Some background on the program *Structure*

Lisa N. Barrow and Peter Beerli

The program *Structure* [Pritchard et al., 2000] uses multilocus genotype data to investigate population structure. The program is used frequently to infer the presence of distinct populations, assigning individuals to populations, studying hybrid zones, identifying migrants and admixed individuals, and estimate population allele frequencies in situations where many individuals are migrants or admixed. *Structure* uses Bayes theorem to calculate posterior distributions and uses Markov Chain Monte Carlo (MCMC).

I. Background

Often in population genetic studies, we record genetic differences among sampled individuals. Some questions we may want to address are: (1) Are populations really different from each other?; and (2) If we have an individual's genotype, can we figure out which population it came from? If, for example, we a have a set of individuals sampled for a number of loci (Table I.

Table I
Example of a genoptype data set.

Individual	Locus1	Locus2
1	AA	...
2	AA	...
3	BB	...
4	CB	...
5	CB	...

We could sort the different genotypes into two groups, e.g. Group 1: AA, AA and Group 2: BB, BB, and CB, but we need some sort of justification for these groupings. Clustering methods like *Structure*, *Structurama* [Huelsenbeck and Andolfatto, 2007], and many others use an optimality criterium to determine how different sampled individuals are.

Clustering methods fall into two main categories: distance-based and model-based methods. Distance-based methods, such as K-means clustering, work by calculating distances between individuals and assigning them to clusters in a way that minimizes the distances between an individual coordinate and a center point for that cluster. *Structure* is in a way very similar to k-means clustering but it does not use distances among individuals. It is a model-based method that takes a probabilistic approach to assign individuals to populations under certain assumptions, for example the populations (clusters) need to be in Hardy-Weinberg equilibrium (HWE), or there is no admixture (each individuals is purely only from one population). In contrast to distance methods such assumptions can be relaxed and *Structure* allows for deviation of the HWE and also for admixed individuals (an individuals could be a mixture of different populations, for example locus 1 has an ancestor in population A, whereas locus 5 has an ancestor in population B.

II. The Model

The basic model assumes Hardy-Weinberg equilibrium within populations and linkage equilibrium between loci within populations. Here we also assume that each locus is independent. For each individual, we want to know its population of origin. We also want to know the allele frequencies in each population.

The vector, Z_i, represents the population of origin of each individual i, and P_j is a vector of the allele frequencies of the population j. Both vectors Z and P are **unknown** and we want to estimate them. We have the data X, which is a long list of genotypes for every individual for every locus. IN our framework we can now (in principle) calculate how probable is it to see a particular data set X if we know Z and P, we want to calculate the likelihood of Z and P or, equivalently, the probability of the data X given **known** Z and **known** P.

$$Pr(X|Z,P)$$

In the best of all worlds we could experiment with different Z and different P and find the best combination that maximizes the probability, unfortunately this is commonly impossible because the problem (finding good Z and P) is too complicated. In order to infer the parameters of interest, Z and P, *Structure* uses a Bayesian approach. This relies on Bayes' theorem:

$$Pr(A|B) = \frac{Pr(A) * Pr(B|A)}{Pr(B)}$$

where the left side is "the posterior probability density of the model A given the data B", $Pr(A)$ is the model prior distribution, $Pr(B|A)$ is the likelihood, and $Pr(B)$ is the probability of the data, which can be thought of as a scalar so that the right hand side is 1.0 if we integrate the posterior distribution over all parameter values. It is common to leave off the denominator and show

$$Pr(A|B) \propto Pr(A) * Pr(B|A).$$

If we now replace A and B with our quantities of interest, the observed genotypes X, the unknown Z and P, we get:

$$Pr(Z,P|X) \propto Pr(Z) \times Pr(P) \times Pr(X|Z,P)$$

This is the most important equation to understand Structure, but it is hiding a lot. It also assumes that we know how to calculate the prior distributions of the parameters and the likelihood. Computing this probability distribution becomes a high-dimensional mess, so *Structure* uses a method called Markov Chain Monte Carlo (MCMC) to approximate the sample, like many of the other programs we will discuss.

