

Identifying genetic loci under selective pressure using a posterior predictive p-value classifier

Toby Dylan Hocking

September 24, 2009

Contents

1	Introduction	4
2	Methods	5
2.1	Simulating genetic drift and selection	5
2.2	The hierarchical bayesian Nicholson model	7
2.3	PPP-value calculation theory	8
2.4	PPP-value calculation implementation	8
3	Results	8
3.1	Simulation verification	8
3.2	Characterization of loci fixation on model fit	8
3.3	Model estimates	10
3.4	Simulation summaries using animations	10
3.5	Prediction rates of the PPP-value classifier	13
4	Conclusions and future work	13

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.	6	To diagnose dependence of model fit on selection state, we plot ancestral allele frequency estimates for each loci versus actual values from the simulation. Neutral loci are well estimated by the model, but loci under balancing and positive selection are not well estimated. . . .	11
4	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.	7	Number of generations and populations in the simulation affects model estimates ancestral allele frequency, agreeing with model expectations.	11
5	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). Also shown are large and small proportions of neutral loci. Note that the model estimates behave similarly regardless of the number of loci included in the model. .	8	Scatter plot matrix for various values of ancestral allele frequency π for a simulated data set (actual simulated values indicated by row/column 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.	12
9		10		

9	Lineplots of differentiation parameter estimates c_j evolving over time. The model was fit for 4 generations (50,100,150,200). 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.	12	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.	15
10	The simulation summary diagnostic plot. Note how there is a 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.	13	Note the optimal cutoffs are near 0.35, according to empirical risk minimization. . .	15
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.	14	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.	16
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.	14	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.	16
13	ROC curves 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.	17	ROC curves 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.	17

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. However, 8 and 12 populations seems to behave similarly, suggesting a threshhold for good behavior between 4 and
8.

18

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.* [5], 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 [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

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

populations under genetic drift. However, loci under selection in addition to genetic drift represent departures from the model hypotheses. Thus we use PPP-values estimates from the model to identify these aberrant 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[6]. 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[8]:

<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 $\pi_i = \alpha_{i,1}(1) = \dots = \alpha_{i,P}(1)$, we draw from a truncated $\beta(0.7, 0.7)$ distribution (Fig-

ure 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 (??). 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.

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

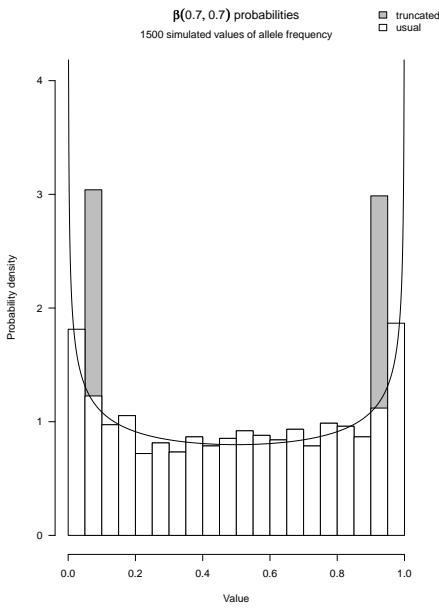


Figure 1: Distribution of ancestral allele frequencies in our simulations follow a truncated $\beta(0.7, 0.7)$ distribution.

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 resistance and sickle-cell anemia affection.

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:

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

We repeat the process for $t = 2, \dots, T$, 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 α_{ij} is the population allele frequency, and N_{ij} is the total number of alleles (red or blue).

This quantity is in turn 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 [5, section 2.2].

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

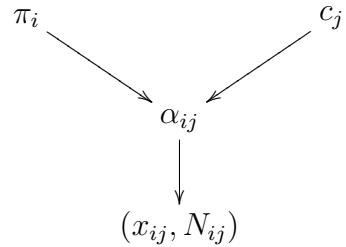
$$\pi_i \sim \beta(a, a)$$

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

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

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 [4].

