Stat 260: Statistics in Genetics

A. Bureau, T. P. Speed

Week 7: Markov chain Monte Carlo

1 Introduction

Models built to represent diverse types of genetic data tend to have a complex structure and computations on these structures are often intractable. We see this for instance in pedigree analysis and in biological sequence analysis. One can then turn to stochastic simulation techniques to obtain approximations. Markov chain Monte Carlo (MCMC) is a simulation method that has proved to be useful and work quite well in many problems in genetics. To discuss MCMC, we will need to know a few properties of Markov chain, so we begin by a quick review.

2 Review of Markov chain

2.1 Definition

Markov chains are random processes $x_1, x_2, \cdots$ having a discrete 1-dimensional index set (here 1, 2, $\cdots$), taking values in state spaces $S_1, S_2, \cdots$, and satisfying the Markov property:

- the distribution of $x_{t+1}$ given all the previous states $x_t, x_{t-1}, \cdots$, depends only on the immediately preceding state $x_t$.

The Markov property is a form of conditional independence, restricting the dependence of the distribution of $x_{t+1}$. It can be weakened to permit longer memory; it can be strengthened, e.g. to complete independence, when the dependence of the distribution of $x_{t+1}$ on $x_t$ is absent. There are many variants. From now on, the Markov chains considered will be
of first order, i.e. given the past $x_{t+1}$ depends only on $x_t$.

Let $P = (p_{ij})$ denote the matrix of transition probabilities between $x_t$ and $x_{t+1}$. The transition probabilities between $x_t$ and $x_{t+2}$ are derived in the following way:

$$
Pr(x_{t+2} = s_k \mid x_t = s_i) = \sum_j Pr(x_{t+2} = s_k, x_{t+1} = s_j \mid x_t = s_i)
$$

$$
= \sum_j Pr(x_{t+2} = s_k \mid x_{t+1} = s_j) Pr(x_{t+1} = s_j \mid x_t = s_i)
$$

$$
= \sum_j p_{ij} p_{jk}
$$

So $p_{ik}^{(2)} = Pr(x_{t+2} = s_k \mid x_t = s_i)$ is the $(i, k)$-element of $P^2$. More generally, $p_{ij}^{(n)}$ is the $(i, k)$-element of $P^n$, which means $P^{(n)} = P^n$.

Typically $1, 2, \cdots, t, t+1, \cdots$ are times, but in some examples, they are positions, e.g. loci along a chromosome. The state spaces may coincide, or they may differ.

### 2.2 Simplest example of a Markov chain

We have two unfair coins: coin A, which gives heads with probability 0.9 and tails with probability 0.1; and coin B, which gives heads with probability 0.1 and tails with probability 0.9. **Neither coin has memory** (or conscience).

We define a random process $x_1, x_2, \cdots$ in time, with values in $S = \{head, tail\}$ as follows:

- Begin by tossing coin A. Now, continue as follows:
  - If the last toss gave a head, make the next toss with coin A; otherwise, make it with coin B.

What will you see? Are the successive tosses independent? What do you expect for the long-run frequency of heads? Is it clear why the process is Markov?

In this simple example, $P = (p_{ij})$ is

$$
\begin{pmatrix}
    0.9 & 0.1 \\
    0.1 & 0.9
\end{pmatrix}
$$

and we can easily check by induction that

$$
P^n = \left( \begin{array}{cc}
    \frac{1}{2} + \frac{1}{2} \times (0.8)^n & \frac{1}{2} - \frac{1}{2} \times (0.8)^n \\
    \frac{1}{2} - \frac{1}{2} \times (0.8)^n & \frac{1}{2} + \frac{1}{2} \times (0.8)^n
\end{array} \right) \rightarrow \left( \begin{array}{cc}
    \frac{1}{2} & \frac{1}{2} \\
    \frac{1}{2} & \frac{1}{2}
\end{array} \right)
$$
as \( n \to \infty \). Note that the rows of \( \lim_{n \to \infty} P^n \) are both the same.

**Exercise 1**: Find an expression for:

\[
\left( \begin{array}{cc}
1 - p & p \\
p & 1 - p \\
\end{array} \right)^n
\]

### 2.3 Stationary distributions

More generally, an array \( \pi = (\pi_i) \) with \( \pi_i \geq 0 \) for all \( i \), and \( \sum \pi_i = 1 \), is termed a **stationary** distribution for a Markov chain with transition probabilities \( P = (p_{ij}) \) when for all \( j \),

\[
\sum_i \pi_ip_{ij} = \pi_j.
\]

Writing \( \pi \) as a column vector, this is equivalent to \( \pi'P = \pi' \).

When \( \pi \) is unique for a transition matrix \( P \), the rows of \( \lim_{n \to \infty} P^n \) all coincide with \( \pi' \). In words, the long-run probability of being in state \( j \) is \( \pi_j \), regardless of the initial state.

### 2.4 An example of Markov chain where the stationary distribution is not unique

The following example is a 6-state version of the Gambler’s ruin Markov chain. You can imagine a gambler with \( 0 \leq x \leq 5 \) dollars who wins 1 dollar with probability \( p \) and loses 1 dollar with probability \( 1 - p \) at each turn of a game. The transition matrix of that Markov chain is:

\[
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 - p & 0 & p & 0 & 0 \\
2 & 0 & 1 - p & 0 & p & 0 \\
3 & 0 & 0 & 1 - p & 0 & p \\
4 & 0 & 0 & 0 & 1 - p & 0 \\
5 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Here there is not a unique stationary distribution, as an attempt to solve the previous equations will reveal. Any distribution \( \pi \) for which \( \pi_1 + \pi_6 = 1 \) will satisfy \( \pi'P = p' \).

Indeed, the lines of the \( n^{th} \) power of \( P \) do not tend to the same distribution. For \( p = \frac{1}{2} \),
as \( n \to \infty \). The long-run frequency of being in any given state definitely depends on the initial state. Intuitively, if you start at state 1 you have a greater chance to end up at 0 than to end up at 5, and conversely if you start at state 4. The states 0 and 5 are termed absorbing.