III. Implementation of *Structure* and MCMC

In the 1950s Markov Chain Monte Carlo was developed [for a short history, see Robert and Casella, 2011], a sampling method that allows us to obtain an approximate sample from a probability distribution, rather than trying to calculate a tricky integral explicitly. A Markov chain has the property that it is "memoryless", that is, the next state it samples depends only on the current state, and has no memory of where the chain has been previously. MCMC is executed in series of steps, usually we need to run thousands or millions of these steps. Basically, the steps are

1) Start by setting arbitrary values for the quantities Q to estimate.
2) Calculate the priors probabilities and the likelihood
3) Propose a change to the quantity Q' of interest, for example change an allele frequency in P.
4) Recalculate the likelihood and prior and compare with the old value.
5) if we accept $r < \frac{\text{new likelihood*prior}}{\text{old likelihood*prior}}$, where r is a uniform random number then we record the new quantities Q' otherwise we record the old quantity Q and reset to the old quantity.
6) Go to step 3 and repeat as many times as is needed.

After the run we can create histograms of the collected quantities and these histograms are equivalent to the posterior distribution.

In Structure, the algorithm starts with initial value $Z^{(0)}$ for Z, then iterates the following steps:

- Step 1: Sample $P^{(m)}$ from Pr($P|X$,$Z^{(m-1)}$) (1)
- Step 2: Sample $Z^{(m)}$ from Pr($Z|X$,$P^{(m)}$) (2)

In Step 1, the allele frequencies are estimated for each population assuming the population of origin for each individual is known. In Step 2, the population of origin for each individual is estimated assuming the allele frequencies for population are known.

In general MCMC works by choosing some arbitrary start, filling out all the values based on some distribution, then go to the next step. The Metropolis algorithm is used as a validation step to see how good the answer is. It is also important to consider: How is the prior distribution determined? We might have the alleles A = 0.4 and B = 0.6. We need a mechanism that delivers numbers that are consistent with each other.

Structure uses the Dirichlet distribution to model the allele frequencies for each locus in a population. This distribution generates prior information for data that is fractional, that is, allele frequencies sum to 1. It has a set of parameters such that if you draw a sample from the Dirichlet distribution, the sum of all samples will be 1. This becomes a more complicated problem if you have three alleles. Two will co-vary to have all of them add up to 1.

The distribution specifies the probability of the allele frequencies p_{kl} for population k at locus l:

$$p_{kl} \sim D(\lambda_1, \lambda_2, \lambda_3, ...\lambda_{J_l})$$

The vector p_{kl} contains a random draw of allele frequencies for population k at locus l. We use the observations to determine how many times we see a given allele combination, and modify the Dirichlet distribution based on these observations, generating a population allele frequency that is biased towards the observed data by changing the λ_i. The higher λ_i is the more confidence we have in that particular allele frequency. This results in a calculation of the likelihood

$$Pr(X|Z,P) = p^*_{kl}$$

where p^*_{kl} is drawn from the data informed Dirichlet

$$p^*_{kl} \sim D(\lambda_1 + \#1_{kl}, \lambda_2 + \#2_{kl}, ...)$$

the $\#_{kl}$ mark the allele frequency calculated from the data X using the population assignment for each individual in Z.

To recapitulate: we start with guessed values for Z and P, then evaluate using MCMC many combinations of Z and P using the procedure sketched out in formula (1) and (2), that leads to a long lost of recorded pairs of Z and P (Table II).

Table II
Record of an MCMC run; m is the iteration or indicator, P is the allele frequency, and Z is the population of origin.

m	p	z
0	p^0	z^0
1	p^1	z^1
2	p^2	z^2
...	...	...
	$E(p)$	$E(z)$

This usually gets run hundreds of thousands of times. The bad estimates at the beginning get thrown out as **burn-in**. MCMC is driven by the data and the prior: Z are informed by the P, and P are driven by the data. Eventually, they arrive at some equilibrium. The average of these recorded values could be a reported as the mean allele frequency for each population and the mean location for each individual. We can also take the values and form histograms (the posterior distribution) that tells us about uncertainty of the estimates.

this section talks about the number of populations and will be filled in on Wednesday

Bibliography

J Huelsenbeck and P Andolfatto. Inference of population structure under a dirichlet process model. *Genetics*, 175: 1787–1802, 2007.

J Pritchard, Matthew Stephens, and P Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

Christian Robert and George Casella. A short history of markov chain monte carlo: Subjective recollections from incomplete data. *Statistical Science*, 26(1): 102–115, 02 2011. doi: 10.1214/10-STS351. URL `http://dx.doi.org/10.1214/10-STS351`.