Introduction — This is not a biology class!

For the purposes of this class, DNA is nothing but a sequence of the four letters A, C, G, T.

Even though a DNA molecule is actually TWO such (complementary) sequences wound together into a helix, only one of the sequences gives rise to genetic instructions so we ignore the double-helical structure of DNA.

A gene, for our purposes, is a DNA segment recognizable by its specific function.

Of course, there is much more to the story than is outlined here; however, I will try to cover only as much of the biology is needed for each new topic.

Each (human) gene is located on one of 23 chromosome pairs. 22 of these are called autosomes; the 23rd pair is called the sex chromosomes. Each autosomal pair consists of two chromosomes of the same size, one from the mother and one from the father. Thus, at each position (locus) on the chromosome, there is a built-in redundancy: One maternal gene and one paternal gene. (This is not QUITE true of the 23rd chromosome; females possess this redundancy but males do not.)

The specific DNA sequence found at a locus may have several forms, known as alleles. Thus, we may say that at each locus, there is a maternal allele and a paternal allele.

Since both the mother and father also have 2 alleles each, the two alleles that the child inherits at a particular locus are not (necessarily) uniquely determined simply because of the parents’ alleles. Mendel’s first law states that each of the two paternal alleles is equally likely to be passed to the offspring, independently of any other offspring of the father and independently of the allele passed by the mother, which is also equally likely to be either of the two possible alleles.
The blood type locus

The canonical example of a locus and its alleles is the blood type locus, or ABO locus. It is located on chromosome 9.

There are 3 alleles that can occur at this locus—A, B, and O. Since each individual receives one maternal allele and one paternal allele, there are 6 possible sets of two alleles an individual can have: A/A, A/B, A/O, B/B, B/O, and O/O. This set is called an individual’s genotype for this locus. Notice that a genotype is unordered, which means that it does not contain information about which parent contributed which allele. (In some cases, it may be helpful to deal with ordered genotypes; in the ABO case, there are 9 possible such genotypes.)

The ABO locus determines a person’s blood type, which can be A, B, AB, or O. Any observable trait arising from a genotype is called a phenotype; thus, there are 4 blood-type phenotypes.

The A phenotype arises from either the A/O genotype or the A/A genotype. Thus, the A allele is said to be dominant with respect to the O allele. Equivalently, O is recessive with respect to A. Similarly, B is dominant with respect to O. Thus, the O phenotype may only arise from the O/O genotype.

The AB phenotype may only arise from the A/B genotype. Since the A/B genotype produces a phenotype different from that of both A/A and B/B, the A and B alleles are said to be codominant.

A locus that has several common alleles (like the ABO locus) is called polymorphic.
Population allele frequencies and Hardy-Weinberg equilibrium

Consider a polymorphic locus with $k$ possible alleles. For the sake of concreteness, we'll take $k = 2$, though the results below may be generalized to larger $k$.

Denote the two alleles by A and B. Each occurs in the population with some frequency. Denote the frequencies $p_A$ and $p_B$. Thus, we must have $p_A + p_B = 1$.

There are three possible (unordered) genotypes: A/A, A/B, B/B. A child inherits the A/A genotype if and only if she gets the A allele from each parent. If we assume that the father passes the A allele with probability $p_A$, the mother passes the A allele with probability $p_B$, and these two events are independent, then the A/A genotype occurs in the child with probability $p_A^2$. Similarly, under these assumptions, the B/B genotype occurs with probability $p_B^2$. Note that there are two mutually exclusive ways that the A/B genotype could occur, so this genotype occurs with probability $2p_A p_B$.

The proportions $p_A^2$, $2p_A p_B$, $p_B^2$ are called the **Hardy-Weinberg proportions**; a population in which A/A occurs with probability $p_A^2$, A/B with probability $2p_A p_B$, and B/B with probability $p_B^2$ is said to be in **Hardy-Weinberg equilibrium**.

The Hardy-Weinberg proportions are an equilibrium in the following sense: Under a certain set of simplifying assumptions, the genotype frequencies stabilize at the proportions, as we now prove.
Approach to Hardy-Weinberg equilibrium

Assume the following about the population in question:

1. Infinite population size
2. Discrete generations
3. Random mating
4. No selection
5. No migration
6. No mutation
7. Equal initial genotype frequencies in the two sexes.

Recall that the locus in question is **diallelic** (i.e., there are two possible alleles) and the proportions of the alleles A and B are \( p_A \) and \( p_B \). Assume this locus is autosomal (i.e., not located on a sex chromosome). At generation 0, denote the genotype frequencies of A/A, A/B, and B/B by \( p_{AA} \), \( p_{AB} \), and \( p_{BB} \), respectively. What will the next generation look like?