2.5 Some properties of Markov chains

A Markov chain is aperiodic if, loosely speaking, the chances of going from one state to another are not periodic in the number of steps needed, with period \( > 1 \).

A state \( j \) can be reached from a state \( i \) when \( p_{ij}^{(n)} > 0 \) for some \( n \). If state \( j \) can be reached from state \( i \) and state \( i \) can be reached from state \( j \), then states \( i \) and \( j \) communicate. A Markov chain is said to be irreducible if all its states communicate, i.e. every state is accessible from every other state in a finite number of steps (\( \forall i,j \exists n = n(i,j) : p_{ij}^{(n)} > 0 \)). An irreducible aperiodic Markov chain is sometimes called ergodic: such chains have a unique stationary distribution which is positive on all states when the number of states is finite.

A Markov chain has good mixing properties if (loosely speaking) it moves quickly through all possible states.

3 General principles of Markov chain Monte Carlo

Suppose that we want to simulate from \( \pi = (\pi_i) \) so that we can estimate an average \( <\phi> = \sum \pi_i \phi(i) \), but that \( \pi \) is only given up to an unknown and incalculable scale factor \( Z \). We'll see such a case shortly. Such problems arose in statistical physics around the time of the development of the hydrogen bomb after the end of World War II. Metropolis et al. [10] had the idea of defining an ergodic transition matrix \( P = (p_{ij}) \) for which \( \pi'P = \pi' \), i.e. \( \pi \) is the unique stationary distribution.
In fact they arranged that for all $i \neq j$, $\pi_ip_{ij} = \pi_jp_{ji}$. It is an easy check that this implies $\pi$ is stationary for $P$; you just have to sum over $j$:

$$
\sum_j \pi_ip_{ij} = \sum_j \pi_jp_{ji},
$$

i.e. $\pi_i = \sum_j \pi_jp_{ji}$

since $\sum_j p_{ij} = 1$.

Then Metropolis et al. [10] estimated the ensemble average $< \phi >$ by the time average $T^{-1}\sum_{t=b+1}^{b+T} \phi(x_t)$, where $x_t$ is a Markov chain with transition matrix $P$. Here $b$ is the burn-in and $T$ a large number of transitions.

They defined their $P$ indirectly, using another symmetric transition matrix $Q = (q_{ij})$. Their rule goes as follows:

- if we are currently in state $i$, propose state $j$ with probability $q_{ij}$, and move from $i$ to $j$ with probability $\min(1, \pi_j/\pi_i)$; otherwise, stay at state $i$.

Clearly, $p_{ij} = \min(1, \pi_j/\pi_i) \times q_{ij}$, and this can be used to check the symmetry condition above, termed detailed balance. Note that $p_{ij} > 0$ iff $q_{ij} > 0$, and so $P$ is irreducible iff $Q$ is irreducible.

An alternative due to Barker [1] replaces the $\min$, above by $\pi_j/(\pi_i + \pi_j)$, giving $p_{ij} = \pi_jq_{ij}/(\pi_i + \pi_j)$. In this case, detailed balance is obvious if $Q$ is symmetric. There is much choice here.

The Metropolis algorithm to generate a Markov chain with transition matrix $P$ as constructed can be summarized as follows:

- Start with $x_0 = $ any state.
- Given $x_{t-1} = i$, say, choose $j$ with probability $q_{ij}$, i.e. use the $i^{th}$ row of $Q$
- Accept this $j$, i.e. put $x_t = j$ with probability $\min(1, \frac{\pi_j}{\pi_i})$, otherwise put $x_t = i$.

W. K. Hastings [3] brought those ideas to statistics and discussed the efficiency of the algorithm in term of the percentage of acceptance. He also generalized the Metropolis algorithm to allow a non-symmetric $Q$. The probability ratio becomes
\[
h_{ij} = \frac{\pi_j q_{ij}}{\pi_i q_{ji}} = \frac{\pi_j q_{ji}}{\pi_i q_{ij}}
\]

The \(h_{ij}\)'s have come to be called Hastings ratios.

**Exercise 2**: Show that detailed balance is satisfied with the ratios defined above.

### 3.1 A simple application

Suppose we want to simulate from \(\pi(x) = c \exp(-x^4), x \in \mathbb{R}\). Clearly such a density is normalizable, but what if we don’t know or can’t evaluate the constant \(c\)?

The Metropolis algorithm could be applied to this problem by doing the following:

1. Set \(x_0 = 0\).
2. Generate \(x_t^*\) from the proposal distribution \(N(x_{t-1}, 1)\)
3. Compute \(h = \frac{\pi(x_t^*)}{\pi(x_t)} = \exp(-x_t^*^4 + x_t^4)\)
4. 
   \[
x_t = \begin{cases} 
   x_t^* & \text{with probability } \min(1, h_t) \\
   x_{t-1} & \text{otherwise}
   \end{cases}
   \]
5. repeat 2 through 4 a number of times

**Exercise 3**: Prove formally or informally that the stationary density of \((x_t)\) is \(\pi(x)\).

### 3.2 The original application of MCMC

A random process \((x_s)\) with values +1 and -1 defined over the vertices \(s\) of a finite graph \(S\) is called Markov if

\[
pr(x_r = 1 \mid \text{rest}) = pr(x_r = 1 \mid \text{neighbours})
\]

where \(\text{rest} = (x_s : s \neq r)\) and \(\text{neighbours} = (x_s : s \text{ adjacent to } r)\).

One famous example - the Lenz-Ising model - has

\[
\log \left\{ \frac{pr(x_r = +1 \mid \text{rest})}{pr(x_r = -1 \mid \text{rest})} \right\} = \beta \sum_{\text{neighbours}} x_s
\]

(1)
which corresponds to the joint distribution (our \( \pi \)) on \((-1,+1)^S\)

\[
pr(x_s : s \in S) = Z^{-1} \exp \left\{ \frac{1}{2} \sum_{r \sim s} x_r x_s \right\}
\]

where \( r \sim s \) signifies that \( r \) and \( s \) are neighbours. Here \( Z \) - the partition function - is the sum of all \( 2^{|S|} \) exponents, for which typically no closed form expression in \( \beta \) exists.