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

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

The model was first implemented using WinBUGS [3]. To provide speed optimizations for the model fitting, a faster model-fitting program was written in FORTRAN.

2.3 PPP-value calculation theory

The PPP-value for locus i is defined as

$$\text{PPP}_i = P [T_i(y_{ij}^{\text{rep}}, \theta) \geq T_i(y_{ij}^{\text{obs}}, \theta) | y^{\text{obs}}]$$

where $y_{ij} = x_{ij}/N_{ij}$ is the allele frequency for locus i and population j , θ is a vector of parameters, and T_i 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:

$$T_i = \sum_{j=1}^P T_{ij}$$

with

$$T_{ij} = \frac{[y_{ij} - E(y_{ij}|\theta_{ij})]^2}{\text{Var}(y_{ij}|\theta_{ij})}$$

where $\theta_{ij} = (\pi_i, c_j, \sigma_{ij}^2)$ is a vector of parameters and σ^2 is N_{ij} times the sampling variance of the observed allele frequency given its true value p_{ij} so that $\sigma^2 = p_{ij}(1 - p_{ij})$

We define the indicator variable

$$I_i = \begin{cases} 1 & \text{if } T_i(y_{ij}^{\text{rep}}, \theta) \geq T_i(y_{ij}^{\text{obs}}, \theta) \\ 0 & \text{otherwise} \end{cases}$$

2.4 PPP-value calculation implementation

Calculation of PPP-values must be made in the context of sampling from a stationary Markov chain. For each iteration t through the chain, we define this indicator value:

$$p_{ij}^t = \begin{cases} 1 & \text{if } \left(\frac{\text{Bin}(N_{ij}, \alpha_{ij}^t)}{N_{ij}} - \pi_i^t\right)^2 > \left(\frac{Y_{ij}}{N_{ij}} - \pi_i^t\right)^2 \\ 0 & \text{otherwise} \end{cases}$$

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

Thus the PPP-value for locus i is given by

$$p_i = \frac{1}{PT} \sum_{j=1}^P \sum_{t=1}^T p_{ij}^t$$

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 [7, 9].

First, we checked how allele frequencies evolve over time for just a few loci (Figure 2). The loci examined show no signs of departure from the hypotheses of our evolution simulator.

However, to verify that all loci behave according to expectations, we used dotplots of final allele frequency (Figure 3). These dotplots efficiently show that the loci evolved according to the simulator hypotheses.

3.2 Characterization of loci fixation on model fit

The dotplots above also clearly show that the loci under selection tend to get fixed at frequencies of 0 or 1 by the end of the simulation. These data seem to violate the initial hypothesis that we are dealing with

