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.

FINISH THIS PART

# **Models and Algorithms**

## **Basic Algorithm**

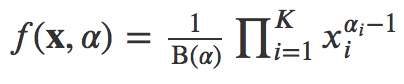
The STRUCUTRE algorithm uses the idea of MCMC (Markov Chain Monte Carlo) convergence in the form of Gibbs Sampling to sample and converge on local maximum likelihood points. STRUCTURE takes as input a collection of N samples which are each genotyped at L loci. It is assumed that the samples represent 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 *qk(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 *zl(i,a)* is introduced. *zl(i,a)* represents the ancestral population for the *a*th allele copy (in the case of diploid individuals *a* would be either 1 or 2) at locus *l* from individual *i*. An additional variable *pklj* representing the frequency of allele *j* at locus *l* in population *k* is needed in order for the sampling of *z*.

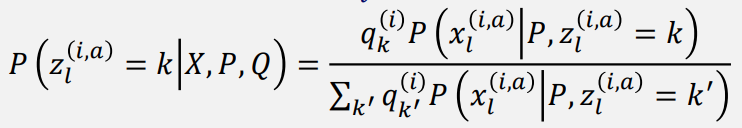
The algorithm starts off by first initializing all the *z*’s through random sampling from a uniform prior. The next step is just to iterate through the Gibbs Sampling process until completion, which I have defined as a set number of iterations. Each iteration follows this series of steps:

1. Run through all of the *z*’s and calculate *nklj*, which represents the counts of allele *j* at locus *l* in population *k*.
2. Run through all of the *z*’s and calculate *mk(i)*, which represents the counts of alleles in sample *i* from population *k*.
3. At each locus, *pklj* is sampled from the Dirichlet distribution, with the parameters (for alleles) set to λ + *nklj*.

The Dirichlet distribution is a multivariate distribution where values of the variants sum to 1, and concentration parameters αi for each variable i, where



1. For each sample *qk(i)* is sampled from the Dirichlet distribution, with the parameters (for populations) set to α + *mk(i)*.
2. Each *zl(i,a)* is sampled from the following distribution:



Where *xl(i,a)* is the *a*th allele at locus *l* in sample *i*, and P(*xl(i,a)*|P, *zl(i,a)* = *k*) = 

## **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 just set *d* to 1 which means that my algorithm assumed a constant rate of change. Meanwhile β*lk* (or β*lk1k2* 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:

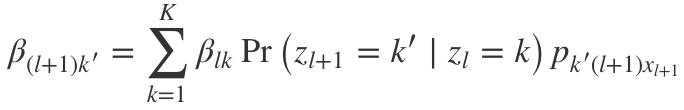
1. Run through all of the *z*’s and calculate *nklj*, which represents the counts of allele *j* at locus *l* in population *k*.
2. Run through all of the *z*’s and calculate *mk(i)*, which represents the counts of alleles in sample *i* from population *k*.
3. At each locus, *pklj* is sampled from the Dirichlet distribution, with the parameters (for alleles) set to λ + *nklj*.
4. For each sample *qk(i)* is sampled from the Dirichlet distribution, with the parameters (for populations) set to α + *mk(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).

1. For each sample the β’s are calculated and then *zl(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 *zl(i,a)* from.
   * For the phased version:
     1. For each individual calculate β*lk* which is



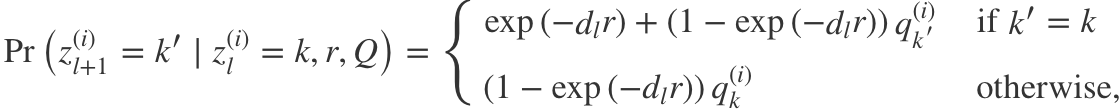
at the first locus and



for each successive locus, where



and



which I will call the *z*-equation (it will show up again).

* + 1. After each β is calculated, we can calculate the probability distribution for *zl(i,a)*, where for the very last locus L



and for each locus before it

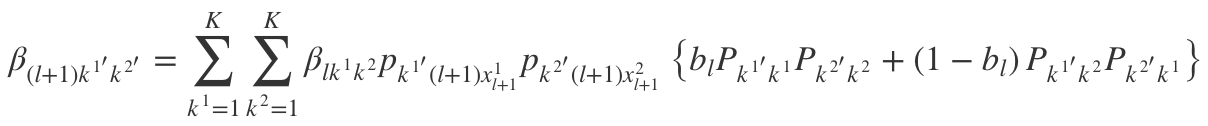


which also uses the *z*-equation. This gives the probability distribution to sample *zl(i,a)* from.

* + For the unphased version:
    1. For each individual calculate β*lk1k2* which is

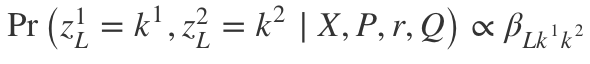


at the first locus and

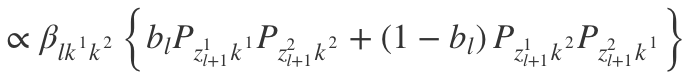
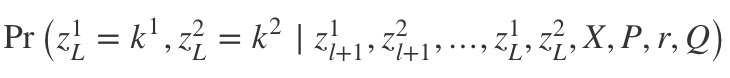


for each successive locus, where Pk’k represents the *z*-equation.

* + 1. After each β is calculated, we can calculate the probability distribution for *zl(i,a)*, where for the very last locus L



and for each locus before it



where *bl* = 0.5 for unphased data and Pk’k represents the *z*-equation. This gives the probability distribution to sample *zl(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

# **Discussion**

# **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