Note that if two states \( i \) and \( j \) differ only at a site \( r \), say \( x_r = +1 \) for \( j \) and \( x_r = -1 \) for \( i \), then

\[
\frac{\pi_j}{\pi_i} = \frac{pr(x_r = +1, \text{rest})}{pr(x_r = -1, \text{rest})} = \frac{pr(x_r = +1 | \text{rest})}{pr(x_r = -1 | \text{rest})}
\]

is readily computed from equation 1 above. In other words, although we cannot normalize this \( \pi \), we can compute ratios of its elements. Choose a \( Q \) such that \( q_{ij} > 0 \) iff \( i \) and \( j \) differ in value at only a single site. For example, choose a site at random and assign its value at random. Such a \( Q \) is clearly symmetric and irreducible; now use the min rule. Alternatively, Barker’s method uses

\[
\frac{\pi_j}{\pi_i + \pi_j} = \frac{pr(x_r = +1, \text{rest})}{pr(\text{rest})} = pr(x_r = +1 | \text{rest})
\]

i.e. Barker chooses the new value at \( r \) from the conditional distribution given its neighbours. This form is now called a Gibbs sampler. Again, there are many possibilities.

### 3.3 The Gibbs sampler

The Gibbs sampler (Geman and Geman [2]) is a special case of the Metropolis-Hastings algorithm that applies to multivariate probability distributions. At each step, one of the variables making up the state, let’s say \( x_r \), is sampled conditional on the value of all variables excluding \( x_r \), denoted \( x_{-r} \). The distribution \( Pr[x_r | x_{-r}] \) must be simple enough to be computed. The Gibbs sampler is used in situations where each variable depends only on other variables within some well-defined neighborhood, like in the Lenz-Ising model.

A simple example in two dimensions is useful to illustrate the algorithm. Suppose we want to simulate from

\[
\pi(x_1, x_2) = c \exp\{-Q(x_1, x_2)\}
\]

where \( Q(x_1, x_2) \) is a positive definite quadratic form. (Pretend that we don’t know how to normalize this.) Begin with \( x^0 = (x_1^0, x_2^0) = (0, 0) \) or any other pair of real values. Given \( x^{t-1} \), update using
\[
x_1^t | x_2^{t-1} \sim N\left(\mu_1 + \rho \frac{\sigma_1}{\sigma_2} (x_2^{t-1} - \mu_2), \sigma_1^2 (1 - \rho^2)\right)
\]
\[
x_2^t | x_1^t \sim N\left(\mu_2 + \rho \frac{\sigma_2}{\sigma_1} (x_1^t - \mu_1), \sigma_2^2 (1 - \rho^2)\right)
\]

where \(\mu_k, \sigma_k\) are mean and standard deviation of \(x_k, \ k = 1, 2\) and \(\rho\) is the coefficient of correlation between \(x_1\) and \(x_2\).

With the Gibbs sampler, there is no rejection step because the acceptance probabilities are all equal to 1. We see this by writing down the Hastings ratios:

\[
h(x, x^*) = \frac{\pi(x_{-r}, x^*_r)q((x_{-r}, x_r), (x_{-r}, x^*_r))}{\pi(x_{-r}, x_r)q((x_{-r}, x_r^*), (x_{-r}, x^*_r))} = \frac{\pi(x_{-r}, x^*_r)\pi(x_r | x_{-r})}{\pi(x_{-r}, x_r)\pi(x_r^* | x_{-r})} = \frac{\pi(x_{-r}, x^*_r)\pi(x_r, x^*_r)}{\pi(x_{-r}, x_r)\pi(x_r, x^*_r)} = 1
\]

4 Early applications of Monte Carlo methods in pedigree analysis

The first simulation technique used in pedigree analysis was not MCMC but rather the Monte Carlo simulation of independent realizations of multilocus ordered genotypes \(X\). Ott [12] expressed the probability of observed phenotypic data \(Y\) at some fixed value of the vector of parameter \(\theta\) (allele frequencies plus recombination fractions) as an expectation that could be approximated by Monte Carlo simulation.

\[
Pr_\theta[Y] = \sum_X Pr_\theta[Y|X]Pr_\theta[X] = E_{\theta}[Pr_\theta[Y|X]]
\]

The Monte Carlo estimate is then simply

\[
\hat{Pr_\theta}[Y] = \frac{1}{n} \sum_i Pr_\theta[Y|X_i]
\]

where \(X_1, X_2, \ldots\) are independent realizations of \(X\) arising with probability \(Pr[X_1], Pr[X_2], \ldots\).

Ott [12] also mentioned a generalization due to K. Lange:

\[
Pr_\theta[Y] = E_{\theta_0} \left[ Pr_\theta[Y|X] \frac{Pr_\theta[X]}{Pr_{\theta_0}[X]} \right]
\]
The advantage of Lange’s formulation is that it allows estimation of \( Pr_\theta[Y] \) at a range of value of \( \theta \) while simulating at a single value of the parameters \( \theta_0 \). For instance, the recombination fractions between a trait locus and marker loci could be varied.

The problem with this Monte Carlo method is that the distribution of \( X, Pr_\theta[X] \), is not conditional on the data. Most values of \( X \) are incompatible with the data, so that \( Pr_\theta[Y|X] = 0 \). The result is that a large proportion of the simulated \( X \)s contribute nothing to \( Pr_\theta[Y] \).

As an example, consider the pedigree in figure 1. The phenotypes at three codominant marker loci, that is the unordered genotypes, are observed only on the four individuals in the last generation. Each marker has four alleles and we know the order of the marker loci on the genetic map. We want to evaluate the probability of these data under a genetic model. The genetic model is defined by the following simplifying assumptions:

- The genotypes of the four founders are sampled from a population in Hardy-Weinberg equilibrium at each of the three loci.
- The three loci are in linkage equilibrium.
- There is no crossover interference between the three loci.

