An Investigation of Population Subdivision Methods in Disease Associations with a Focus on Markov Chain Monte Carlo

Lian Garton
Department of Computer Science
Brown University
lian@cs.brown.edu

May 2, 2008

Acknowledgements

I would like to thank first and foremost Professor Sorin Istrail for advising and motivating me throughout the process of working on this project, having suggested this topic in the first place, and for pushing me to present my work as an honors thesis. I would also like to thank Professor Dan Weinreich for acting as my second reader. Aurojit Panda, Itay Neeman, and Jason Davis also deserve a lot of credit for all of their help, especially during the late nights.

Contents

Abstract

Methods of population stratification are important for disease association studies, in which unknown population subdivisions can cause major complications. We present a comparison between existing methods of determining population subdivisions, most notably Markov Chain Monte Carlo and PCA-correlated SNPs. We also extend the basic Markov Chain Monte Carlo algorithm for small numbers of individuals by implementing and applying the Propp-Wilson algorithm to ensure convergence.

1 Introduction

The major complicating factor for disease association studies is the existence of unknown subdivisions in the populations under study. As such, an important area of ongoing research is the identification and inference of population structure. Our work centers on investigating and comparing existing methods and measures of population stratification and applying a more rigorous theoretical approach to real data.

Much of the focus of our research is on the application of Markov chain Monte Carlo (MCMC) methodology to the problem of population structure inference. MCMC involves sampling from a probability distribution to make predictions or inferences about model parameters. Samples are gathered from a constructed Markov chain that is run long enough to converge to the necessary stationary distribution. As applied to the population subdivision problem, the distribution of individuals or individual loci amongst population groups given the frequencies of alleles within each population group is of particular interest. Given a Markov chain constructed with this distribution, estimates of the model parameters, such as which individuals belong to which populations, can be obtained.

Some notable work by Jonathan Pritchard, Daniel Falush, et al. in this area has already resulted in the development of the program STRUCTURE, which infers population structure using multilocus genotype data using an MCMC algorithm. Their ongoing research has focused primarily on extending their software program to relax initial limitations, such as permitting linked loci and null alleles. Our research aims to focus on potential improvements to the application of the MCMC algorithm to the population subdivision problem by implementing some of the more theoretically rigorous approaches to MCMC, namely the Propp-Wilson algorithm, which ensures convergence to the stationary distribution.

Another important component of my work is a discussion of the existing methods and measures of subdivisions in population. Predating the work by Jonathan Pritchard, Daniel Falush, et al., a series of measures of population stratification were proposed based on F-statistics. For example, Wrights F_{ST} measure represents the correlation between genes in a particular subdivision of the total population. A much more recent development was the suggestion by Paschou et al. to use Principal Components Analysis (PCA), and identify PCA-correlated SNPs to capture population structure. By examining these other methodologies, I hope to contextualize my work on MCMC methods within the broader scope of proposed approximations to the problem of population structure identification.

2 Background

2.1 Markov Chain Monte Carlo (MCMC)

Monte Carlo integration involves evaluating the expected value of a function, say f by sampling from a set of random variables, say X, a total of n times according to the probability distribution of the random variables, say π, and then approximating the expected value by summing up the function applied to each of the n selected random variables and dividing by n. Markov Chain Monte Carlo utilizes a Markov chain to sample from X according to the distribution π.

2.1.1 Markov Chains

A Markov chain [5] is a stochastic process with the Markov property, meaning that future states depend only on the present state, not past states. This random process can be represented as a sequence of random variables $\{X_0, X_1, X_2, ...\}$, where each variable takes its value from a finite state space $S = \{s_1, s_2, ..., s_k\}$. A transition matrix P defines the probability of transitioning from a state X_t to X_{t+1}, given that $X_t = s_i$. In addition, an initial distribution, say $\mu^{(0)}$, defines the probability of X_0 taking its value from each of the possible states S.

When simulating a Markov chain on a computer, it is useful to define an initiation function and an update function, both of which take random numbers in the range $[0, 1]$. The initiation function maps intervals along the range $[0, 1]$ to states $\{s_1, s_2, ..., s_k\}$ according to the initial distribution, in order to return a random initial state. The update function performs a similar mapping, while taking into account the current state $s \in S$ to determine the probabilities of each candidate state, and returns the next randomly chosen state.

