Implementing FastSTRUCUTRE Algorithm with Linked Loci Extension

Michael Li

Introduction

The STRUCTURE algorithm created by Pritchard et al. in 2000 is a model-based clustering algorithm for inferring population structure from multilocus genotype. This method, which was later improved upon with extensions such as the linked loci model published by Falush et al. in 2003 has made a significant impact on the biological research community. It can be used for inferring and assigning individuals to distinct populations, studying hybrid zones, identifying admixed individuals, and estimating population allele frequencies from data with a significant amount migrant or admixed samples.

The goal of this project is to reimplement the STRUCUTRE algorithm and its linked loci extensions, which improves population inference, allows for the detection of admixture events farther in the past, and more accurately estimates statistical uncertainty. The models will be tested using real sequenced genotype data from dwarf and normal sized whitefish populations, with the results of each model analyzed and compared.

Models and Algorithms

Basic Algorithm and Admixture Model

The STRUCUTRE algorithm uses the idea of MCMC (Markov Chain Monte Carlo) convergence in the form of Gibbs Sampling to sample and converge on local maximum likelihood points. STRUCTURE takes as input a collection of N samples which are each genotyped at L loci. It is assumed that the samples represent a mix of K populations, and one of the goals of the algorithm is to correctly assign the individuals to these populations.

In the most basic no-admixture model (not implemented in this project), each individual originates from one of the populations, which each has its own set of allele frequencies at each locus which is predicted by the algorithm. However, the obvious drawback to this most basic form is that in reality individuals are likely to have recent ancestors from more than one population. As an improvement, the admixture model was introduced, which assumes an individual receives some proportion of its ancestry from each population. In this model $q_k^{(i)}$ represents the proportion of sample i's genome that can be attributed to population k. The admixture model also makes it possible for an individual's different allele copies to come from different populations, and to account for this $z_l^{(i,a)}$ is introduced. $z_l^{(i,a)}$ represents the ancestral population for the ath allele copy (in the case of diploid individuals a would be either 1 or 2) at locus l from individual i. An additional variable p_{klj} representing the frequency of allele j at locus l in population k is needed in order for the sampling of z.

The algorithm starts off by first initializing all the z's through random sampling from a uniform prior. The next step is just to iterate through the Gibbs Sampling process until completion, which I have defined as a set number of iterations. Each iteration follows this series of steps:

1. Run through all of the z's and calculate n_{klj}, which represents the counts of allele j at locus l in population k.
2. Run through all of the z's and calculate $m_k^{(i)}$, which represents the counts of alleles in sample i from population k.
3. At each locus, p_{klj} is sampled from the Dirichlet distribution, with the parameters (for alleles) set to $\lambda + n_{klj}$.
   The Dirichlet distribution is a multivariate distribution where values of the variants sum to 1, and concentration parameters α_i for each variable i, where
   $$f(\mathbf{x}, \boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$
4. For each sample $q_k^{(i)}$ is sampled from the Dirichlet distribution, with the parameters (for populations) set to $\alpha + m_k^{(i)}$.
5. Each $z_l^{(i,a)}$ is sampled from the following distribution:
   $$P\left(z_l^{(i,a)} = k \middle| X, P, Q\right) = \frac{q_k^{(i)} P\left(x_l^{(i,a)} \middle| P, z_l^{(i,a)} = k\right)}{\sum_{k'} q_{k'}^{(i)} P\left(x_l^{(i,a)} \middle| P, z_l^{(i,a)} = k'\right)}$$
   Where $x_l^{(i,a)}$ is the ath allele at locus l in sample i, and $P(x_l^{(i,a)}|P, z_l^{(i,a)} = k) = p_{klx_l^{(i,a)}}$

Linked Loci Extension

Although the basic model for STRUCTURE works successfully, it doesn't accurately reflect real life since it assumes that the z's within each individual is independent. Each z is not independent, which is due to three sources of linkage disequilibrium. The first is variation in ancestry within the sampled individuals, which leads to correlations in markers across the genome even if they are unlinked since individuals with large proportions of ancestry from a population k will have many alleles that are in common with k. This type of linkage disequilibrium, also called "mixture LD", and is modeled by the admixture model.