![Figure 1: Example pedigree with marker data at 3 loci.](image-url)
Ott’s Monte Carlo approach asks to generate multilocus ordered genotypes. Figure 2 shows a realization of multilocus ordered genotypes, one that is actually compatible with the data. To generate such genotypes on a pedigree, we would use gene dropping. In this method, founder genotypes are first simulated according to population frequencies. The genes are “dropped” down the pedigree by simulating gene flow according to Mendel’s laws and the recombination fractions. Let’s compute the probability of generating the genotypes of figure 2. Denote by $p_{ij}$ the probability of allele $j$ at locus $i$. Then the probability of the founder genotypes is $(p_{11}^3 p_{12}^2 p_{13}^1 p_{14}^1)(p_{21}^2 p_{22}^2 p_{23}^1)(p_{31}^2 p_{32}^1 p_{33}^1)$.

In the 12 meioses in this pedigree, there has been 2 recombinations in the first interval and 1 recombination in the second interval. Let $r_1$ and $r_2$ be the recombination fractions for the two intervals. Then the probability of the allele segregation in the pedigree is $(\frac{1}{2})^{12} r_1^2 r_1^{10} r_2 r_2^{11}$.

The probability $Pr_\theta[X]$ is equal to the product of the two above quantities. The probability $Pr_\theta[Y|X]$ is equal to 1, since the codominant marker phenotypes are compatible with the genotypes.

![Figure 2: A realization of ordered genotypes compatible with the marker data in figure 1](image)

The realization of ordered genotypes in figure 2 is only one of $((4^3)^2)^4 \times (2^3)^{12} = 2^{84} \approx 1.6 \times 10^{25}$ possibilities just in this small pedigree (4 alleles at 3 loci on 2 haplotypes for each of 4 founders and 2 possible transmissions at each of 3 loci in 12 meioses). It is here possible
to count that there are 18,648 genotypes compatible with the data at the first marker locus, 29,568 at the second and 71,484 at the third. The total number of tri-locus genotypes is the product of those three numbers and is about $3.94 \times 10^{13}$. This is only a tiny fraction of all possible ordered genotypes. Even if the genotype configurations compatible with the data tend to have higher probability than the incompatible ones, it would on average take a very long time before we see a genotype realization compatible with the data.

4.1 Assessing the power of a genetic study

Another early application of simulation methods is in estimating the power of a linkage study. The idea is to simulate marker data on the pedigrees available for a genetic study under an hypothesized genetic model before genotyping pedigree members.

The method presented here was introduced by Ploughman and Boehnke [13], and Ott [11]. The first step is to simulate genotypes conditional on the observed disease phenotypes. This is done by peeling the pedigree at the hypothesized disease locus in a suitable order up to a final individual, saving all the $\alpha$ and $\beta$ terms computed along the way. The peeling operation gives the joint probability of the genotypes of the last individual $I$ at the disease locus and the disease phenotype data on all the people $P_{\theta}[x_I, Y]$ where $\theta$ represents the parameter values of the chosen disease model (allele frequencies and penetrances).

The genotype of that last individual is then sampled from its distribution conditional on disease phenotype.

$$P_{\theta}[x_I|Y] = \frac{P_{\theta}[x_I, Y]}{\sum_{x_I} P_{\theta}[x_I, Y]}$$

Sampling of the genotypes of each individual then proceeds in reverse peeling order, conditioning on the genotypes already sampled. This gives rise to two possible cases when it comes to process a nuclear family: either the pivot individual whose genotype was sampled first is a parent or it is a child. When the pivot is a parent, the genotype of its spouse is sampled first. Without loss of generality, assume the pivot is the father. Let $anc(i)$ denote the ancestors of individual $i$ and $desc(i)$ his descendents. The genotype of the mother is sampled from:

$$P_{\theta}[x_m|x_p, Y_1, \ldots, Y_m] \propto P_{\theta}[Y_{desc(m)}|x_m, x_p] P_{\theta}[x_m, Y_m, Y_{anc(m)}]$$

The two terms were saved during the peeling operation. If $m$ is a founder, then the second term simplifies to $P_{\theta}[x_m, Y_m]$. 

11
When the pivot is a child $c$ in the nuclear family, the genotype of one of his parents, say his father, is sampled first from:

$$Pr_\theta[x_c|x_c, Y_1, \ldots, Y_p] \propto \sum_{x_m} Pr_\theta[x_p, Y_p, Y_{\text{anc}(p)}] Pr_\theta[x_m, Y_m, Y_{\text{anc}(m)}] Pr_\theta[Y_{\text{desc}(m)c}|x_m, x_p] Pr[x_c|x_m, x_p]$$

The first three terms come from the peeling computations. The first two simplify to $Pr_\theta[x_p, Y_p]$ (respectively $Pr_\theta[x_m, Y_m]$) when the father (respectively the mother) is a founder. The third term simplifies to 1 when the parents have no kid other than the pivot. The next step to sample the mother’s genotype is now conditional on the pivot child genotype. The genotype conditional distribution is given by:

$$Pr_\theta[x_m|x_c, x_p, Y_1, \ldots, Y_m] \propto Pr_\theta[Y_{\text{desc}(m)c}|x_m, x_p] Pr_\theta[x_m, Y_m, Y_{\text{anc}(m)}] Pr[x_c|x_m, x_p]$$

In both of the above cases, the sampling of the genotypes of the children who are not the pivot for that nuclear family proceeds the same way. The probability distribution of the genotype of kid $k$ is:

$$Pr_\theta[x_k|x_m, x_p, Y_1, \ldots, Y_k] \propto Pr[x_k|x_m, x_p] Pr_\theta[Y_k|x_k] Pr_\theta[Y_{>k}|x_k]$$

where $> k$ refers to all the individuals “below” $k$ in the pedigree as defined last week.

Once the disease locus genotypes of all the individuals have been generated, the genotypes at a marker having a recombination fraction $r$ with the disease locus can be simulated by first sampling the founder genotypes according to the marker allele frequencies and then simulating the gene flow down the pedigree according to $r$.