Markov chains can be either reducible or irreducible. An irreducible Markov chain has the property that every state can be reached by every other state. This means that there is no state s_i from which there is no chance of ever reaching a state s_j, even given a large amount of time and many transitions in between. Mathematically, this translates to each state s_i having a strictly positive probability of transitioning to every other state s_j. A reducible Markov chain is simply a Markov chain that is not irreducible.

Markov chains can also either be periodic or aperiodic. The period of a state s_i is defined as the greatest common divisor (gcd) of the set of times the chain has a positive probability of returning to s_i, given that $X_0 = s_i$ (i.e. we start with state s_i). If the period is one, the Markov chain is said to be aperiodic, otherwise it is considered periodic. For example, a Markov chain with two states s_1 and s_2, with s_1 transitioning to s_2 with probability 1 and s_2 transitioning to s_1 with probability 0.5, would be periodic. Starting with $X_0 = s_1$, the chain has a positive probability of returning to s_1 at times 2, 4, 6, 8, ..., therefore the chain is periodic with period 2.

Given a Markov chain that is both irreducible and aperiodic, some powerful properties hold. First, let us define a stationary distribution, π, A stationary distribution is a probability distribution on the set of states S such that if the initial distribution of states $\mu^{(0)}$ is the stationary distribution, then the distribution of states for every time following that will also be the stationary distribution. According to a set of proven theorems, an irreducible and aperiodic Markov chain has one and only one stationary distribution π, towards which the distribution of states converges as time approaches infinity, regardless of the initial distribution.

An important consideration is whether the Markov chain is reversible. A Markov chain with stationary distribution π and transition matrix P is said to be reversible if the stationary distribution is reversible, meaning $\pi_i P_{i,j} = \pi_j P_{j,i}$. If a distribution for a given Markov chain is reversible, then that distribution is the stationary distribution. This is an important theorem, since if it can be shown that a particular distribution of interest is reversible for a Markov chain, then the Markov chain will converge to that distribution.

2.1.2 Markov Chain Monte Carlo Implementations

Various implementations of Markov Chain Monte Carlo [4] exist to ensure that the distribution of interest is indeed the stationary distribution of the Markov chain by defining the way in which state updates are carried out. The general algorithm is known as Metropolis-Hastings, of which the Metropolis algorithm, single-component Metropolis-Hastings, and Gibbs sampling are special cases.

The Metropolis-Hastings algorithm depends on an acceptance-rejection criterion to ensure that the Markov chain converges to the distribution of interest π. According to the algorithm, during a given step t, a new state X_{t+1} is sampled from some proposal distribution Q, which can depend on the current state X_t. Thus, a new candidate state Y is chosen from $Q(X_{t+1}|X_t)$. The critical step is then determining whether to accept the candidate state and thus let $X_{t+1} = Y$, or reject the candidate state and let $X_{t+1} = X_t$. The candidate state is accepted with probability:

$$\alpha(X_t, Y) = \min\left(1, \frac{\pi(Y)Q(X_t|Y)}{\pi(X_t)Q(Y|X_t)}\right) \tag{1}$$

It can be shown that this algorithm results in a reversible Markov chain with distribution π. We first note that the probability of transitioning to state X_{t+1} given the current state X_t is equivalent to: (1) the probability of choosing X_{t+1} from the proposal distribution Q times the probability of accepting X_{t+1} ($\alpha(X_t, X_{t+1})$) plus, (2) in the case where $X_{t+1} = X_t$, the probability of rejecting all other candidates Y. The probability of rejecting all other candidates Y is given by 1 minus the probability of accepting each candidate Y according to the acceptance criterion. Using an indicator variable $I_{X_{t+1}=X_t}$ that is 1 when $X_{t+1} = X_t$ and 0 otherwise, we can write this mathematically as:

$$\begin{aligned} P(X_{t+1}|X_t) &= Q(X_{t+1}|X_t)\alpha(X_t, X_{t+1}) \\ &\quad + I_{X_{t+1}=X_t}[1 - \int Q(Y|X_t)\alpha(X_t|Y)dY] \end{aligned} \tag{2}$$