The proportion of A/A in generation 1 may be computed by an argument such as the following:

If we consider matings between an A/A individual and an A/B individual, such matings produce 2\( p_{AA} p_{AB} \) of the offspring in generation 1. By Mendel’s first law, these offspring are A/A with probability 1/2 and A/B with probability 1/2. Thus, a proportion \( \frac{1}{2} (2p_{AA} p_{AB}) \) of generation 1 will be A/A offspring from A/A×A/B matings. This calculation and the remaining calculations are summarized in the table below:

<table>
<thead>
<tr>
<th>Parents' genotypes</th>
<th>Frequency of this type of pair</th>
<th>Offspring</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( p_{AA} )</td>
<td>( p_{AA} )</td>
<td>0</td>
</tr>
<tr>
<td>A/A, A/A</td>
<td>( 2p_{AA} p_{AB} )</td>
<td>( p_{AA} )</td>
<td>0</td>
</tr>
<tr>
<td>A/A, A/B</td>
<td>( 2p_{AA} p_{BB} )</td>
<td>( p_{BB} )</td>
<td>0</td>
</tr>
<tr>
<td>A/B, A/B</td>
<td>( p_{AB}^2 )</td>
<td>( p_{AB}^2 )</td>
<td>0</td>
</tr>
<tr>
<td>A/B, B/B</td>
<td>( 2p_{AB} p_{BB} )</td>
<td>( p_{AB} )</td>
<td>0</td>
</tr>
<tr>
<td>B/B, B/B</td>
<td>( p_{BB}^2 )</td>
<td>( p_{BB} )</td>
<td>0</td>
</tr>
</tbody>
</table>

| Totals             | \( \frac{1}{2} p_{AA} + \frac{1}{2} p_{AB} \) | \( \frac{1}{4} p_{AA} + \frac{1}{2} p_{AB} \) | \( \frac{1}{2} p_{AB} \) |

Note that the Hardy-Weinberg proportions have now been achieved, since \( p_A = (p_{AA} + \frac{1}{2} p_{AB}) \) and \( p_B = (p_{BB} + \frac{1}{2} p_{AB}) \). Therefore, regardless of the initial values \( p_{AA}, p_{AB}, \) and \( p_{BB} \), Hardy-Weinberg equilibrium is attained after a single generation in this idealized model.
Approach to Hardy-Weinberg equilibrium (cont’d)
X-linked loci

Consider our idealized population for a locus on the X sex chromosome. Once again, suppose the
locus is diallelic with alleles A and B having frequencies $p_A$ and $p_B$. Then the possible genotypes for
a female are $A/A$, $A/B$, and $B/B$, whereas they are simply $A$ and $B$ for a male (note that assumption
7 doesn’t make sense in this case).

We now change notation slightly, keeping track only of the proportion of $A$ alleles for each sex in
each generation. We thus define $f_n$ and $m_n$ as the proportion of $A$ alleles for females and males,
respectively, at generation $n$.

What should Hardy-Weinberg equilibrium look like in this case? Assuming that females and males
each comprise $1/2$ of the population, note that the initial population proportion of $A$ alleles is
$(2f_0 + m_0)/3$. We will show that in fact, $f_n$ and $m_n$ both converge to this value as $n$ increases (even
without assuming equal numbers of males and females). Set $p = (2f_0 + m_0)/3$.

Since a male always inherits his X-chromosome from his mother, we know $m_n = f_{n-1}$ for $n \geq 1$.

Each of a female’s X-chromosomes is equally likely to come from either parent, so

$$f_n = (f_{n-1} + m_{n-1})/2 \quad \text{for } n \geq 1. \quad (1)$$

Doubling equation (1) and adding $m_n = f_{n-1}$ gives

$$2f_n + m_n = 2f_{n-1} + m_{n-1} \quad \text{for all } n \geq 1. \quad (2)$$

Note that equation (2) implies $(2f_n + m_n)/3 = p$ for all $n$.

How fast does $f_n$ approach $p$, the equilibrium point? Consider

$$f_n - p = \frac{1}{2}f_{n-1} + \frac{1}{2}m_{n-1} - \frac{3}{2}p + \frac{1}{2}p = \frac{1}{2}f_{n-1} + \frac{1}{2}m_{n-1} - \frac{3}{2}\left(\frac{2}{3}f_{n-1} + \frac{1}{3}m_{n-1}\right) + \frac{1}{2}p = -\frac{1}{2}(f_{n-1} - p).$$

Therefore, we obtain

$$f_n - p = \left(-\frac{1}{2}\right)^n (f_0 - p),$$

which means that the female allele frequency approaches equilibrium geometrically, halving the
difference each generation. Since $m_n = f_{n-1}$, the male allele frequency approaches equilibrium at
the same rate but lags the female frequency by one generation.
The MN blood group locus and gene counting

The MN blood group locus is a diallelic locus with alleles M and N. Its function is not important to us; instead, we merely note that the alleles M and N are codominant, which means that each of the three genotypes (M/M, M/N, and N/N) gives rise to a different phenotype.