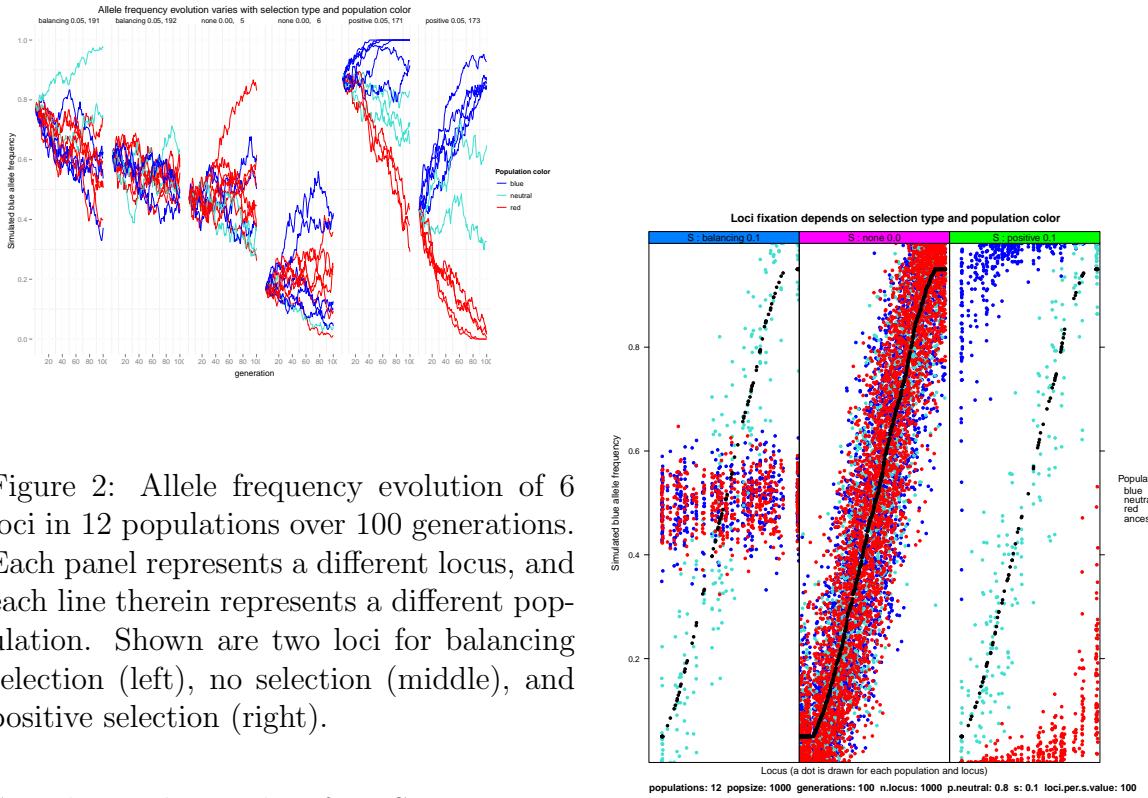


Figure 2: Allele frequency evolution of 6 loci in 12 populations over 100 generations. Each panel represents a different locus, and each line therein represents a different population. Shown are two loci for balancing selection (left), no selection (middle), and positive selection (right).

SNP data. That is, data from SNP microarrays is necessarily biased to favor loci which we have already observed are polymorphic. This phenomenon is called the “recruitment bias” in the literature.

To characterize if the model estimates are sensitive to the recruitment 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

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.

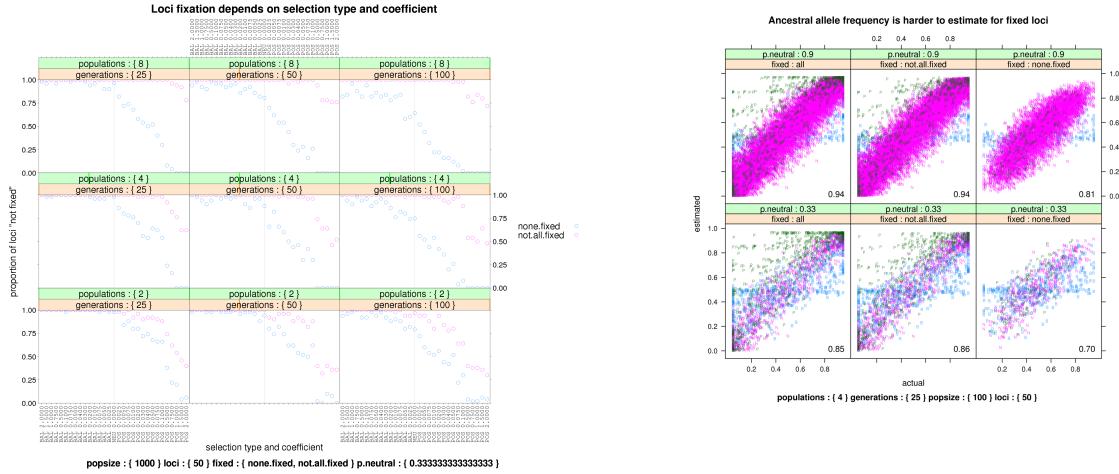


Figure 4: Percent of loci left after throwing out “fixed” loci, according to 2 criteria outlined in the text. Note how loci under strong positive selection are the loci which get excluded.