We now consider the acceptance criterion α, which we can rearrange as follows:

$$\alpha(X_t, X_{t+1}) = \min\left(1, \frac{\pi(X_{t+1})Q(X_t|X_{t+1})}{\pi(X_t)Q(X_{t+1}|X_t)}\right) \tag{3}$$

$$\begin{aligned}\frac{\pi(X_t)Q(X_{t+1}|X_t)\alpha(X_t,X_{t+1})}{\pi(X_{t+1})Q(X_t|X_{t+1})} &= \min\left(\frac{\pi(X_t)Q(X_{t+1}|X_t)}{\pi(X_{t+1})Q(X_t|X_{t+1})},1\right) && (4)\\ \frac{\pi(X_t)Q(X_{t+1}|X_t)\alpha(X_t,X_{t+1})}{\pi(X_{t+1})Q(X_t|X_{t+1})} &= \alpha(X_{t+1},X_t) && (5)\\ \pi(X_t)Q(X_{t+1}|X_t)\alpha(X_t,X_{t+1}) &= \pi(X_{t+1})Q(X_t|X_{t+1})\alpha(X_{t+1},X_t) && (6)\end{aligned}$$

Rewriting equation (2) we see that:

$$\begin{aligned}Q(X_{t+1}|X_t)\alpha(X_t,X_{t+1}) &= P(X_{t+1}|X_t)\\ &\quad -I_{X_{t+1}=X_t}[1-\int Q(Y|X_t)\alpha(X_t|Y)dY] && (7)\\ Q(X_t|X_{t+1})\alpha(X_{t+1},X_t) &= P(X_t|X_{t+1})\\ &\quad -I_{X_t=X_{t+1}}[1-\int Q(Y|X_{t+1})\alpha(X_{t+1}|Y)dY] && (8)\end{aligned}$$

Substituting these equations (7 and 8) into (6) and canceling out the equivalent indicator variable terms that appear on both sides, we finally see that

$$\pi(X_t)P(X_{t+1}|X_t) = \pi(X_{t+1})P(X_t|X_{t+1}) \tag{9}$$

Since this is the equation that needs to hold for the Markov chain to be reversible with distribution π, we can be sure that π is the stationary distribution of the Markov chain created by the Metropolis-Hasting algorithm. Thus, as long as the Markov chain is irreducible and aperiodic, the Metropolis-Hastings algorithm will result in samples that converge to the distribution of interest π.

Gibbs sampling is a special case of Metropolis-Hastings. However, the proposal distribution Q is taken to be the full conditional distribution for the stationary distribution π, so candidates are always accepted. Johannes and Polson [6] present a proof showing that single component Gibbs sampling results in a reversible Markov chain with stationary distribution π.

2.1.3 Propp-Wilson

The Propp-Wilson algorithm [5], or coupling from the past, involves running several copies of a Markov chain from some time in the past up to time 0 in order to guarantee convergence to the stationary distribution.

The algorithm works as follows:

1. Choose an increasing series of positive integers $(N_1, N_2, ...)$. A good candidate is $(N_1, N_2, ...) = (1, 2, 4, 8, ...)$ for reasons described shortly.

2. Start a counter $c = 1$.

3. Run a Markov chain starting at every state in the state space $S = \{s_1, s_2, ..., s_k\}$ for time $-N_c$ up to time 0, using an update function given the random numbers $U_{-N_m+1}, U_{-N_m+2}, ..., U_0$, which are the same for each of the k chains.

4. If all of the chains end up in the same state at time 0, then the Markov chain has converged to the stationary distribution.

5. Otherwise, increase the counter by 1 and go back to step 3, reusing the same random numbers $U_{-N_{m-1}+1}, U_{-N_{m-1}+2}, ..., U_0$ for time $-N_{m-1}$ to time 0.

Since $(N_1, N_2, ...)$ can be any increasing series of positive integers, one might wonder why the series $(1, 2, 4, 8, ...)$ was chosen. To understand why this is a good candidate, define an integer random variable N' as the minimum n such that the chains starting at time $-n$ converge to the same state at time 0. We now consider the alternative series $(1, 2, 3, 4, ...)$. Using this series, the number of time units the Markov chain needs to be run is

$$1 + 2 + 3 + 4 + \cdots + N' = \sum_{i=1}^{N'} i = \frac{N'(N' + 1)}{2} \tag{10}$$

which grows according to N'^2. In contrast, the series $(1, 2, 4, 8, ...)$, requires that the Markov chain needs to be run for the number of time units given by

$$1 + 2 + 4 + 8 + \cdots + N' = \sum_{i=0}^{\log_2 N'} 2^i = 2^{\log_2(N')+1} - 1 = 2 \cdot N' - 1 \tag{11}$$

which grows only linearly with N'. Thus, the series $(1, 2, 4, 8, ...)$ than $(1, 2, 3, 4, ...)$ is more efficient since it grows linearly as opposed to polynomially with N'.