Interesting aside: This group seems to follow Hardy-Weinberg proportions pretty nicely. Summarized below are data on the frequencies of MM, MN, and NN blood groups for a sample of 6129 Americans of European descent (thanks for this dataset, Neill):

<table>
<thead>
<tr>
<th></th>
<th>MM</th>
<th>MN</th>
<th>NN</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observed number</td>
<td>1787</td>
<td>3039</td>
<td>1303</td>
<td>6129</td>
</tr>
<tr>
<td>Expected number</td>
<td>1783.8</td>
<td>3045.4</td>
<td>1299.8</td>
<td>6129.0</td>
</tr>
<tr>
<td>Expected proportion</td>
<td>.291</td>
<td>.497</td>
<td>.212</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Sample frequency of M allele = $p_M = (2 \times 1787 + 3039)/(2 \times 6129) = .539$.
Sample frequency of N allele = $p_N = (2 \times 1303 + 3039)/(2 \times 6129) = .461$.
Expected proportion of MM = $p_M^2 = .291$.
Expected proportion of MN = $2p_Mp_N = .497$

Notice something about the table above: In order to compute the expected proportions under Hardy-Weinberg, it was necessary to estimate the allele frequencies. The method used, counting the number of each type of allele in the sample and dividing by the total number of alleles in the sample, is sometimes called gene counting.

The next major topic in the course is a general method of statistical estimation, called maximum likelihood estimation, that will figure very prominently in many of the topics to follow. Estimation of allele frequencies in the MN blood group locus by gene counting will serve as our first example of maximum likelihood estimation.
Probability densities

One thing I hope you’ve seen before in a probability class is the topic of probability densities or probability mass functions. As a quick refresher, we’ll consider the specific example of the MN blood group locus and its allele frequencies.

First, a quick note on terminology: You may recall that we usually speak of a continuous random variable (or vector) having a probability density function, while a probability mass function plays the analogous role for a discrete random variable (or vector). The distinction between these two types of functions is unimportant for our purposes, so I’ll refer only to densities, even for discrete variables.

To build an example of a density, consider the MN locus. Assume that the allele frequencies of M and N are, respectively, \( p_M \) and \( p_N \) (thus \( p_M + p_N = 1 \)). Also assume that the population in question is in Hardy-Weinberg equilibrium. Let \( n \) denote the size of a sample of individuals to be selected at random and phenotyped.

Under these assumptions, what is the probability of seeing \( a \) MM’s, \( b \) MN’s, and \( c \) NN’s? (Assume that \( a + b + c = n \), or else the probability is zero). The probability that any one particular individual selected is MM, MN, or NN equals \( p_M^2 \), \( 2p_Mp_N \), or \( p_N^2 \), respectively — this is the Hardy-Weinberg assumption. Thus, any particular arrangement of \( a \) MM’s, \( b \) MN’s, and \( c \) NN’s occurs with probability \( p_M^{2a}(2p_Mp_N)^b p_N^{2c} \). There are \( \binom{n}{a,b,c} \) arrangements possible, so the probability we seek is

\[
\binom{n}{a,b,c} p_M^{2a}(2p_Mp_N)^b p_N^{2c}.
\]

(3)

If you don’t know what \( \binom{n}{a,b,c} \) means, don’t worry about it too much; because it does not depend on \( p_M \) and \( p_N \), it will turn out to be unimportant in maximum likelihood estimation.

The density (3) expresses the probability of seeing a particular set of data (namely, \( a \), \( b \), and \( c \)) under a model (namely, Hardy-Weinberg equilibrium) governed by a particular set of parameters (namely, \( p_M \) and \( p_N \)). We often write a density as a function of the data:

\[
f(a,b,c) = \binom{n}{a,b,c} p_M^{2a}(2p_Mp_N)^b p_N^{2c}.
\]
Likelihood functions and maximum likelihood estimation

Recall the density function from the previous page:

\[
f(a, b, c) = \left( \frac{n}{a, b, c} \right)^2 (2p_M p_N)^b (2p_M p_N)^c.
\]  

On the previous page, we considered the parameters to be fixed and \( f(\cdot) \) to be a function of the data \( a, b, c \); this is why the function above is considered a density function. However, if we instead suppose that the data are fixed (say, because we have observed them) and view the right side of (4) as a function of the parameters \( p_M \) and \( p_N \), then the very same expression is called the likelihood function. Usually we denote the likelihood function by the letter \( L(\cdot) \); thus, in the case of the MN blood group locus, we have

\[
L(p_M, p_N) = \left( \frac{n}{a, b, c} \right)^2 (2p_M p_N)^b (2p_M p_N)^c.
\]  

Because equation (5) is a real-valued function of \( p_M \) and \( p_N \), it makes sense to talk of the maximum of this function. The value of \((p_M, p_N)\) that maximizes the likelihood, usually denoted \((\hat{p}_M, \hat{p}_N)\), is called (for obvious reasons) the maximum likelihood estimator, or MLE, of \((p_M, p_N)\).

Maximum likelihood estimators have many appealing properties that we won’t discuss here; suffice it to say that maximum likelihood estimation will play a large role in this course.

Intuitively, the maximum likelihood estimator is the value of the parameter(s) for which the observed data are most probable.