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 4). Essentially this told us that loci under strong positive selection tend to be the ones which get called “fixed” and excluded from the model.

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 5). This plot indicates that the model fits are similar, regardless of the number of loci included in the dataset.

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.

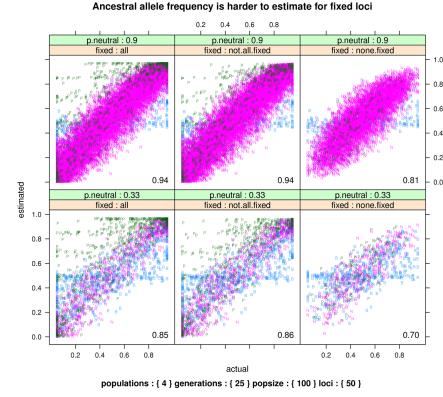


Figure 5: Scatterplots of estimated and simulated ancestral allele frequency. The Nicholson model was fit for all loci, and 2 subsets of loci which were “not fixed” (see text). Also shown are large and small proportions of neutral loci. Note that the model estimates behave similarly regardless of the number of loci included in the model.

3.3 Model estimates

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.

To visualize the influence of the number of generations on each of these diagnostic plots, we used the animation package[10] 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.org>

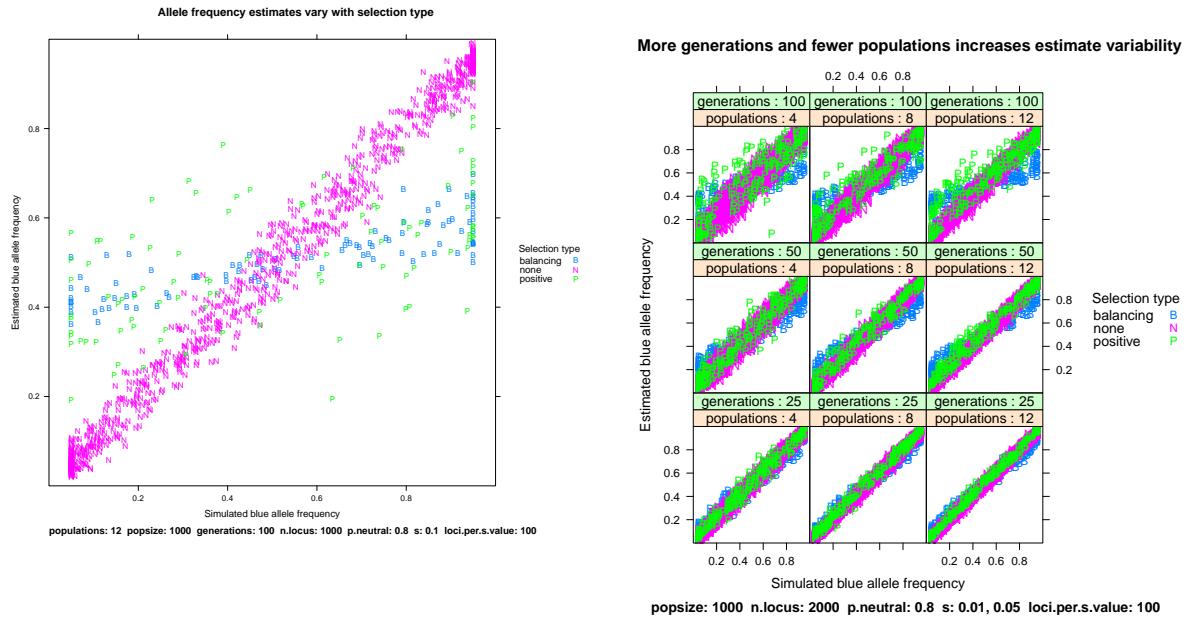


Figure 6: To diagnose dependence of model fit on selection state, we plot ancestral allele frequency estimates for each loci versus actual values from the simulation. Neutral loci are well estimated by the model, but loci under balancing and positive selection are not well estimated.