Given that the number of states k can be very large, requiring the simulation of a possibly unrealistic number of Markov chains, an important idea in implementing Propp-Wilson is sandwiching. Sandwiching applies to Markov chains that have some sort of ordering and allows a fewer number of initial states to be considered. Namely, if a set of states can be ordered in such a way that the convergence of the first and the last state guarantees convergence of the inner states, then only the first and the last state need to be simulated from time $-N_m$ to time 0 until they converge.

2.2 Population Subdivision Problem

The basic population subdivision problem requires separating M individuals into K populations, based on the genotypes, given by genetic markers, of the individuals.

2.2.1 Pritchard's Markov Chain Monte Carlo Approach

Jonathan Pritchard et al. developed a Markov Chain Monte Carlo algorithm for inferring population structure using genotype data [8]. They assumed that each population "is modeled by a characteristic set of allele frequencies."

In the initial algorithm, they specify three vectors, X, P, and Z, where X is the given observed genotypes, P is the unknown allele frequencies for each population, and Z is the unknown population of origin of the individuals. It is assumed that the populations are in Hardy-Weinberg equilibrium and that loci within populations are in complete linkage equilibrium. They define the following elements in each of the vectors:

$$\begin{aligned}
(x_l^{(i,1)}, x_l^{(i,2)}) &= \text{genotype of individual } i \text{ at locus } l \text{, where} \\
& \quad\; i = 1, 2, ..., M \text{ and } l = 1, 2, ..., L \\
z^{(i)} &= \text{population of origin of individual } i \\
p_{klj} &= \text{frequency of allele } j \text{ at locus } l \text{ in population } k \\
& \quad\; \text{where } k = 1, 2, ..., K \text{ and } j = 1, 2, ..., J_l \text{ and} \\
& \quad\; J_l \text{ is the number of distinct alleles observed at locus } l
\end{aligned}$$

They propose using Markov Chain Monte Carlo methods to sample from the distribution $\Pr(Z, P|X)$ and infer Z and P from $(Z^{(0)}, P^{(0)})$, $(Z^{(1)}, P^{(1)})$, ..., $(Z^{(N)}, P^{(N)})$. They use a Gibbs sampler with stationary distribution $\pi(Z, P) = \Pr(Z, P|X)$ and an initial uniform distribution for Z. The algorithm is a two-step process:

1. Sample $P^{(n)}$ from $\Pr(P|X, Z^{(n-1)})$.

2. Sample $Z^{(n)}$ from $\Pr(Z|X, P^{(n)})$.

Step 1 involves sampling new allele frequencies for each population given X and the previous Z. The allele frequencies for each locus l and each population k are modeled by the Dirichlet distribution, namely $p_{kl} \sim \mathcal{D}(\lambda_1, \lambda_2, ...\lambda_j)$. Specifically,

$$p_{kl}|X, Z \sim \mathcal{D}(\lambda_1 + n_{kl1}, \lambda_2 + n_{kl2}, ...\lambda_j + n_{klJ_l}) \tag{12}$$

where $\lambda_1 = \lambda_2 = \lambda_j = 1.0$ to give a uniform distribution of allele frequencies and n_{klj} is the number of copies of allele j at locus l observed in individuals belonging to population k according to Z.

Step 2 involves sampling new populations for each individual given X and the new P. The genotypes are assumed to be random independent drawing of alleles from the population frequency distribution of alleles, meaning $\Pr(x_l^{(i,a)} = j|Z, P) = p_{z^{(i)}lj}$. Reversing this idea, we see that the $\Pr(Z|X, P)$ can be calculated as

$$\Pr(z^{(i)} = k|X, P) = \frac{\Pr(x^{(i)}|P, z^{(i)} = k)}{\sum_{k=1}^{K} \Pr(x^{(i)}|P, z^{(i)} = k)} \tag{13}$$

where $\Pr(x^{(i)}|P, z^{(i)} = k) = \prod_{l=1}^{L} p_{klx^{(i,1)}} p_{klx^{(i,2)}}$. This assumes that an equal proportion of the sample comes from each population.

Jonathan Pritchard et al. also extended the original algorithmic framework to handle admixed populations by adding a vector Q that records the proportion of an individual's genome that originates from a given population. Further extensions allowed accounting for linked loci, a new prior model for allele frequencies based partly on Wright's F_{ST} measure, [2] and the inclusion of dominant markers and null alleles [3].

2.2.2 Paschou's Principal Component Analysis (PCA) Approach

Paschou et al. propose using Principal Component Analysis (PCA) to identify SNPs capturing population structure, which can then be used to perform k-means or other clustering and subdivide the individuals into populations [7].