The linked loci extension improves on the admixture model by also considering "admixture LD", which is the second source of linkage disequilibrium. In reality, admixture happens when chromosomes are broken into chunks and then swapped, so each individual allele and locus is not independent from the preceding and subsequent alleles at different loci. The linked loci model incorporates methods to deal with admixture LD, but still doesn't take into consideration the third type of linkage disequilibrium "background LD".

The linked loci model still follows the general process described above in the admixture model, but implements some changes in the updating of q, p, and z. In addition, the linkage model introduces a new variables r and β. We assume that the breakpoints between chunks happens at random as part of a Poisson process with a rate of r per unit of genetic distance, and the ancestry for each chunk is independently drawn according to q, which is still the expected ancestry proportions for each individual. In the formulas for the model r is multiplied by d, the genetic distance between loci, which is assumed to be known. However, since my data is not human data but instead genotypes from whitefish populations with no data on genetic distances between

loci, I set d to 1 which means that my algorithm assumed a constant rate of change. Meanwhile β_{lk} (or $\beta_{lk^1k^2}$ depending on the version of the linked loci model) is used in the forward-backward calculation of probability matrix that the z's are sampled from.

This model still starts off with the initialization of all the z's through random sampling from a uniform prior, and then iterates through the Gibbs Sampling process until completion. Each iteration consists of:

1. Run through all of the z's and calculate n_{klj}, which represents the counts of allele j at locus l in population k.
2. Run through all of the z's and calculate $m_k^{(i)}$, which represents the counts of alleles in sample i from population k.
3. At each locus, p_{klj} is sampled from the Dirichlet distribution, with the parameters (for alleles) set to $\lambda + n_{klj}$.
4. For each sample $q_k^{(i)}$ is sampled from the Dirichlet distribution, with the parameters (for populations) set to $\alpha + m_k^{(i)}$.
   However, this time q is updated using a Metropolis-Hastings step. This means that for each iteration the total likelihood of the q's is calculated and if the new likelihood is higher than the old the new values are accepted. Otherwise the new values are accepted with probability = (new likelihood)/(old likelihood).
5. For each sample the β's are calculated and then $z_l^{(i,a)}$ is sampled. This is where the linked loci model splits into 2 versions: one where it is assumed that individuals are either haploid or that the phase is known, and one for data with unphased or partially phased diploids. These two versions have different β calculations and probability distributions to sample $z_l^{(i,a)}$ from.
   - For the phased version:
     1. For each individual calculate β_{lk} which is
        $$\beta_{1k} = q_k p_{k1x_1}$$
        at the first locus and
        $$\beta_{(l+1)k'} = \sum_{k=1}^{K} \beta_{lk} \Pr\left(z_{l+1} = k' \mid z_l = k\right) p_{k'(l+1)x_{l+1}}$$
        for each successive locus, where
        $$\Pr\left(z_1^{(i)} = k \mid r, Q\right) = q_k^{(i)}$$
        and
        $$\Pr\left(z_{l+1}^{(i)} = k' \mid z_l^{(i)} = k, r, Q\right) = \begin{cases} \exp(-d_l r) + (1 - \exp(-d_l r))\, q_{k'}^{(i)} & \text{if } k' = k \\ (1 - \exp(-d_l r))\, q_k^{(i)} & \text{otherwise,} \end{cases}$$
        which I will call the z-equation (it will show up again).
     2. After each β is calculated, we can calculate the probability distribution for $z_l^{(i,a)}$, where for the very last locus L
        $$\Pr(z_L = k \mid X, P, r, Q) \propto \beta_{LK}$$
        and for each locus before it
        $$\Pr(z_l = k \mid z_{l+1}, \ldots, z_L, X, P, r, Q) \propto \beta_{lk} \Pr(z_{l+1} \mid z_l = k, r, Q)$$
        which also uses the z-equation. This gives the probability distribution to sample $z_l^{(i,a)}$ from.