Figure 7: Number of generations and populations in the simulation affects model estimates ancestral allele frequency, agreeing with model expectations.

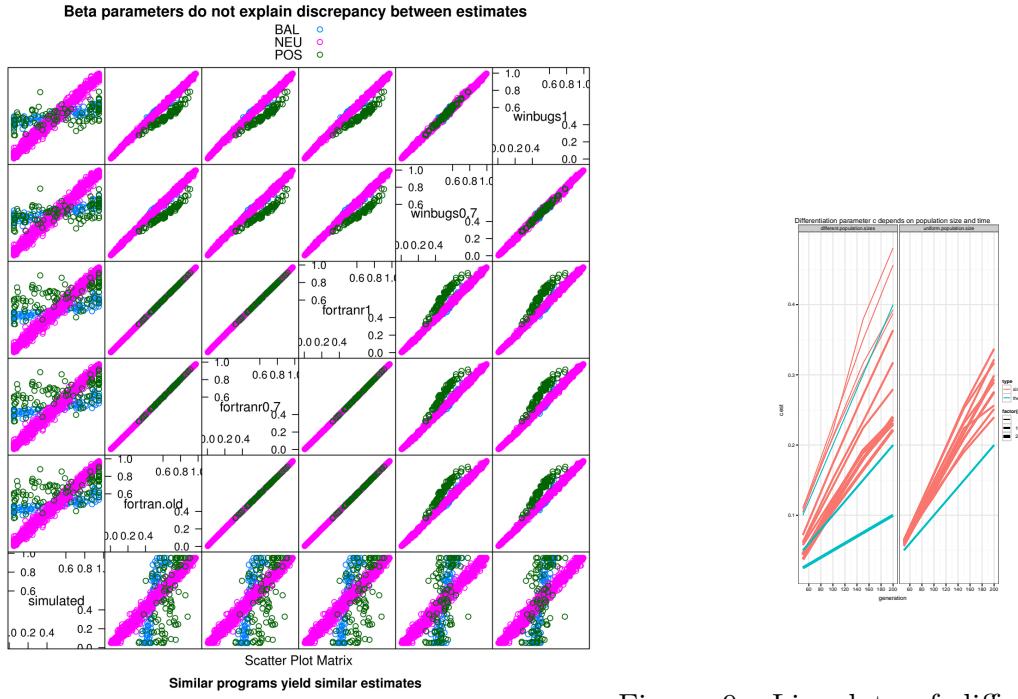


Figure 8: Scatter plot matrix for various values of ancestral allele frequency π for a simulated data set (actual simulated values indicated by row/column simulated). Models using priors that follow a $\beta(1, 1)$ (indicated by fortranr1 and winbugs1) and $\beta(0.7, 0.7)$ (fortran.old, fortranr0.7, 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.

Figure 9: Lineplots of differentiation parameter estimates c_j evolving over time. The model was fit for 4 generations (50, 100, 150, 200). 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.

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 $s_i \in \{0.001, 0.01, 0.032, 0.1, 1\}$.

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). Then we evaluated false positive and false negative rates for each possible decision rule; that is, each possible cutoff for the PPP-value (Figure 14, Figure 15, Figure 16).

ROCs were also traced, to compare all 15 simulations at the same time.

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).