Their algorithm operates on an $m \times n$ matrix A with m individuals and n SNPs per individual. The elements of the matrix A are $\{-1, 0, +1\}$, representing, respectively, homozygous for allele 1, heterozygous, and homozygous for allele 2. The algorithm operates under the supposition that there is a larger number of SNPs than individuals ($m \leq n$).

To perform PCA, the Singular Value Decomposition (SVD) of the matrix A is calculated. One of the three matrices returned by the SVD is a matrix U of orthonormal vectors that are linear combinations of the columns of SNPs. These vectors are termed "eigenSNPs." The common technique known as dimensionality reduction is to estimate the matrix A by using only the first k columns of the matrix U. Thus we can consider only the fist k eigenSNPs. To find actual SNPs that correspond to these eigenSNP abstractions, we can sort the columns of A according to scores given by the sum of the squared coefficients of the top k eigenSNPs. The top k columns of this newly sorted matrix should then represent the SNPs that capture the most structure of the data.

Paschou et al. also suggest a method for determining the number of principal components k of the matrix A. To do this, they consider whether the ith principal component results in a matrix that has significantly (which they say is 15%) more structure compared to a random permutation of the matrix. To evaluate the significance of the ith principal component, the top singular value of a matrix A_{m-k} reconstructed from the ith and smaller principal components is compared to the top singular value of the same matrix A_{m-k} with the rows randomly permuted. If the top singular value of A_{m-k} is more than 15% of the randomly permuted version, then the ith principal component is considered significant.

2.2.3 Wright's F_{ST} Measure

Wright's F_{ST} is a measure of population subdivision, not a method for determining the subdivisions among a set of individuals [1]. It represents "the correlation between genes within a deme or subdivision (S) relative to the genes of the total population." It is calculated as follows:

$$F_{ST} = \frac{\mathrm{var}_S(p)}{\overline{p}(1-\overline{p})} \tag{14}$$

Thus, F_{ST} is a measure of the variance of allele frequencies across subdivisions divided by the maximum variance, which is the variance that will result from the allele frequencies within each subdivision diverging completely from the average population frequency.

3 Discussion of Methods and Results

To further the current work surrounding the Population Subdivision problem, we extended Pritchard's MCMC algorithm and performed a cross validation of the MCMC and PCA approaches. We implemented all of the algorithms in Mathematica version 6.0, which provided useful built-in functionality such as the ability to sample from a specified distribution and compute the singular value decomposition of a matrix.

3.1 Propp-Wilson Extension to Pritchard's MCMC Algorithm

The benefit of Propp-Wilson is that it guarantees convergence to the stationary distribution, which in the case of Pritchard's MCMC algorithm is the $\Pr(Z, P|X)$. In order to guarantee this convergence, Propp-Wilson requires a Markov chain to be run from all possible states $S = \{s_1, ..., s_n\}$. In the context of the MCMC algorithm, we need to consider the possible states for Z and P. However, since P depends on Z, and we are generally primarily focused on Z, we can focus on the possible states for Z. The set of possible states for Z is the set of groupings of individuals into k populations.

The algorithm works well for very small numbers of individuals (less than 10), but unfortunately grows exponentially with the number of individuals. For $k = 2$ populations and n individuals, the number of possible states for Z is 2^{n-1}, and in general is $O(n^k)$.

3.2 Cross Validation of MCMC and PCA Methods

We begin with the idea that PCA-correlated SNPs should capture population structure better than randomly selected SNPs. To confirm this idea, we propose running the MCMC algorithm on sets of individuals from different populations using both PCA-correlated SNPs and randomly selected SNPs.

We used data from the Human Genome Diversity Project (HGDP)-Centre dEtude du Polymorphisme Humain (CEPH) Cell Line Panel, which consisted of 2834 SNPs taken across 36 genomic regions in 927 individuals from 7 continental regions. To perform the cross validation, we chose individuals randomly from a set number of populations, $k = \{2, 3, 4, 5, 6, 7\}$. (Note that separating individuals into one population is not informative.) For each of the 36 genomic regions, we identified the top PCA-correlated SNP, and then chose one SNP randomly from each region. We ran the MCMC algorithm with a set burn-in time of 10000 iterations on both the individuals' PCA-correlateds SNPs and the individuals' randomly selected SNPs. We compared the resulting population subdivision to what we know from the original data set is the true continental subdivisions of the individuals.