5 MCMC in pedigree analysis

Markov chain Monte Carlo allows to simulate from $Pr_\theta[X|Y]$, the distribution of $X$ conditional on the observed phenotype $Y$, insuring that all realizations of $X$ are compatible with the data. That probability distribution can be expressed as:

$$Pr_\theta[X|Y] = \frac{Pr_\theta[X,Y]}{Pr_\theta[Y]}$$

The denominator is the probability of the data at $\theta$ and is not computable; in fact it is a quantity we are trying to approximate. Notice that $Pr_\theta[Y]$ is a constant throughout the computations over $X$, so the acceptance probabilities of the Metropolis-Hastings algorithm are computed using only ratios of $Pr_\theta[X,Y]$. These joint probabilities must be computable.

Different definitions of $X$ are possible. We will study the two most common options: ordered genotypes and inheritance vectors. Combinations of the two have also been used.
5.1 MCMC with ordered genotypes

The researchers who first applied MCMC to pedigree analysis defined a Markov chain on ordered genotypes. Following Geman and Geman [2], E. A. Thompson and her students used the Gibbs sampler. Around the same time, K. Lange and others [4] used a Metropolis algorithm.

With ordered genotypes, we saw in section 4 that the computation of $Pr_{\beta}[X, Y]$ is easy. However the space of $X$ is huge and it is difficult to design proposals that move far in that space, i.e. that update many genotypes simultaneously.

The Gibbs sampler updating the genotype of one individual at one locus given all else is easy to implement due to the local nature of the dependency. We need to compute $Pr[x_{im}(l), x_{ip}(l)|x_{im}(-l), x_{ip}(-l), x_{-i}, y]$ which involves only the phenotype of $i$ and the genotypes of his parents and children. (If we do not keep track of parental origin but only phase, $x_{im}(l)$ depends also on the spouse genotypes). Furthermore, along the chromosome the genotype at one locus depends only on the genotype at the previous and the following loci under the no-crossover interference assumption. If $i$ is a female, we get the following expression:

$$Pr[x_{im}(l), x_{ip}(l)|x_{im}(-l), x_{ip}(-l), x_{-i}, y] = \frac{Pr[x_{im} | x_m]Pr[x_{ip} | x_p]Pr[y(l)|x_i(l)] \prod_{k \in kids(i)} Pr[x_{km} | x_i]}{\sum_{x_i(l)} Pr[x_{im} | x_m]Pr[x_{ip} | x_p]Pr[y(l)|x_i(l)] \prod_{k \in kids(i)} Pr[x_{km} | x_i]} \tag{2}$$

where $x_i, x_p$ and $x_m$ refer to the multilocus ordered genotypes of individual $i$, his father and mother respectively. In reality only the genotypes at loci $l - 1, l$ and $l + 1$ are needed under the no-crossover interference assumption.

To illustrate a Gibbs proposal, we return to the example pedigree of figure 1. This time, let’s assume that the ordered genotypes on figure 2 are the current state of the Markov chain and for the next Gibbs steps we update the genotype of individual 13 at locus 2. There are three genotypes consistent with the genotype configuration of the other individuals in the pedigree: $1|2|1$ and $1|3$. We first compute the paternal and maternal transmission probabilities for each allele in these three genotypes.

$$Pr \left[ \begin{array}{c} x_{13p} = 2 \\ 3 \end{array} \right] \left[ \begin{array}{c} x_1 = 2 \\ 1 \end{array} \right] = \frac{1}{2} r_1 r_2$$

$$Pr \left[ \begin{array}{c} x_{13p} = 2 \\ 3 \end{array} \right] \left[ \begin{array}{c} x_1 = 1 \\ 2 \end{array} \right] = \frac{1}{2} r_1 r_2$$
\[ Pr \begin{bmatrix} x_{13m} = 4 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} x_2 = 1 \\ 3 \\ 2 \end{bmatrix} = \frac{1}{2} \bar{r}_1 \bar{r}_2 \]

\[ Pr \begin{bmatrix} x_{13m} = 4 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} x_2 = 1 \\ 3 \\ 2 \end{bmatrix} = \frac{1}{2} \bar{r}_1 \bar{r}_2 \]

Then we need the transmission probabilities from individual 13 to his kids.

\[ Pr \begin{bmatrix} x_{23p} = 2 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} x_{13} = 2 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{2} (\bar{r}_1 \bar{r}_2 + \bar{r}_1 \bar{r}_2) \]

\[ Pr \begin{bmatrix} x_{23p} = 2 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} x_{13} = 2 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{2} \bar{r}_1 \bar{r}_2 \]

\[ Pr \begin{bmatrix} x_{23p} = 2 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} x_{13} = 2 \\ 1 \\ 3 \end{bmatrix} = \frac{1}{2} \bar{r}_1 \bar{r}_2 \]

\[ Pr \begin{bmatrix} x_{24p} = 4 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} x_{13} = 2 \\ 1 \\ 3 \end{bmatrix} = \frac{1}{2} \bar{r}_1 \bar{r}_2 \]

Since the phenotype of individual 13 is unobserved, the term for the genotype probability is always 1. We now have all the elements needed to compute the numerator of the right-hand term of equation 2 for each of the three possible genotypes. For genotype 1|1 this gives:

14
$$Pr \left[ \begin{array}{c|c|c}
1_{3p} & 2 \\
1 & 3 \\
\end{array} \right] \quad Pr \left[ \begin{array}{c|c|c}
x_1 & 2 \\
1 & 3 \\
\end{array} \right] \quad Pr \left[ \begin{array}{c|c|c}
x_2 & 4 \\
1 & 2 \\
\end{array} \right] \quad Pr \left[ \begin{array}{c|c|c}
x_3 & 4 \\
1 & 3 \\
\end{array} \right] \quad Pr \left[ \begin{array}{c|c|c}
x_{13} & 2 \\
1 & 3 \\
\end{array} \right]$$

$$\begin{align*}
\frac{1}{16} (\tilde{r}_1 r_2) (\tilde{r}_1 \tilde{r}_2) (r_1 r_2 + r_1 \tilde{r}_2) (r_1 r_2 + \tilde{r}_1 \tilde{r}_2)
\end{align*}$$

Finding the expression for genotypes 2|1 and 1|3 is left as an exercise. Once the three terms have been computed, they are normalized by dividing by their sum and the genotype is sampled from the resulting conditional probability distribution.