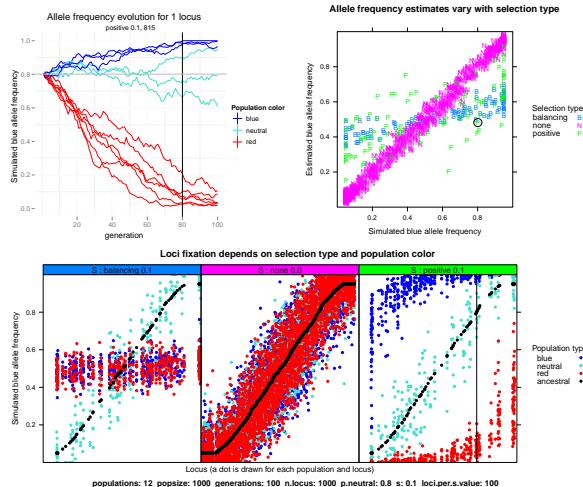


Figure 10: The simulation summary diagnostic plot. Note how there is a time series plot for a single locus in the upper left. That same locus is highlighted with a circle in the upper right ancestral estimate plot, and with a vertical line in the bottom dot-plot.

4 Conclusions and future work

More work needs to be done to characterize the expected number of false positives and false negatives in a real dataset.

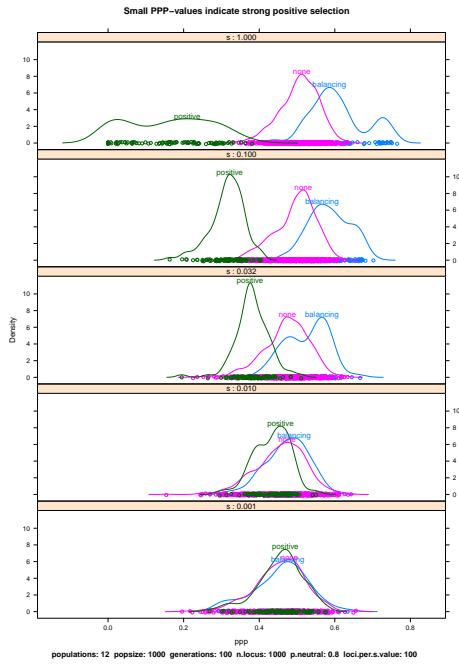


Figure 11: Density estimates for PPP-values of each selection state, given data sets simulated with different selection strengths s_i . Note how it gets easier to distinguish selection as the selection strength parameter increases.

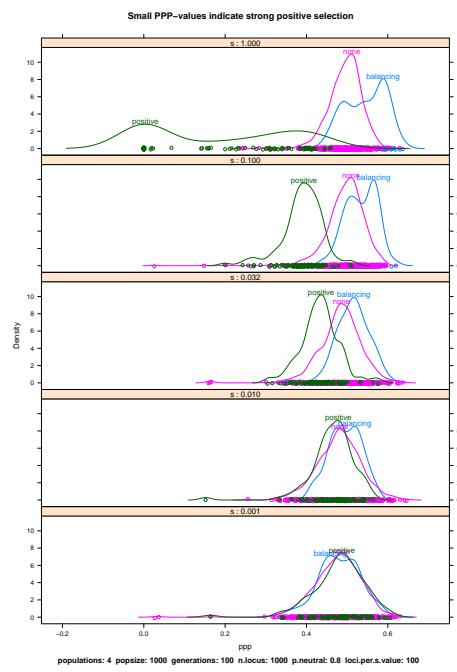


Figure 12: Density estimates for PPP-values, for only 4 populations. With fewer populations it is more difficult to distinguish the behavior of loci under selection.

