Identifying loci under selective pressure using a PPP-value classifier

Toby Dylan Hocking

September 22, 2009

1 Introduction

The explosion of molecular marker data in animal populations from technologies such as Single Nucleotide Polymorphism (SNP) microarrays 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. One area 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 complex Bayesian models of evolutionary systems. In this work, we investigate one such Bayesian model used to describe evolution of domestic animation 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.*(2002)[3], hereafter referred to as the Nicholson model. This model attempts to estimate ancestral population allele frequencies for each marker, and a measure of divergence for each population which is analogous to the classical F_{st} of population genetics.

The new idea put forth in this work is a classifier for markers under selection, based on Posterior Predictive P-values (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 evolution 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 aberrent loci.

Furthermore, we validate the model using a simulation of evolution by genetic drift and selection. The simulator has been implemented using the R programming language[4]. 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[6]:

`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 $\pi_i = \alpha_{1,j}(1) = ... = \alpha_{L,j}(1)$, we draw from a truncated $\beta(0.7, 0.7)$ distribution Figure 1.

If we assign the colors blue and red to the 2 alleles, and we define $\alpha_{ij}(t)$ is the blue allele frequency, 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

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

frequency, independent of population color:

$$\alpha^*_{ij}(t) = \text{rbinom}(\text{popsize}, \alpha_{ij}(t-1))$$

The effect of drift grows more important relative to selection as population size popsize 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^{BB}_{ij}	w^{BR}_{ij}	w^{RR}_{ij}	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:

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

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 alleles is modeled as $x_{ij} \sim \text{Binomial}(\text{popsize}, \alpha_{ij})$, where α_{ij} is the population allele frequency for locus i and population j. 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 [3, section 2.2]. The distributions of the hyperparameter for ancestral allele frequency follows $\pi_i \sim \beta(a, a)$ and the population divergence parameter follows $c_j \sim U[0, 1]$.

The relationship between model parameters is more clearly summarised in the following diagram:

$$\begin{tikzcd}
\pi_i \arrow[dr] & & c_j \arrow[dl] \\
& \alpha_{ij} \arrow[d] & \\
& (X_{ij}, N_{ij}) &
\end{tikzcd}$$

MCMC techniques are used to sample from the following posterior distributions:

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

2.3 PPP-value calculation

Calculation of PPP-values in the Markov Chain is used for the classifier. For each iteration t through the chain, we say the data fits the model if the following criterion is met:

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

Thus the PPP-value for locus i is given by

$$p_i = \frac{1}{T} \sum_{t=1}^{T} p_i^t$$

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 [5, 7].

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.

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.

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 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").

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.

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 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.

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.

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.

3.3 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 6: 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.

Figure 6.

To visualize the influence of the number of generations on each of these diagnostic plots, we used to the animation package[8] 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/`

3.4 Model estimates

3.5 Sensitivity of the PPP-value classifier

4 Conclusions

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] 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.

[4] 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.

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

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

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

Figure 7: 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 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.

Differentiation parameter c depends on population size and time
different.population.sizes
uniform.population.size
c.est
0.1
0.2
0.3
0.4
60
80
100
120
140
160
180
200
generation
type
simulated
theoretical
factor(popsize)
500
1000
2000

Figure 9: foo

Figure 10: foo

Figure 11: foo

Figure 12: foo

Figure 13: foo

Figure 14: foo

Figure 15: foo

Figure 16: foo

Figure 17: foo

Figure 18: foo

Figure 19: foo

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