**Exercise 4:** Prove for ordered genotypes that the Gibbs proposal is always accepted.

The problem with the Gibbs sampler is that it updates a single genotype at a time, so it is slow and it may be reducible. Lin and Speed [9] present an instance of reducibility. Figure 3 shows the phenotype of two sibs at a marker and figure 4 two states of the ordered genotypes that do not communicate under the Gibbs sampler.

![Diagram of genotypes](chart)

**Figure 3:** Example marker phenotype of two sibs to demonstrate reducibility of the Gibbs sampler

There are a few possible fixes to the reducibility problem. One is to modify the genetic model to allow the chain to step into states inconsistent with the usual rules of genetics (for example a kid with genotype AA that has a parent with genotype BB) (Sheehan and Thomas [7], Sobel and Lange [14]). Impossible states have to be discarded to get valid estimates. Another approach is to find all the irreducible components of the state space, called islands and to
define Metropolis steps allowing to jump from one island to another. This idea is due to S. Lin (Lin [7], Lin and Speed [9]). Lin et al [6] devised an algorithm to identify the islands, but the whole process of finding the islands and setting up the proposal for the Metropolis jumps is not fully automated. The work involved for each different pedigree prevents the widespread use of this method.

5.2 MCMC with inheritance indicators

In recent years, inheritance indicators, (aka as segregation or meiosis indicators) have increasingly been used as latent variables for MCMC instead of ordered genotypes. There are multiple reasons for that. First, the space of inheritance indicators is much smaller than the space of ordered genotypes. Second, inheritance indicators do not always constrain the allelic types of genes since the allelic types of the transmitted genes are not specified, so that sometimes a Markov chain defined on inheritance indicators is irreducible where a Markov chain defined on ordered genotypes is not. Third, the identity by descent configurations between relatives are easier to extract from inheritance indicators than ordered genotypes.

Inheritance indicators can be arranged in a matrix \( v \) where each row \( v_{i*} = (v_{i,l} : l = 1, \ldots, L) \) contains the inheritance indicators for one meiosis at all loci and each column \( v_{*l} = (v_{i,l} : i = 1, \ldots, M)' \) contains the inheritance indicators for all meioses at one locus. The simplest proposal \( q(v^*|v) \) to implement a Metropolis algorithm is to change the grand parental origin of the genes at one locus in one meiosis. The proposal distribution to update meiosis \( i \) at locus \( l \) depends on the current state \( v \) of the inheritance indicators only through the inheritance indicators \( v_{i,-l}, v_{i,l} \) and \( v_{i,l+1} \). The odds ratios of the proposal distribution for a switch from 0 to 1 are given in table 1.

To compute the Hastings ratios, we need to multiply the proposal odds ratio by the ratio of the states probabilities. Recall that

\[
h = \frac{q(v, v^*) Pr[Y, v^*]}{q(v^*, v) Pr[Y, v]} = \frac{Pr[v^*_{i,l}|v_{i,-l}, v]}{Pr[v_{i,l}|v_{i,-l}, v]} \frac{Pr[Y|v^*]}{Pr[Y|v]}
\]
Table 1: Odds ratios of the proposal distribution

| $v_{i,i-1}$ | $v_{i,i}$ | $Pr[v_{i,i} = 1 | v_{i-1,i}]$ | $Pr[v_{i,i} = 0 | v_{i-1,i}]$ |
|-------------|------------|-------------------------------|-------------------------------|
| 1           | 1          | $\frac{\bar{p}_{i-1} \bar{p}_i}{p_{i-1} p_i}$ | $\frac{\bar{p}_{i-1} p_i}{p_{i-1} \bar{p}_i}$ |
| 1           | 0          | $\frac{\bar{p}_{i-1} p_i}{p_{i-1} \bar{p}_i}$ | $\frac{\bar{p}_{i-1} \bar{p}_i}{p_{i-1} p_i}$ |
| 0           | 1          | $\frac{\bar{p}_{i-1} \bar{p}_i}{p_{i-1} p_i}$ | $\frac{\bar{p}_{i-1} \bar{p}_i}{p_{i-1} p_i}$ |
| 0           | 0          | $\frac{\bar{p}_{i-1} \bar{p}_i}{p_{i-1} p_i}$ | $\frac{\bar{p}_{i-1} \bar{p}_i}{p_{i-1} p_i}$ |

To evaluate $Pr[Y_i | v_{si}]$ the same procedure is used as in the Lander-Green algorithm. A detailed description can be found in appendix A of week 8 of the Spring 1998 edition of the lecture notes.

Once again, the pedigree of figure 1 is used as an example. Figure 5 shows one inheritance indicator configuration compatible with the observed genotypes, and we assume that it is the current state of a Markov chain. We propose to switch the paternal indicator of individual 13 at locus 2 from 1 to 0.

Figure 5: The inheritance indicators corresponding to the ordered genotypes in figure 2
We need to compute $Pr[Y_2|v_{*2}]$ and $Pr[Y_2|v_{2*}]$. Here the probabilities can be computed intuitively. In the current configuration $v_{*2}$ the sibs 11 and 13 share no allele identical by descent (IBD) at locus 2, so the alleles in the kids of one are distinct by descent from the alleles in the kids of the other, and the sub-families can be treated separately. The two kids of individual 11 share two alleles IBD. Either the alleles of type 1 comes from individual 11 and the alleles of type 2 comes from individual 12 or it is the reverse. So, the probability in this first family is $2p_{21}p_{22}$. The kids of individual 13 share one allele of type 1 identical by descent and it comes from individual 13 since the paternal indicators of both kids have the same value. The alleles 2 and 3 come from individual 14. The probability in this second family is then $p_{21}p_{22}p_{23}$. Taking the product of the two, we get $Pr[Y_2|v_{*2}] = 2p_{21}^2p_{22}^2p_{23}$. 

