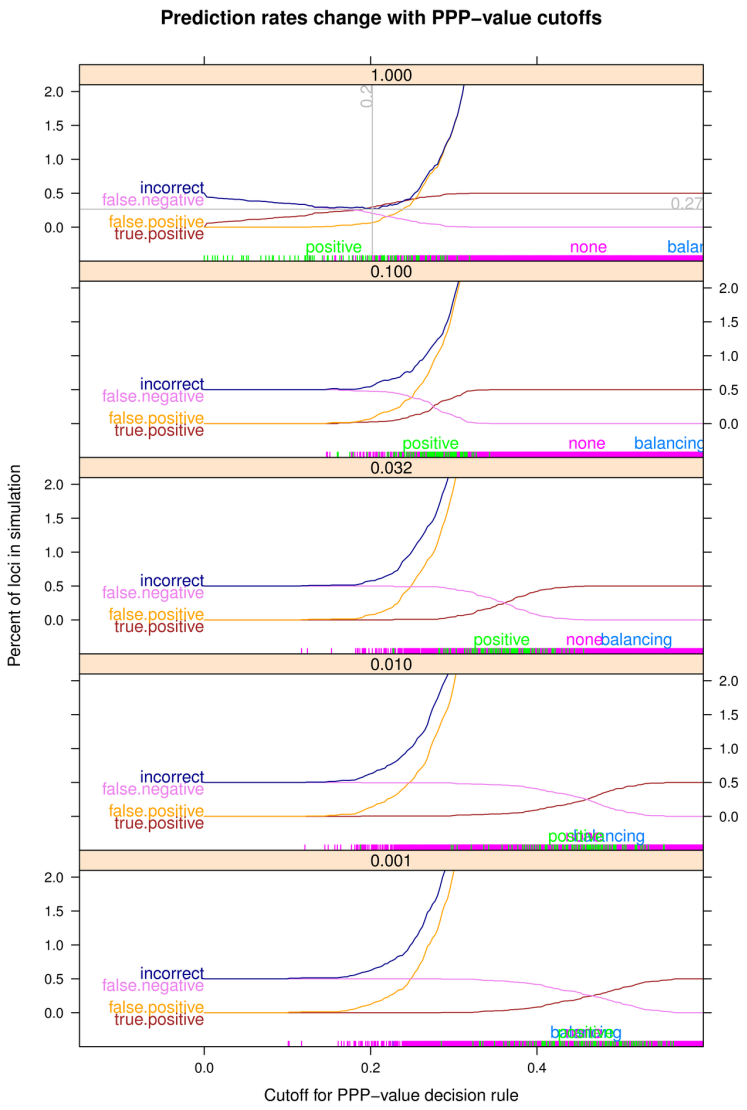


Figure 16: Lineplots of true positive, false positive, and false negative rates using the PPP-value classifier on a data set with many neutral loci. Note that with very many neutral alleles, the rate of false positives ascends very quickly. In this situation, the best cutoff value is around 0.2.