Notice that it is redundant to use \( p_N \) in expression (5) since \( p_N = 1 - p_M \). Therefore, we can rewrite everything in terms of \( p_M \):

\[
L(p_M) = \left( \frac{n}{a, b, c} \right)^2 [2p_M (1 - p_M)]^b (1 - p_M)^c.
\]  

Our goal is to find the value of \( p_M \) that maximizes the function above.
More on maximum likelihood estimation

Before we do a little bit of simple calculus to find the maximizer of expression (6), we’ll employ a
very common mathematical trick and take the logarithm of the likelihood function. We can do this
because finding a maximizer of \( L(p_M) \) is equivalent to finding a maximizer of \( \log L(p_M) \). It turns out
to be convenient mathematically to work with \( l(p_M) = \log L(p_M) \) instead of \( L(p_M) \). The function
\( l(p_M) \) is called the loglikelihood function, for obvious reasons. In our case, the loglikelihood
function is

\[
l(p_M) = \log \left( \frac{n}{a, b, c} \right) + 2a \log p_M + b \log [2p_M (1 - p_M)] + 2c \log (1 - p_M).
\]

(By the way, when I write \( \log \), I always mean natural logarithm, or logarithm base \( e \).)

To find the maximizer of the loglikelihood, take the derivative with respect to \( p_M \):

\[
l'(p_M) = \frac{2a}{p_M} + \frac{b(1 - 2p_M)}{p_M(1 - p_M)} - \frac{2c}{1 - p_M} = \frac{2a(1 - p_M) + b(1 - 2p_M) - 2cp_M}{p_M(1 - p_M)}.
\]

Setting the derivative equal to zero and solving for \( p_M \) gives \( \hat{p}_M = (2a + b)/(2a + 2b + 2c) \), or
\( \hat{p}_M = (2a + b)/2n \). This implies that \( \hat{p}_N = (2c + b)/2n \), and these maximum likelihood estimates
of \( p_M \) and \( p_N \) are exactly the same estimates we used earlier in the MN blood group example. To
make this more concrete, let’s recall the actual dataset:

<table>
<thead>
<tr>
<th>Observed number</th>
<th>MM</th>
<th>MN</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1787</td>
<td>3039</td>
<td>1303</td>
<td></td>
</tr>
</tbody>
</table>

Thus \( a = 1787, b = 3039, \) and \( c = 1303 \) and so we have, as before,

\[
\hat{p}_M = \frac{2(1787) + 3039}{2(1787 + 3039 + 1303)} \quad \text{and} \quad \hat{p}_N = 1 - \hat{p}_M = \frac{2(1303) + 3039}{2(1787 + 3039 + 1303)}.
\]

In summary, the maximum likelihood estimator for the population allele frequencies for the MN
blood group under Hardy-Weinberg equilibrium is quite simple to derive and it coincides with the
gene counting result.

I hope that seeing that the MLE gives a reasonable answer in this simple case will give you a bit of
faith in the method of maximum likelihood estimation since we next move to an example in which
gene counting is not possible in the same way and it is not so easy to compute the MLE.
Estimating ABO locus allele frequencies

Consider once again the ABO locus with genotypes A/A, A/O, B/B, B/O, A/B, and O/O. Assume that when we collect data we do not have individuals’ genotypes available, only their phenotypes (A, B, AB, or O). We collect a sample of n individuals and denote the number of A by \(n_A\), the number of B by \(n_B\), etc., so that \(n_A + n_B + n_{AB} + n_O = n\).

Writing the likelihood in this case is similar to last time, though with an extra twist: If \(p_A\), \(p_B\), and \(p_O\) are the population allele frequencies (and we assume Hardy-Weinberg equilibrium), then the probability of an A phenotype is actually \(p_A^2 + 2p_Ap_O\). Of course, the B phenotype is similar. The likelihood looks like this:

\[
L(p_A, p_B, p_O) = \prod_{i=1}^{n} \left( p_A^2 + 2p_Ap_O \right)^{n_A} \left( p_B^2 + 2p_Bp_O \right)^{n_B} \left( 2p_Ap_B \right)^{2n_O} \tag{8}
\]

Once again, one parameter above, say \(p_O\), is redundant since \(p_A + p_B + p_O = 1\), so after substituting \(1 - p_A - p_B\) for \(p_O\) and taking the logarithm, we obtain the loglikelihood

\[
l(p_A, p_B) = \log \left( \frac{n_A}{n} \right) + n_A \log \left( p_A^2 + 2p_A(1-p_A-p_B) \right) \\
+ n_B \log \left( p_B^2 + 2p_B(1-p_A-p_B) \right) + n_{AB} \log \left( 2p_Ap_B \right) + 2n_O \log \left( 1-p_A-p_B \right)
\]

If we take partial derivatives with respect to \(p_A\) and \(p_B\) and set them equal to zero, we get a royal mess! Therefore, we will come up with a different way to find the maximum likelihood estimator in this case, namely, the EM algorithm.