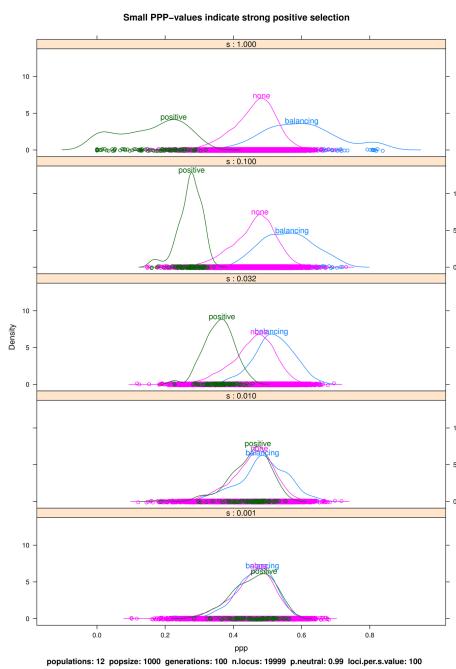


Figure 13: Density estimates for PPP-values, when there is an abundance of neutral loci. In this case the densities are clearly distinguishable, but the sheer number of neutral loci makes a linear cutoff rule suboptimal.

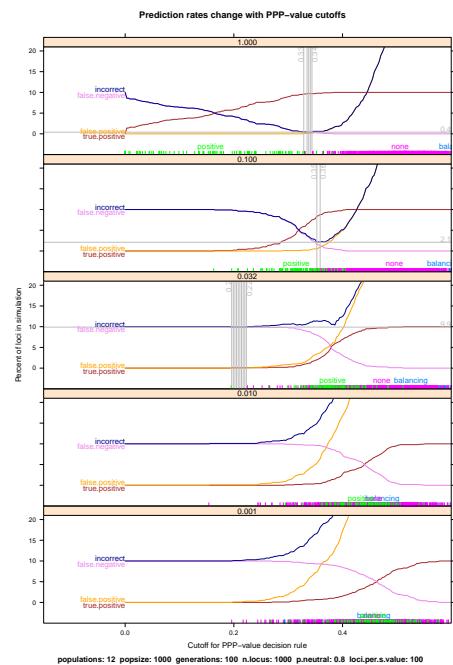


Figure 14: Note the optimal cutoffs are near 0.35, according to empirical risk minimization.

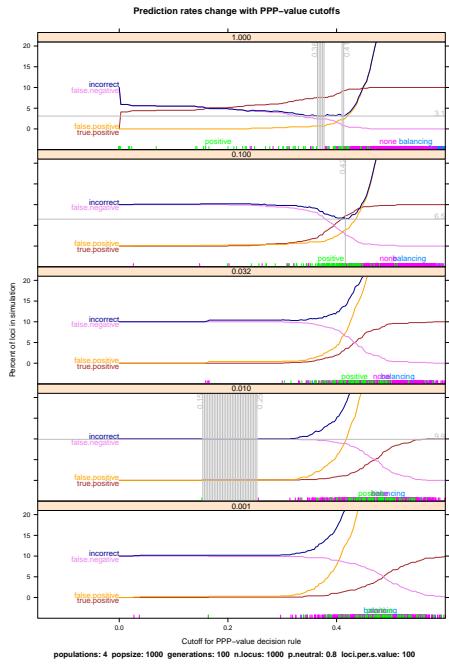


Figure 15: Note that optimal cutoffs are near 0.4, according to empirical risk minimization, but that only high selection values s_i are detected. Best values for incorrect prediction are not as low as in the case where there are 12 populations.

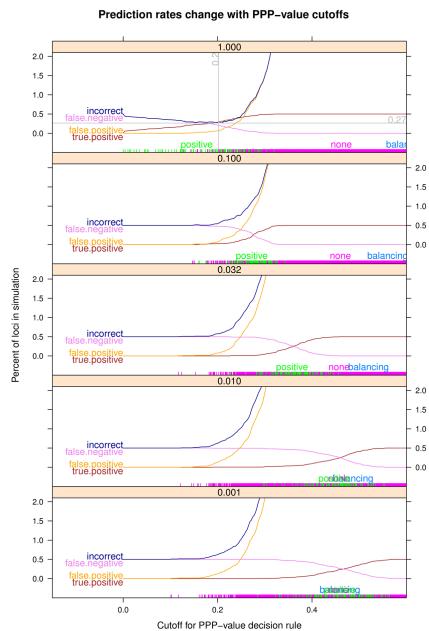


Figure 16: Note that with very many neutral alleles, the rate of false positives ascends very quickly. In this situation, the best cutoff value is around 0.2.