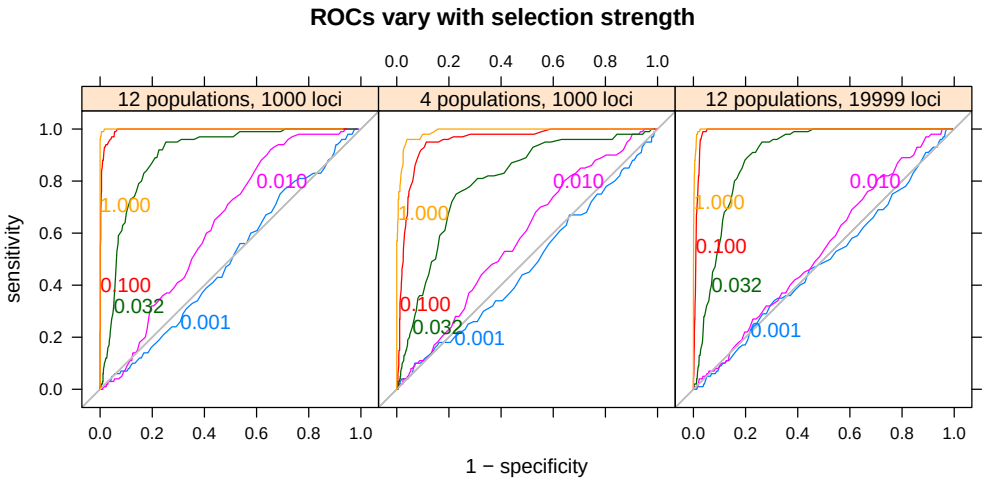


Figure 17: ROCs for several selection strengths, neutral allele concentrations, and population numbers. As shown in the densityplots, increasing selection strengths s_i tend to increase the area under the curve.

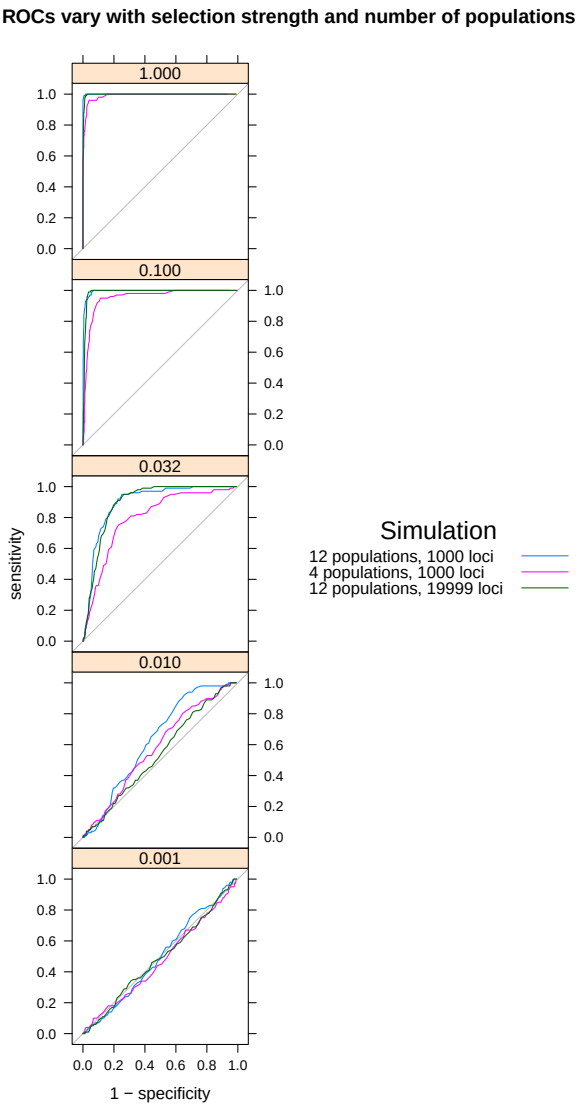


Figure 18: ROCs for several selection strengths, neutral allele concentrations, and population numbers. Note how the data sets with 4 populations generate decision rules which are noticeably less powerful than those in the other 2 simulations.