But first, consider the following clever method for estimating the allele frequencies. If we knew the allele frequencies, then we could estimate the frequencies of all 6 genotypes, since for example the expected proportion of the A phenotypes that are A/A is \(\hat{n}_{A/A} = \frac{p_A^2}{p_A^2 + 2p_Ap_O}\). Thus, we could set

\[
\hat{n}_{A/A} = n_A \left[ \frac{p_A^2}{p_A^2 + 2p_Ap_O} \right], \quad \hat{n}_{A/O} = n_A \left[ \frac{2p_Ap_O}{p_A^2 + 2p_Ap_O} \right], \\
\hat{n}_{B/B} = n_B \left[ \frac{p_B^2}{p_B^2 + 2p_Bp_O} \right], \quad \hat{n}_{B/O} = n_B \left[ \frac{2p_Bp_O}{p_B^2 + 2p_Bp_O} \right],
\]

and of course \(\hat{n}_{A/B} = n_{AB}\) and \(\hat{n}_{O/O} = n_O\). But if we could estimate all the genotype frequencies, then we could estimate the allele frequencies by gene counting just as we did in the MN blood group example, namely \(\hat{p}_A = (2n_{A/A} + n_{A/O} + n_{A/B})/2n\), \(\hat{p}_B = (2n_{B/B} + n_{B/O} + n_{A/B})/2n\), and \(\hat{p}_O = (2n_{O/O} + n_{B/O} + n_{A/O})/2n\).

The argument is circular: We started with estimates of allele frequencies and ended with the same. Nonetheless, if we iterate repeatedly in this way, alternating between updating the genotype frequencies and updating the allele frequencies, it turns out that eventually the allele frequencies converge to a solution. All we need is a starting value of \((p_A, p_B, p_O)\), which can be taken to be anything, such as \((1/3, 1/3, 1/3)\).
Iterative algorithms

The Oxford English Dictionary defines iteration as the repetition of an operation upon its product. Thus, an iterative algorithm is one that proceeds in discrete steps, operating at each step on the result of the previous step. The ABO allele frequency estimation method mentioned at the bottom of the previous page is an example of an iterative algorithm because it repeatedly uses one value of \((\hat{p}_A, \hat{p}_B, \hat{p}_O)\) to produce another value of \((\hat{p}_A, \hat{p}_B, \hat{p}_O)\).

Let’s see how this method works on an actual example. Since the algorithm proceeds in steps, or iterations, it will be convenient to adopt notation that keeps track of the iteration number. Thus we will denote the allele frequency estimates at iteration \(k\) by \((\hat{p}_A^{(k)}, \hat{p}_B^{(k)}, \hat{p}_O^{(k)})\).

We need some data. Let’s use the following data, collected in a study of peptic ulcers:

<table>
<thead>
<tr>
<th>Blood type</th>
<th>A</th>
<th>B</th>
<th>AB</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>186</td>
<td>38</td>
<td>13</td>
<td>284</td>
</tr>
</tbody>
</table>

Thus \(n = 521\). From these numbers, if we start the algorithm with \((\hat{p}_A^{(0)}, \hat{p}_B^{(0)}, \hat{p}_O^{(0)}) = (1/3, 1/3, 1/3)\), then \[\frac{\hat{\theta}^2}{\hat{\theta}_A^{(0)} + 2\hat{\theta}_A^{(0)}\hat{\theta}_O^{(0)}} = 1/3\], so we get

\[
\hat{n}^{(0)}_{A/A} = 186 \left[\frac{1}{3}\right] = 62, \quad \hat{n}^{(0)}_{A/O} = 186 \left[\frac{2}{3}\right] = 124, \\
\hat{n}^{(0)}_{B/B} = 38 \left[\frac{1}{3}\right] = 12.7, \quad \hat{n}^{(0)}_{B/O} = 38 \left[\frac{2}{3}\right] = 25.3.
\]

This yields in turn

\[
\hat{p}_A^{(1)} = \frac{(2n_{A/A}^{(0)} + n_{A/O}^{(0)} + n_{A/B})}{2n} = (2 \times 62 + 124 + 13)/(2 \times 521) = .250, \\
\hat{p}_B^{(1)} = \frac{(2n_{B/B}^{(0)} + n_{A/O}^{(0)} + n_{A/B})}{2n} = (2 \times \frac{38}{3} + \frac{76}{3} + 13)/(2 \times 521) = .061, \\
\hat{p}_O^{(1)} = \frac{(2n_{O/O}^{(0)} + n_{B/O}^{(0)} + n_{A/O})}{2n} = (2 \times 284 + 124 + \frac{76}{3})/(2 \times 521) = .688.
\]

We then repeat the process, using \((\hat{p}_A^{(1)}, \hat{p}_B^{(1)}, \hat{p}_O^{(1)})\) to find \((\hat{p}_{A/A}^{(1)}, \hat{n}_{A/O}^{(1)}, \hat{n}_{B/B}^{(1)}, \hat{n}_{B/O}^{(1)})\) and then \((\hat{p}_A^{(2)}, \hat{p}_B^{(2)}, \hat{p}_O^{(2)})\), etc. Here is a summary of what happens in 7 iterations:

<table>
<thead>
<tr>
<th>Iteration, (k)</th>
<th>(\hat{p}_A^{(k)})</th>
<th>(\hat{p}_B^{(k)})</th>
<th>(\hat{p}_O^{(k)})</th>
<th>Iteration, (k)</th>
<th>(\hat{p}_A^{(k)})</th>
<th>(\hat{p}_B^{(k)})</th>
<th>(\hat{p}_O^{(k)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.3333</td>
<td>0.3333</td>
<td>0.3333</td>
<td>4</td>
<td>0.2137</td>
<td>0.0501</td>
<td>0.7362</td>
</tr>
<tr>
<td>1</td>
<td>0.2505</td>
<td>0.0611</td>
<td>0.6884</td>
<td>5</td>
<td>0.2136</td>
<td>0.0501</td>
<td>0.7363</td>
</tr>
<tr>
<td>2</td>
<td>0.2185</td>
<td>0.0505</td>
<td>0.7311</td>
<td>6</td>
<td>0.2136</td>
<td>0.0501</td>
<td>0.7363</td>
</tr>
<tr>
<td>3</td>
<td>0.2142</td>
<td>0.0502</td>
<td>0.7357</td>
<td>7</td>
<td>0.2136</td>
<td>0.0501</td>
<td>0.7363</td>
</tr>
</tbody>
</table>
EM algorithms: Preliminaries

An EM algorithm is a method for obtaining maximum likelihood estimates iteratively in certain cases, namely, those in which we can identify some idea of complete data, which includes but is not limited to the data we actually observe (the observed data). The additional data that is part of the complete data but is unobserved is sometimes called missing data. Thus, EM algorithms are typically associated with problems involving missing data. We will show that the iterative scheme we presented for estimating allele frequencies at the ABO locus is one example of an EM algorithm.

Of course, if it were possible to find an MLE directly, without resorting to an EM algorithm, we would presumably do so. Thus, EM is useful in situations in which

1. The observed data likelihood is difficult to maximize or work with;
2. By defining the complete data in a clever way, we obtain a complete data likelihood that is easier to work with.

Consider the ABO example. The observed data are the numbers \((n_A, n_B, n_{AB}, n_O)\). The observed data loglikelihood underneath equation (8) on p. 10 is very difficult to work with. However, by defining the complete data in the “obvious” way, namely as the genotype counts instead of the phenotype counts, we have a likelihood that is much simpler to work with just as in the MN locus example. Thus, we define the complete data to be \((n_{AA}, n_{AO}, n_{AB}, n_{BO}, n_O)\) and obtain as the loglikelihood

\[
h(p_A, p_B) = K + 2n_{AA}\log p_A + n_{AO}\log[2p_A(1 - p_A - p_B)] + 2n_{AB}\log p_B + n_{BO}\log[2p_B(1 - p_A - p_B)] + n_{AB}\log(2p_Ap_B) + 2n_O\log(1 - p_A - p_B),
\]

where \(h(p_A, p_B)\) denotes the complete data loglikelihood and \(K\) is a constant not involving \(p_A\) or \(p_B\), namely the logarithm of \(\binom{n}{n_{AA}, n_{AO}, n_{AB}, n_{BO}, n_O}\).

Equation (9) might look pretty ugly, but in reality it’s almost identical in form to the MN locus loglikelihood (7), which we’ve already shown to be quite manageable. Thus, the observed data and the complete data as we’ve defined them seem to satisfy the 2 conditions above, so next we’ll see how to construct an EM algorithm.

Because any EM algorithm is iterative, we need to choose starting values of the parameters. On the previous page, we chose \((p_A^{(0)}, p_B^{(0)}, p_C^{(0)}) = (1/3, 1/3, 1/3)\). Of course, we’ve dropped \(p_O\) from the likelihood and replaced it by \(1 - p_A - p_B\), so really we only need \(p_A^{(0)}\) and \(p_B^{(0)}\). Often, if there is more than one parameter, we collect all of them into a single vector, say \(\theta\). Thus, we will take \(\theta^{(0)} = (1/3, 1/3)\). This is done solely for notational convenience, and there is no harm in switching back and forth between \(\theta\) and \((p_A, p_B)\).
EM algorithms: Mechanics

Once we’ve decided on how we’re going to define the complete data, we can start iterating. Suppose $k$ denotes the iteration number, so the current parameter vector is $\theta^{(k)}$. Notice that the complete data loglikelihood—equation (9) in our ABO example—involves the data. For a moment, let’s forget about the data being fixed and go back to thinking of them as random variables, as on p. 7 of the notes. The first step of each EM iteration is to take the expectation of the complete data loglikelihood (where the missing data are random). In order to take such an expectation, of course, we need to know what value of $\theta$ governs the random missing data—we take the expectation under the current parameter value $\theta^{(k)}$. Furthermore, the expectation we take will be conditional on the observed data. For brevity, we call this expectation, which will be a function of $\theta$, $Q_k(\theta)$. In symbols, we wish to define

$$Q_k(\theta) = E_{\theta^{(k)}}[h(\theta) \mid \text{observed data}],$ $	ext{sometimes written as } E \left[ h(\theta) \mid \text{observed data, } \theta^{(k)} \right],$$

where $h(\theta)$ denotes the complete data loglikelihood. Finding this expectation is called the E step. The M step will require us to maximize the result as a function of $\theta$. Notice that the expression in (10) contains both $\theta$ and $\theta^{(k)}$. They are different! The latter is the current parameter vector, whereas the former is simply a dummy variable.

This may have gotten a bit confusing, so let’s walk through it in the ABO example.