When we switch the paternal indicator of individual 13 at locus 2 from 1 to 0 the four kids in the third generation are forced to share an allele IBD. It has to be allele 1 since it is the only one found in all four kids. The alleles of type 2 in individuals 21 and 22 are still IBD, and now must come from individual 12, since allele 1 comes from individual 11. As before, in the second family the alleles 2 and 3 come from individual 14. This results in $Pr[Y_2|v_{*2}] = p_{21}p_{22}^2p_{23}$. We can now write down the Hastings ratio:

\[ h = \frac{\bar{r}_1 r_2}{r_1 \bar{r}_2} \frac{p_{21}^2 p_{22}^2 p_{23}}{2p_{21}^2 p_{22}^2 p_{23}} = \frac{r_1 r_2}{r_1 \bar{r}_2} \frac{1}{2p_{21}} \]

The proposal to update a single inheritance indicator suffers from low acceptance rates if the loci are tightly linked. For instance, if we have a string of indicators 000 at three loci separated by small genetic distances and we switch the indicator at the locus in the middle, the proposed state 010 with two recombinations will have much lower probability than the original state (unless the phenotype data favor the proposed state).

5.2.1 The whole meiosis Gibbs sampler

We have seen last week that the Lander-Green algorithm, which is a version of the forward-backward algorithm, allows us to perform computation on the HMM formed by phenotype data and inheritance vectors along a chromosome with a complexity linear in the number of loci. The exponential growth of the number of inheritance vectors at each locus makes it impractical for large pedigrees, but Thompson and Heath [18] realized that it was possible to take advantage of the linear complexity in the number of loci to implement a proposal to sample the inheritance indicators for one meiosis at all loci. Because this type of proposal updates more inheritance indicators and does not need to introduce extra recombinations to make a move, it has a higher acceptance rate and moves more quickly over the space of inheritance indicators than a proposal to update a single inheritance indicator. In that
sense, it is more efficient.

This proposal is set up as a Gibbs sampler, which means that the inheritance indicators for one meiosis at the whole array of loci $v_i$ will be sampled from their conditional distribution given all others inheritance indicators and given the phenotype data $Pr[v_{i*}]|{v_{k*}, k \neq i}, Y]$. Thompson and Heath [18] showed how it can be done by adapting the Lander-Green algorithm to do the computation for a single meiosis, holding all other inheritance indicators fixed.

The $\alpha$ terms of the recursion are redefined for a single meiosis $i$. We write $OMI = \{v_{k*}, k \neq i\}$ for the other meioses indicators.

$$\alpha_1(v_{i,1}) = Pr[Y_1, v_{i,1}|OMI] = Pr[Y_1|v_{*,1}]Pr[v_{i,1}]$$

$$\alpha_i(v_{i,i}) = Pr[Y_1, \ldots, Y_i, v_{i,i}|OMI] = \alpha_1(v_{i,i-1})Pr[v_{i,i}|v_{i,i-1}]Pr[Y_i|v_{*,i}]$$

The term $Pr[v_{i,i}|v_{i,i-1}]$ is the simple $2 \times 2$ matrix

$$\begin{bmatrix}
1 - r_{i-1} & r_{i-1} \\
r_{i-1} & 1 - r_{i-1}
\end{bmatrix}$$

and $Pr[Y_i|v_{*,i}]$ is computed like in the Lander-Green algorithm.

At the end of this forward iteration, we get $\alpha_L(v_{i,L}) = Pr[Y, v_{i,L}|OMI]$ which is easily normalized to $Pr[v_{i,L}|Y, OMI]$ since there are only two possible values of $v_{i,L}$. Now $v_{i,L}$ is sampled from its distribution conditional on phenotype data and inheritance indicators at other meioses, as the Gibbs sampler prescribes. Other inheritance indicators in $v_i$ are sampled iteratively in a backward order. Suppose $v_{i,L}, \ldots, v_{i+1}$ have been sampled. Then $v_{i,L}$ is sampled from

$$Pr[v_{i,i}|v_{i,i+1}, \ldots, v_{i,L}, Y, OMI] = \frac{\alpha_L(v_{i,i})Pr[v_{i,i+1}|v_{i,i}]}{\alpha_L(0)Pr[v_{i,i+1} = 0|v_{i,i}] + \alpha_L(1)Pr[v_{i,i+1} = 1|v_{i,i}]}$$

The end result is a new realization of $v_{i*}$.

5.2.2 Irreducibility issue

Even though Markov chains defined on inheritance indicators are less prone to reducibility than chains defined on ordered genotypes, a Gibbs sampler updating one meiosis at a time is reducible in many cases. For the example of figure 3, the valid states for $v$ are
No transition is possible with the update of only one meiosis. Here we can get an irreducible chain by updating all paternal or maternal indicators. This is not a general solution for multigenerational pedigrees where more constraints on allelic identity are present.

5.2.3 The whole locus Gibbs sampler

The whole meiosis Gibbs sampler takes advantage of the linear complexity in the number of loci of the Lander-Green algorithm to update all meiosis indicators for one meiosis. The reverse peeling algorithm of Ploughman and Boehnke [13] and Ott [11] described in section 4.1 samples the genotypes of all the individuals at one locus conditionally on their phenotype in an efficient way. It is not too difficult to adapt the reverse peeling algorithm to update all the inheritance indicators at one locus and define a whole locus Gibbs sampler. Two modifications are required. First, the ordered genotype at the updated locus \( l \) must be sampled given not only the marker phenotype at locus \( l \) but also the inheritance indicators at flanking loci. This is achieved by replacing the term \( Pr[x_{c,l} | x_{m,l}, x_p,l] \) by

\[
Pr[x_{c,l}, v_{cp,l-1}, v_{cm,l-1}, v_{cp,l+1}, v_{cm,l+1}|x_{p,l}, x_{m,l}] = Pr[x_{c,l}, x_{p,l}, x_{m,l}] Pr[v_{cp,l-1}|x_{cp,l}, x_{p,l}] Pr[v_{cp,l+1}|x_{cp,l}, x_{p,l}] Pr[v_{cm,l-1}|x_{cm,l}, x_{m,l}] Pr[v_{cm,l+1}|x_{cm,l}, x_{m,l}]
\]