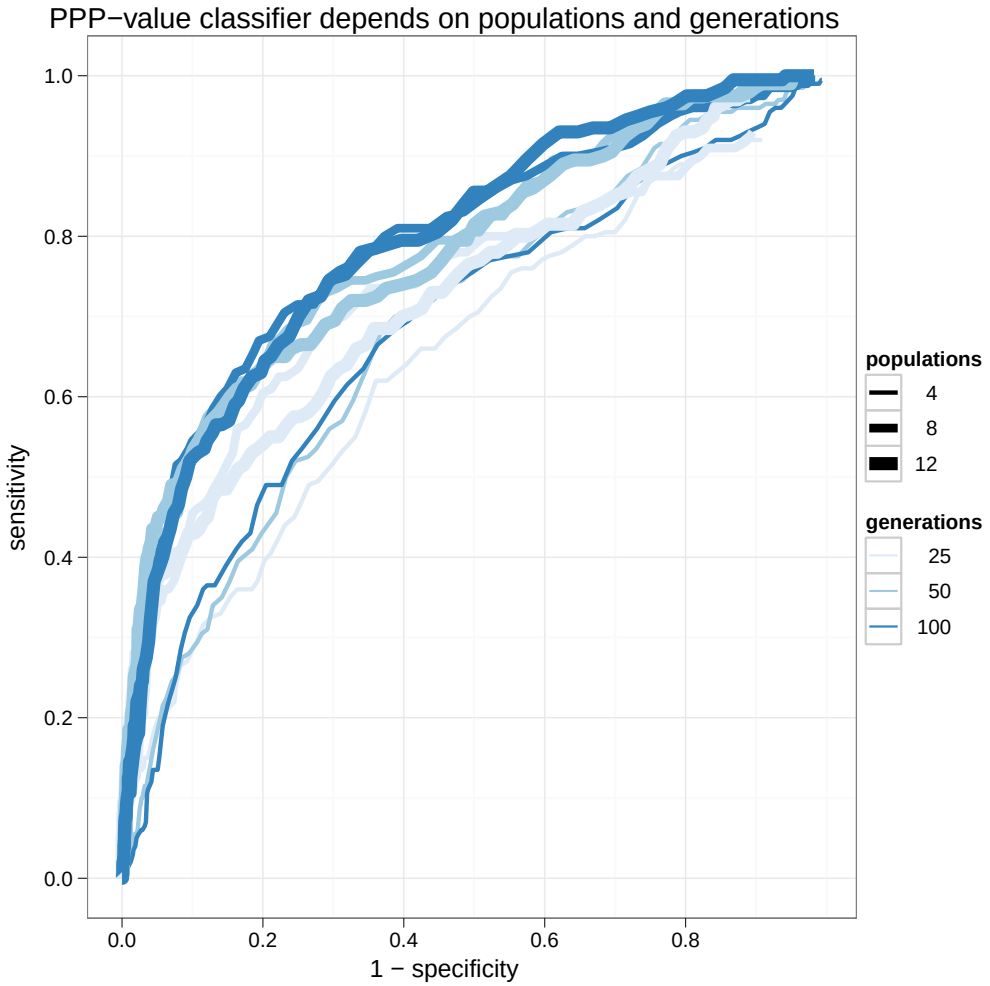


Figure 19: ROCs show slight dependence of the PPP-value classifier on number of populations and generations. It is more difficult to accurately predict selection state for a smaller number of generations and populations.

4 Conclusions and future work

The Nicholson model investigated in this work has been thoroughly tested by exhaustive simulation of genetic drift and selection. The Nicholson model is a simple, robust, and useful Bayesian framework for modeling population divergence under genetic drift from a recent common ancestor.

To extend the Nicholson model to account for loci under selection, we looked for loci which did not fit the Nicholson model well. Our extension of the Nicholson model used PPP-values to classify the selection state of each locus.

The PPP-values are an effective classifier for selection state when the selection coefficient of the locus is sufficiently strong ($s_i > 0.01$). The false positive rate of the classifier drops as the selection strength of the loci increases. But if there are an overwhelming majority of loci not under selection, there will inevitably be high false positive rates.

Several graphical methods were used to visualize the data, including use of statistical animations to understand the behavior of all loci in the simulations. Such methods can be adapted as illustrative teaching tools to facilitate rapid comprehension of these multivariate data.

In this study, we did simulations and sensitivity analysis on every possible threshhold to characterize the classifier. However, with real data sets, we will need a more concrete criterion for choosing the PPP-value threshhold for the classifier.