- For the unphased version:
    1. For each individual calculate $\beta_{lk^1k^2}$ which is
    $$\beta_{1\ k^1k^2} = q_{k^1} q_{k^2} p_{k^1 1 x_1^1} p_{k^2 1 x_1^2}$$
    at the first locus and
    $$\beta_{(l+1)k^{1\prime}k^{2\prime}} = \sum_{k^1=1}^{K} \sum_{k^2=1}^{K} \beta_{lk^1k^2} p_{k^{1\prime}(l+1)x_{l+1}^1} p_{k^{2\prime}(l+1)x_{l+1}^2} \left\{ b_l P_{k^{1\prime}k^1} P_{k^{2\prime}k^2} + (1-b_l) P_{k^{1\prime}k^2} P_{k^{2\prime}k^1} \right\}$$
    for each successive locus, where $P_{k'k}$ represents the *z*-equation.
    2. After each β is calculated, we can calculate the probability distribution for $z_l^{(i,a)}$, where for the very last locus L
    $$\Pr\left(z_L^1 = k^1, z_L^2 = k^2 \mid X, P, r, Q\right) \propto \beta_{Lk^1k^2}$$
    and for each locus before it
    $$\Pr\left(z_L^1 = k^1, z_L^2 = k^2 \mid z_{l+1}^1, z_{l+1}^2, \ldots, z_L^1, z_L^2, X, P, r, Q\right) \propto \beta_{lk^1k^2} \left\{ b_l P_{z_{l+1}^1 k^1} P_{z_{l+1}^2 k^2} + (1-b_l) P_{z_{l+1}^1 k^2} P_{z_{l+1}^2 k^1} \right\}$$
    where $b_l = 0.5$ for unphased data and $P_{k'k}$ represents the *z*-equation. This gives the probability distribution to sample $z_l^{(i,a)}$ from.

I wrote the code for each of the three models (admixture, linked loci phased, and linked loci unphased), although the linked loci phased version doesn't work correctly with my data since it contains diploid unphased data.

Results

The performance of each model was tested using sequenced AFLP genotype data from whitefish, provided by Louis Bernatchez and mentioned by Falush et al 2007. The data comes from two distinct populations: dwarf sized and normal sized fish. This made it an optimal for the testing of the STRUCTURE algorithm, since I could just compare the population assignments when k=2 to the actual distinct population group each individual was from. However, determining population structure from this dataset has been challenging for many conventional analysis methods due to the uncertainty about the underlying genotypes from using AFLPs in addition to the presence of null alleles and the limitations of genotype calling in polyploids. When looking at the data itself, each locus is represented by 0, 1, 2 representing bi-allelic data, or a 9 representing missing info. The lack of specific positioning for each allele and large amounts of missing data in certain samples made the phased linked loci model pretty much useless, but I was still able to implement correct and consistently working admixture and unphased linked loci STRUCUTRE algorithms.

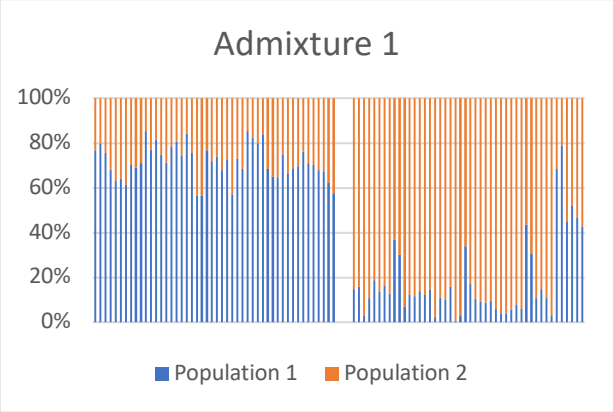
The output of my STRUCUTRE algorithm returns the *q* matrix, which is the matrix of admixture proportions in each population for each individual. Here is an example output for a few individuals (from the same population) run with a version of the algorithm that did not do a very good job in separating them into populations vs a model that did a much better job:

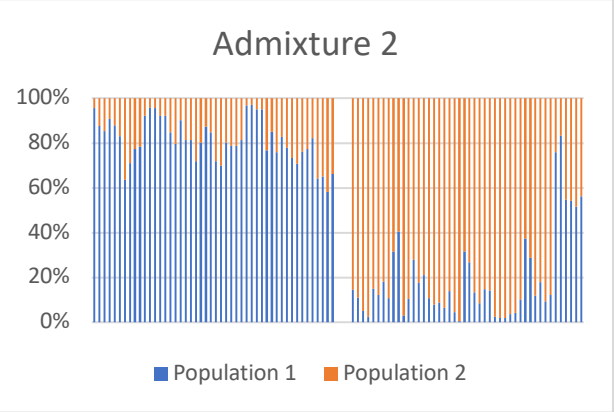
```
[ 0.6835475  0.3164525]
[ 0.78673809  0.21326191]
[ 0.44744202  0.55255798]
[ 0.52228686  0.47771314]
[ 0.46741067  0.53258933]
[ 0.42656434  0.57343566]
```

```
[ 0.10733103  0.89266897]
[ 0.00309699  0.99690301]
[  6.63917409e-04   9.99336083e-01]
[  2.85510323e-04   9.99714490e-01]
[ 0.00275558  0.99724442]
[  1.03696313e-04   9.99896304e-01]
```

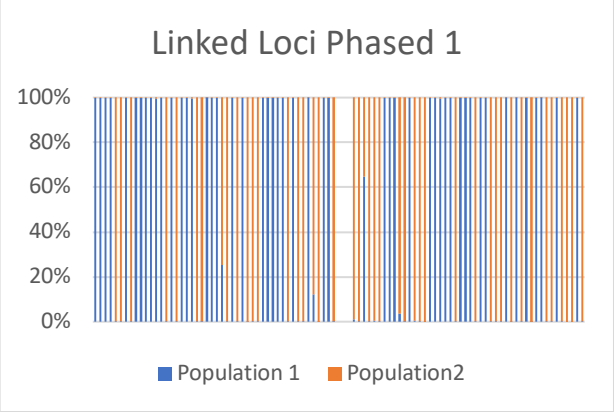
Outputs were then copied into excel, where graphs were made to visually show the admixture proportions of the individuals. Below are two examples from each of the models I implemented with the algorithm set to 80 iterations and k=2, where the two distinct populations are on each side of the white space in the middle of each graph:

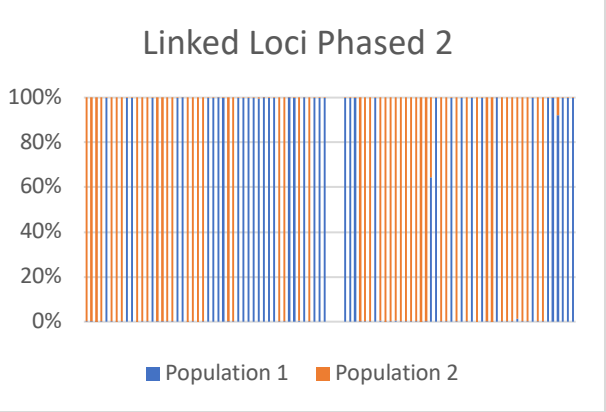
Admixture Model



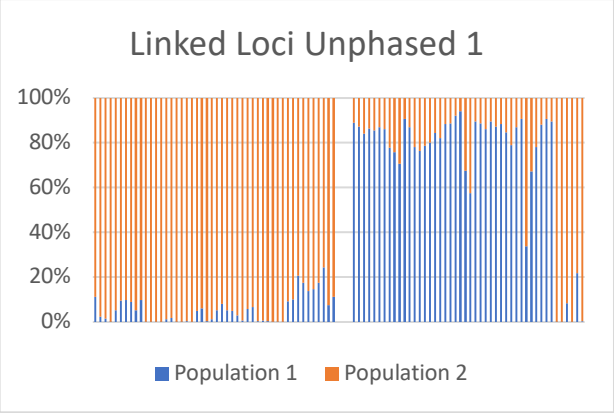


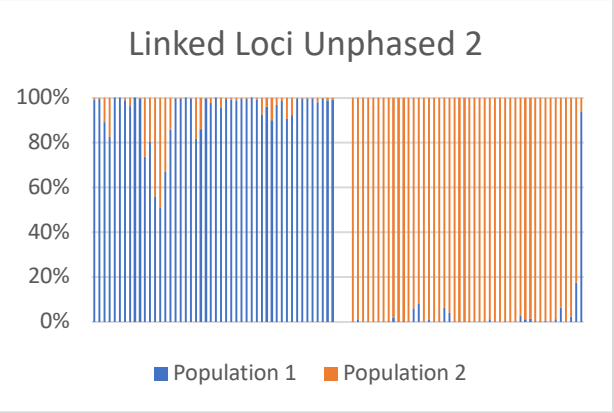
Linked Loci Phased Model





Linked Loci Unphased Model



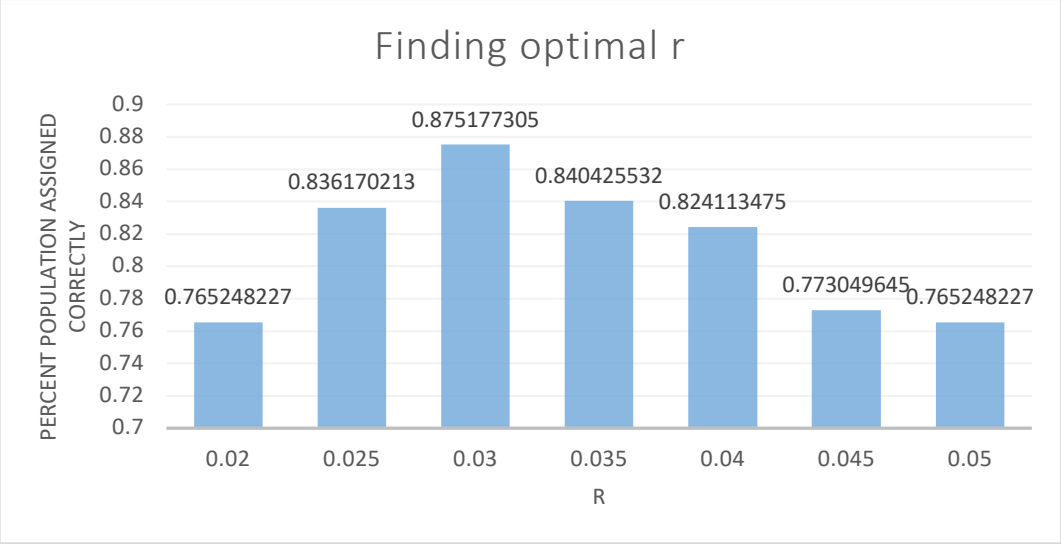


Each bar represents one individual and the different colors that make up the bar represent the percentage of the sample's genome attributed to the different populations. In this case, since k is set to 2, there are two colors representing the two ancestry populations. The samples are arranged by their actual population, with the first half left of the gap being individuals from dwarf sized whitefish and the second half right of the gap being individuals from normal sized whitefish. In total there were 48 dwarf sized whitefish samples and 46 normal sized whitefish samples.