As evidenced by the results shown in Figure 1, the MCMC algorithm was much better at subdividing individuals into populations using the PCA-correlated SNPs instead of randomly selected SNPs from the 36 genomic regions.

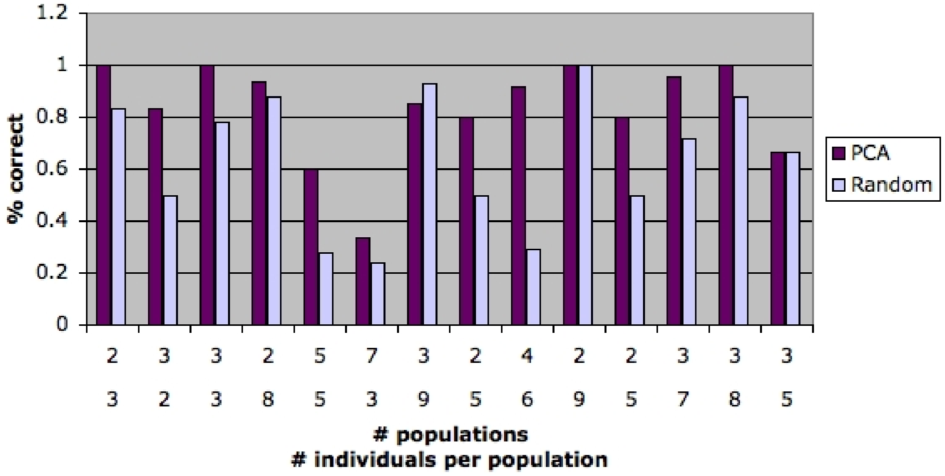

Figure 1: Comparing the ability of the MCMC algorithm to subdivide individuals into populations using PCA-correlated SNPs and randomly selected SNPs to the ground truth using the HGDP-CEPH data.

4 Conclusions

We have seen that Propp-Wilson is impractical for more than a small number of individuals, although the convergence guarantees make this approach enticing. With regards to the cross-validation, we showed that PCA-correlated SNPs distinguish individuals better than random SNPs, providing evidence for the idea that PCA-correlated SNPs capture population structure and that the MCMC algorithm can correctly identify populations of individuals when given population-dependent genotype data.

5 Future Work

To continue the work presented in this paper, obtaining results from other data sets (e.g. HapMap) would provide more evidence that could prove useful in the analysis of the existing population subdivision methods. Also, we would like to investigate whether the limited results of the Propp-Wilson algorithm could be useful in perhaps providing better estimates of convergence time, or if there is some way to limit the exponential state space to produce meaningful results for larger data sets in a reasonable amount of time. We would also like to consider better controls for cross-validation, as the computational complexity of the problem limits the ability to perform the MCMC algorithm over large numbers of SNPs and individuals.

References

[1] Excoffier, L. ”Analysis of Population Subdivision.” Handbook of Statistical Genetics. Ed. DJ. Balding et al. John Wiley & Sons, Ltd, 2001. 271-307.

[2] Falush D, Stephens M, and Pritchard JK. ”Inference of Population Structure Using Multilocus Genotype Data: Linked Loci and Correlated Allele Frequencies.” Genetics 164 (August 2003): 1567-1587.

[3] Falush D, Stephens M, and Pritchard JK. ”Inference of Population Structure Using Multilocus Genotype Data: Dominant Markers and Null Alleles.” Blackwell Publishing Ltd, 2007.

[4] Gilks WR, Richardson S, and Spiegelhalter DJ, Ed. Markov Chain Monte Carlo in Practice. New York: Chapman & Hall/CRC, 1996.

[5] Häggström, Olle. Finite Markov Chains and Algorithmic Applications. Cambridge: Cambridge University Press, 2002.

[6] Johannes, Michael and Polson, Nicholas. ”MCMC Methods for Continuous-Time Financial Econometrics.” (http://home.uchicago.edu/∼lhansen/JP_handbook.pdf) December 2003.

[7] Paschou P, Ziv E, Burchard EG, Choudhry S, Rodriguez-Cintron W, et al. ”PCA-Correlated SNPs for Structure Identification in Worldwide Human Populations.” PLoS Genetics 3(9): e160. doi:10.1371/journal.pgen.0030160 (2007)

[8] Pritchard JK, Stephens M, and Donnelly P. ”Inference of Population Structure Using Multilocus Genotype Data.” Genetics 155 (June 2000): 945-959.