A hypothesis of the Nicholson model is that all loci are independent. Genetic markers are found on linear chromosomes, so some loci are closer than others. Some markers may even be in the same gene. Thus the hypothesis of loci independence is clearly false, suggesting that some model of correlation between the loci could be introduced into the model.

The Nicholson model supposes that all populations diverge from their common ancestor at the same time, which is false. Thus another beneficial model complication would be to introduce some parameters that model the more tree-like structure of real genetic histories.

To come up with a useful genome annotation method, we would need a method of synthesizing classification of loci into classification of genomic regions. Such methods for combining classifier predictions exist, but would need to be adapted for this particular use with PPP-values.

More work needs to be done to characterize the expected number of false positives and false negatives in a real dataset. In particular, a comparison of this classifier with other existing models of locus selection state should be done.

The model we used only accounts for genetic drift, and detects selection

as aberrations from the model. An enhanced selection state classifier could result if we introduced a more complicated Bayesian model, with parameters for the selection state and/or coefficient.

Finally, we should apply this model to several well-characterized empirical datasets. Due to the portability of the accompanying R package, this should not be a difficult task. The results of our simulations suggest that we will be able to accurately identify loci under strong selection using this method.

5 Acknowledgements

Jean-Louis Foulley and Mathieu Gautier are responsible for the initial development of using PPP-values in the Nicholson model for a selection state classifier [2].

Mathieu Gautier was responsible for the initial implementation of the Nicholson model in FORTRAN. The code used to fit the model in the R package `nicholsonppp` was adapted from his code.

I thank my principal advisor, Mathieu Gautier, for his insightful advice and guidance on this project. I also thank my secondary advisor, Jean-Louis Foulley, for theoretical insights into Bayesian statistics, and in particular, the derivation of the variance used in PPP-value calculations.

The research groups of Génétique Animale et Biologie Intégrative (GABI) and Population, Statistique et Génome (PSGen) at l'Institut National de la Recherche Agronomique (INRA) in Jouy-en-Josas graciously provided office space and computing resources for this project.

Thanks to Gilles Celeux of the SELECT laboratory of the Institut National de Recherche en Informatique et en Automatique (INRIA), who provided funding for this project.

References

[1] Mark A. Beaumont and David J. Balding. Identifying adaptive genetic divergence among populations from genome scans. *Molecular Ecology*, 13:969–980, 2004.

[2] Jean-Louis Foulley and Mathieu Gautier. Detecting selected loci via hierarchical bayesian models and kindred methodology with an application to SNPs in cattle. In *Statistical Methods for Post-genomic Data workshop, SMPGD'09*, AgroParisTech, Paris, January 2009.

[3] A. Gelman, X. L. Meng, and H. S. Stern. Posterior predictive assessment of model fitness via realized discrepancies (with discussion). *Statist. Sinica*, 6:733–760, 1996.

[4] W. R. Gilks, A. Thomas, and D. J. Spiegelhalter. A language and program for complex Bayesian modeling. *The Statistician*, 43:169–178, 1994.

[5] W. K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[6] George Nicholson, Albert V. Smith, Frosti Jónsson, Ómar Gústafsson, Kári Stefánsson, and Peter Donnelly. Assessing population differentiation and isolation from single-nucleotide polymorphism data. *J. R. Statist. Soc.*, 64:695–715, 2002.

[7] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. ISBN 3-900051-07-0.

[8] Deepayan Sarkar. *lattice: Lattice Graphics*, 2009. R package version 0.17-25.

[9] Stefan Theußl and Achim Zeileis. Collaborative Software Development Using R-Forge. *The R Journal*, 1(1):9–14, May 2009.

[10] Hadley Wickham. *ggplot2: An implementation of the Grammar of Graphics*, 2009. R package version 0.8.3.

[11] Sewall Wright. Evolution in mendelian populations. *Genetics*, 16:97–159, 1931.

[12] Yihui Xie. *animation: Demonstrate Animations in Statistics*, 2009. R package version 1.0-4.