The graphs above clearly show that the linked loci phased model did not work, while the admixture and linked loci unphased models did a very good job assigning individuals and their alleles to the correct populations, with the linked loci unphased model preforming better (as expected). It should be noted that the last few (6 out of 94) samples contained very large amounts of missing data, which would explain the errors the models consistently produced at the tail end of the graphs.

Although the linked loci unphased model did perform better than the basic admixture model, it is important to also consider the run time of the algorithm for scaling. On my computer (2015 MacBook Pro with 2.2 GHz Intel Core i7 processor) the admixture model was able to run through 80 iterations with the whitefish data in about 75 seconds while the linked loci model took significantly longer to run at around 8 minutes. The much more computationally intensive linked loci model will not scale as well when number of samples, number of loci, or k is increased. This is important to consider when choosing between the models for the analysis of large datasets.

In addition to testing the differences between the models, I also looked at various values for the rate of change *r*. During the development of the linked loci model I really struggled because even though I was implementing the algorithm correctly, neither the phased nor unphased model worked better than random assignments. Eventually I discovered that the problem was my *r* value, which I had set to 0.01 to represent the approximation of crossovers per megabase for humans. It wasn't until I decided to try different values for *r* that I started seeing results for the linked loci unphased model. To find the optimal value I ran the algorithm 20 times each for various values of *r* and graphed the results:



To make the graph I created a simple method that assigned each individual to the population it had the largest proportion of ancestry from. This assignment was then compared to the actual populations, and the percent of correctly assigned individuals was calculated. I averaged this value for 20 runs of STRUCUTRE linked loci unphased algorithm for each value of r between 0.02 and 0.05 at intervals 0.005. This is shown in the graph above, which led me to the conclusion that setting $r = 0.03$ was the optimal value for the whitefish data I was using.

Discussion

Looking at the results we can see that, when implemented correctly, the basic admixture model STRUCTURE algorithm is able to reliably distinguish between different populations and assign alleles to different ancestry populations with a high rate of success. In addition, the more complicated and advanced linked loci unphased model STRUCUTRE algorithm is even better with more distinct and correctly assigned admixture proportions.

Unfortunately, the run time of the models is an issue, and even though the linked loci model is improved, it takes significantly longer to run and thus when analyzing large datasets one must consider the trade-off between accuracy and speed. Another issue is the rate of change variable r, which will be different for different species. In some later studies of STRUCUTRE a sampling and then Metropolis-Hastings update for r is suggested to converge on the optimal value of r for the particular dataset run at the time, but I chose not to implement this due to worries about further increasing the run time of the algorithm, which is already a problem.

Nevertheless, the linked loci model that is able to take into consideration admixture linkage disequilibrium is an important addition to the model. Future models of STRUCTURE may even be able to consider background linkage disequilibrium to give even more accurate estimates, although the full coalescent approaches to this problem is computationally challenging. However, the possibility of further improving the STRUCUTRE model is very exciting, as it could potentially improve on the current results in the study of major events in history for species.

References

1. Pritchard J. K., Stephens M., Donnelly P., 2000 Inference of population structure using multilocus genotype data. Genetics 155: 945–959.
2. Falush D, Stephens M, Pritchard JK. 2003 Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. Genetics 164:1567–87.
3. Porras-Hurtado, Liliana et al. 2013 An overview of STRUCTURE: applications, parameter settings, and supporting software. Frontiers in genetics vol. 4 98. 29
4. Guo, Yuxin et al. 2018 Autosomal DIPs for population genetic structure and differentiation analyses of Chinese Xinjiang Kyrgyz ethnic group. Scientific reports vol. 8, 11054. 23
5. Raj, A., Stephens, M., Pritchard, J.K. 2014 fastSTRUCTURE: variational inference of population structure in large SNP data sets. Genetics 197: 573–589
6. Falush, D., Stephens, M., & Pritchard, J. K. (2007). Inference of population structure using multilocus genotype data: dominant markers and null alleles. *Molecular ecology notes*, *7*(4), 574-578.