The last four terms are similar; the first one is given below:

\[
Pr[v_{cp,l-1}|x_{cp,l}, x_{p,l}] = \begin{cases} \frac{1}{r_{l-1}} r_{l-1}^{-1}(1-r_{l-1})^{1-r_{cp,l-1}} & \text{if } x_{cp,l} = x_{pp,l} \\ (1-r_{l-1}) v_{cp,l-1} r_{l-1}^{-1} & \text{if } x_{cp,l} = x_{pm,l} \end{cases}
\]

Once the ordered genotypes at locus \( l \) of all the individuals in the pedigree have been sampled using reverse peeling, it remains to convert the ordered genotypes into inheritance indicators. For informative meioses, i.e. when the parent is heterozygous, it is known whether the paternal or maternal allele is transmitted, and this determines the inheritance indicator. For uninformative meioses, i.e. when the parent is homozygous, the inheritance indicator \( v_{i,l} \) is sampled conditionally on \( v_{i,l-1} \) and \( v_{i,l+1} \).

An important advantage of the whole locus Gibbs sampler is that it is irreducible, provided that all recombination fractions between loci are strictly positive. There is no need to identify islands of communicating states for each new pedigree; the Markov chain on the space
of inheritance vectors is setup automatically.

The locus sampler is still subject to slow mixing when loci are tightly linked for the same reason as for the single indicator sampler: the low probability of moving to a state with more recombination events than the current one. To improve mixing, a good approach is to alternate whole meiosis updates and whole locus updates. As long as some locus updates are performed, the chain combining the two types of steps will be irreducible. The relative proportions of the two steps will optimize mixing depends on the pedigree analyzed and has to be set empirically. Since the locus sampler tends to require much more computing time than the meiosis sampler, it makes sense to perform many more meiosis updates than locus updates, maybe in a 10 to 1 ratio.

5.3 Computation of linkage statistics

Now that we know how to simulate from $Pr_{\theta}[X|Y]$, we need to see how we can apply it to linkage analysis in pedigrees where a some individuals are affected by a genetic disease. We will only consider the parametric approach in which a genetic model for the trait is specified. The simple models used in pedigree analysis usually include a single major gene with two alleles. In this case, the disease phenotype depends only on the genotype at one trait locus. The trait locus is like any other locus on the map, except that its position is unknown. The common approach is to compute likelihood ratios between different locations for the trait locus. When we perform MCMC, we make use of the following equality (Thompson and Guo [16], Thompson [17]):

$$\frac{L(\theta)}{L(\theta_0)} = E_{\theta_0} \left[ \frac{Pr_{\theta}[X,Y]}{Pr_{\theta_0}[X,Y]} \right]$$

(3)

where $Y$ includes both marker and disease phenotype. The recombination fractions between the trait locus and its flanking markers and possibly the interval in which the trait locus is placed differ between $\theta$ and $\theta_0$.

Proof of equality 3:

$$\frac{L(\theta)}{L(\theta_0)} = \frac{Pr_{\theta}[Y]}{Pr_{\theta_0}[Y]} = \sum_X \frac{Pr_{\theta}[X,Y]}{Pr_{\theta_0}[X,Y]}$$

$$= \sum_X \frac{Pr_{\theta}[X,Y]Pr_{\theta_0}[X,Y]}{Pr_{\theta_0}[X,Y]}$$

$$= \sum_X Pr_{\theta}[X,Y]Pr_{\theta_0}[X|Y]$$

21
\[ E_{\theta_0} \left[ \frac{Pr_\theta[X, Y]}{Pr_{\theta_0}[X, Y]} \right] \]

The MCMC estimate of the likelihood ratio is given by

\[ \frac{1}{N} \sum_{j=b+1}^{N+b} \frac{Pr_\theta[X_j, Y]}{Pr_{\theta_0}[X_j, Y]} \]

which again involves only ratios of joint probabilities of \( X \) and \( Y \).

We can estimate the likelihood ratio over a range of values of \( \theta \) by simulating at a single value \( \theta_0 \). One issue here is the choice of \( \theta_0 \). The variance of the estimate will be lowest for \( \frac{Pr_\theta[X, Y]}{Pr_{\theta_0}[X, Y]} \) close to 1, that is for \( \theta \) close to \( \theta_0 \). Thompson [17] has suggested placing the trait locus at a number of locations along the marker map and to run a separate chain at each location. This gives good estimates of \( LR(\theta) \) for values of \( \theta \) that place the trait locus in the same interval as where it is under \( \theta_0 \). The disadvantage of the method is that it requires a complicated reweighting scheme to combine realizations generated under different values \( \theta_0 \) and estimate likelihood ratios between distant locations for the trait locus, for instance a point in an inter-marker interval against an unlinked location.

E. Sobel and K. Lange [4, 15] proposed instead to simulate only with the trait locus unlinked to the map of markers and further to exclude the inheritance indicators at the trait locus from the Markov chain states. Only the inheritance indicators at marker loci are simulated conditional on marker phenotype. Equation 3 can still be applied because disease phenotype is independent from marker phenotypes when the trait locus is unlinked, under a model where trait locus and marker loci are in linkage equilibrium. For every realization of that reduced \( X \), the probability ratio \( \frac{Pr_\theta[X, Y]}{Pr_{\theta_0}[X, Y]} \) can be evaluated over a range of values of \( \theta \) covering the whole genetic map. We no longer have latent variables at the trait locus, so we need to apply pedigree peeling (Elston-Stewart) at that locus, but under the Markov assumption which states that \( Pr[X_i|X_{-i}] = Pr[X_i|X_{i-1}, X_{i+1}] \), the computation of the probability of the disease phenotypes involves only the latent variables \( X \) at the marker loci flanking the trait locus. This holds whether \( X \) is ordered genotypes or inheritance indicators.

One might think that because the simulation is performed with the disease locus unlinked, the estimates will suffer from high variability. In practice variance is not too large, and being able to estimate the whole likelihood ratio curve in one run is a big advantage. Making more iterations of the chain to compensate for the increased variability usually takes less time than running multiple chains.
6 Further readings


References