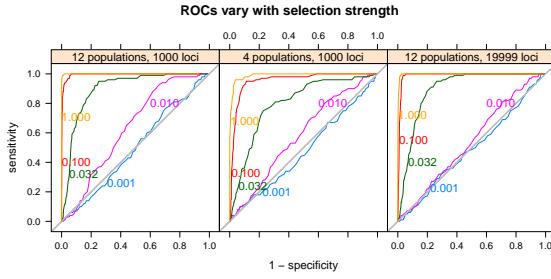


Figure 17: ROCs for several selection strengths, neutral allele concentrations, and population numbers. As shown in the densityplots, increasing selection strengths s_i tend to increase the area under the curve.

We should apply this model to a real dataset. It will be very easy since the interface to the model uses R.

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

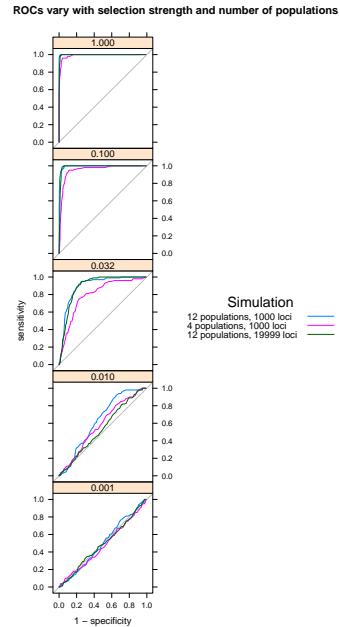


Figure 18: ROCs for several selection strengths, neutral allele concentrations, and population numbers. Note how the data sets with 4 populations generate decision rules which are noticeably less powerful than those in the other 2 simulations.

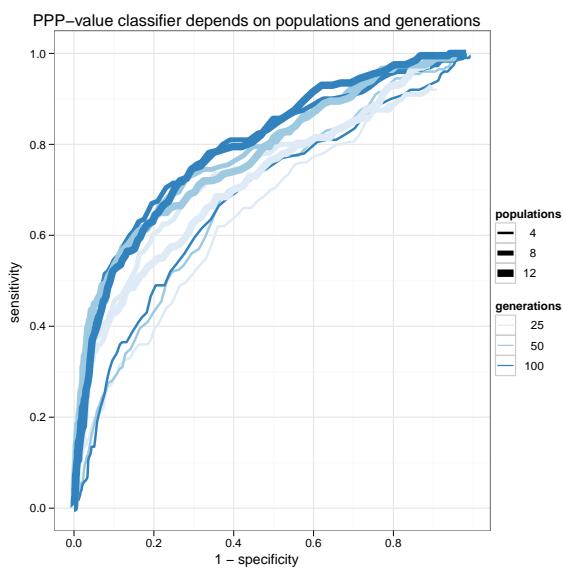


Figure 19: ROCs show slight dependence of the PPP-value classifier on number of populations and generations. It is more difficult to accurately predict selection state for a smaller number of generations and populations. However, 8 and 12 populations seems to behave similarly, suggesting a threshold for good behavior between 4 and 8.

- [4] W. K. Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- [5] 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.
- [6] 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.
- [7] Deepayan Sarkar. *lattice: Lattice Graphics*, 2009. R package version 0.17-25.
- [8] Stefan Theußl and Achim Zeileis. Collaborative Software Development Using R-Forge. *The R Journal*, 1(1):9–14, May 2009.
- [9] Hadley Wickham. *ggplot2: An implementation of the Grammar of Graphics*, 2009. R package version 0.8.3.
- [10] Yihui Xie. *animation: Demonstrate Animations in Statistics*, 2009. R package version 1.0-4.