Consider only the first nonconstant term of the complete data loglikelihood (9), $2n_{A/A} \log p_A$. According to the description above, we wish to take the expectation of this quantity using the parameter values $\theta^{(k)}$ conditional on the observed data $(n_A, n_B, n_{AB}, n_O)$. In symbols, we want

$$E_{\theta^{(k)}}[2n_{A/A} \log p_A \mid (n_A, n_B, n_{AB}, n_O)].$$

For the time being, $\log p_A$ is merely a dummy variable; as far as the expectation above is concerned, it is a constant. Therefore, we may rewrite (11) as

$$2p_A E_{\theta^{(k)}}[n_{A/A} \mid (n_A, n_B, n_{AB}, n_O)].$$

(12)

To abbreviate some of the notation, denote the conditional expectation of $n_{A/A}$ in (12) by $\hat{n}_{A/A}^{(k)}$.

Conditional on $n_{A/A} + n_{A/O} = n_A$, the expectation of $n_{A/A}$ under the parameters $(p_A^{(k)}, p_B^{(k)})$ is

$$\hat{n}_{A/A}^{(k)} = n_A \left[ \frac{(p_A^{(k)})^2}{(p_A^{(k)})^2 + 2p_A^{(k)}(1 - p_A^{(k)} - p_B^{(k)})} \right].$$

(See why?) Similarly, we can find the expectations of $n_{A/O}$, $n_{B/B}$, and $n_{B/O}$ conditional on the observed data under the parameters $\theta^{(k)}$. The conditional expectations of $n_{AB}$ and $n_O$ conditional on $(n_A, n_B, n_{AB}, n_O)$ are obvious—they are just $n_{AB}$ and $n_O$, respectively.

As we did in equation (13), we’ll abbreviate notation by writing $\hat{n}_{A/O}^{(k)}$, $\hat{n}_{B/B}^{(k)}$, and $\hat{n}_{B/O}^{(k)}$ for the conditional expectations of $n_{A/O}$, $n_{B/B}$, and $n_{B/O}$. No such abbreviation is needed for $n_{AB}$ and $n_O$. 

13
EM algorithms: Mechanics (cont’d)

Putting everything together, the E step in the ABO example involves finding the conditional expectation seen in (10). It will be a function of \( \theta, \theta^{(k)} \), and the observed data \((n_A, n_B, n_{AB}, n_O)\):

\[
Q_k(\theta) = K + 2\hat{n}_{A/A}^{(k)} \log p_A + \hat{n}_{A/O}^{(k)} \log [2p_A(1 - p_A - p_B)] + 2\hat{n}_{B/B}^{(k)} \log p_B + n_{AB} \log (2p_A p_B) + 2n_O \log (1 - p_A - p_B). \tag{14}
\]

Notice that all of the \( p_A^{(k)} \) and \( p_B^{(k)} \) are hiding in terms like \( \hat{n}_{A/A}^{(k)} \) along with \( n_A \) and \( n_B \).

So much for the E step. The M step is to maximize the function (of \( \theta \)) obtained in the E step. The value of \( \theta \) that maximizes this function becomes the next iterate, \( \theta^{(k+1)} \). Then we repeat the whole process.

Maximizing the function in (14) may be done by taking partial derivatives with respect to \( p_A \) and \( p_B \), then setting each of them equal to zero and solving these two equations in two unknowns. Fortunately, this is not hard to do. After a bit of simplification, the partial derivatives are seen to be

\[
\frac{\partial Q_k(\theta)}{\partial p_A} = \frac{2\hat{n}_{A/A}^{(k)} + \hat{n}_{A/O}^{(k)} + n_{AB}}{p_A} - \frac{\hat{n}_{A/O}^{(k)} + \hat{n}_{B/O}^{(k)} + n_O}{1 - p_A - p_B} \tag{15}
\]

\[
\frac{\partial Q_k(\theta)}{\partial p_B} = \frac{2\hat{n}_{B/B}^{(k)} + \hat{n}_{B/O}^{(k)} + n_{AB}}{p_B} - \frac{\hat{n}_{A/O}^{(k)} + \hat{n}_{B/O}^{(k)} + n_O}{1 - p_A - p_B}.
\]

Setting each of these partials equal to zero, we get a pretty easy system to solve. Denoting the solution by \((p_A^{(k+1)}, p_B^{(k+1)})\), we have

\[
p_A^{(k+1)} = \frac{2\hat{n}_{A/A}^{(k)} + \hat{n}_{A/O}^{(k)} + n_{AB}}{2n} \quad \text{and} \quad p_B^{(k+1)} = \frac{2\hat{n}_{B/B}^{(k)} + \hat{n}_{B/O}^{(k)} + n_{AB}}{2n}.
\]

In other words, the EM algorithm is exactly the same as the iterative scheme described on p. 11!

To sum up, an EM algorithm consists of these steps:

1. Define complete data; obtain the complete data loglikelihood \( h(\theta) \).
2. Choose a value of \( \theta^{(0)} \) and set \( k = 0 \).
3. Let \( Q_k(\theta) = E_{\theta^{(k)}} [h(\theta) \mid \text{observed data}] \). This is the E step.
4. Maximize \( Q_k(\theta) \) and define \( \theta^{(k+1)} \) to be the maximizer. This is the M step.
5. Increment \( k \) and return to step 3.
EM algorithms: How they work

The important thing to realize about an EM algorithm is that the M step does not maximize the observed data loglikelihood; nonetheless, we may prove that at each iteration, the value of the observed data loglikelihood is increased. This is the so-called ascent property of an EM algorithm. Essentially, the ascent property asserts that

$$Q_k(\theta) > Q_k(\theta^{(k)}) \quad \text{implies} \quad l(\theta) > l(\theta^{(k)}).$$

(16)

Therefore, all we must do is to choose $\theta^{(k+1)}$ in such a way that $Q_k(\theta^{(k+1)}) > Q_k(\theta^{(k)})$—for example, by defining $\theta^{(k+1)}$ to be the maximizer of $Q_k(\theta)$—and we ensure that the observed data likelihood that we’re trying to maximize is driven uphill.

We will not prove the ascent property (16); however, it isn’t hard to understand what drives it. Consider the following diagram.

![Diagram](image)

Suppose the horizontal axis is the $\theta$-axis for some problem and we’re trying to maximize the observed data loglikelihood as a function of $\theta$, represented here by the upper curve. Furthermore, suppose that this loglikelihood is not easy to maximize using direct (paper-and-pencil) methods. Take $\theta^{(0)} = 0$ to be the starting value for an iterative scheme.

The lower curve represents the $Q_0(\theta)$ function (though in reality, $Q_0(\theta)$ would be shifted up or down by a constant so that the two curves don’t actually touch at $\theta = 0$; this translation is unimportant in maximizing so we depict the two curves as tangent to one another). Note that because the lower curve touches the upper curve at $\theta^{(0)}$ and lies completely below the upper curve, it must be true that any $\theta$ that increases the value of the lower curve must also increase the value of the upper curve. This is exactly the ascent property!

The “magic” of an EM algorithm is that the E step constructs a function $Q_k(\theta)$ that (up to a constant) is tangent to and lies completely below the observed data loglikelihood. Thus, when we maximize $Q_k(\theta)$ during the M step, we guarantee an increase in the observed data loglikelihood as desired.
Estimation of inbreeding coefficient

Consider the maximum likelihood estimation of parameters in the following situation.

In an inbred population, the inbreeding coefficient \( f \) is the probability that two genes of a random person at some locus are both copies of the same ancestral gene. Suppose there are \( m \) codominant alleles at the locus, \( A_1, \ldots, A_m \), and \( p_i \) is the frequency of allele \( A_i \). We observe \( n_{ij} \) people of genotype \( A_i/A_j \) in a random sample. We wish to estimate the parameters \( f, p_1, \ldots, p_m \).

If we assume that the population is in equilibrium, we may use maximum likelihood to obtain the results. We call any genotype of the form \( A_i/A_i \) **homozygous**, whereas \( A_i/A_j \) is a **heterozygous** genotype if \( i \neq j \). To determine the frequency of the \( A_i/A_i \) genotype, condition on whether the two \( A_i \) alleles are copies of the same ancestral gene:

\[
P(A_i/A_i) = P(A_i/A_i | \text{copies}) P(\text{copies}) + P(A_i/A_i | \text{not copies}) P(\text{not copies})
\]

\[
= p_i f + p_i^2 (1 - f).
\]

Similarly, because for \( i \neq j \) we have \( P(A_i/A_j | \text{copies}) = 0 \), the frequency of the \( A_i/A_j \) genotype is \( 2(1 - f)p_ip_j \).

Suppose \( m = 3 \). Then the data are the numbers \( \{n_{11}, n_{12}, n_{13}, n_{22}, n_{23}, n_{33}\} \). Using the results derived in the preceding paragraph, we write the loglikelihood for the observed data:

\[
l(f, p_1, p_2, p_3) = C + n_{11} \log[f p_1 + (1 - f) p_1^2] + n_{12} \log[2(1 - f)p_1p_2]
\]

\[
+ n_{13} \log[2(1 - f)p_1p_3] + n_{22} \log[f p_2 + (1 - f) p_2^2]
\]

\[
+ n_{23} \log[2(1 - f)p_2p_3] + n_{33} \log[f p_3 + (1 - f) p_3^2].
\]

The objective of maximum likelihood estimation, of course, is to find \((f, \hat{p}_1, \hat{p}_2, \hat{p}_3)\) to maximize the function in (17) subject to the constraint \( p_1 + p_2 + p_3 = 1 \). We may eliminate the constraint by replacing \( p_3 \) by \( 1 - p_1 - p_2 \) throughout the loglikelihood (17). This will be our approach in what follows.

Aside: We could use an EM algorithm here. Suppose the complete data tell us precisely how many of each genotype are copies of the same ancestral gene. In other words, suppose the complete data are \( \{n_{11}^C, n_{12}^C, n_{13}^C, n_{22}^C, n_{23}^C, n_{33}^C\} \). Then the complete-data loglikelihood would be

\[
h(f, p_1, p_2) = C + n_{11}^C \log[f p_1 + (1 - f) p_1^2] + n_{12}^C \log[2(1 - f)p_1p_2]
\]

\[
+ n_{13}^C \log[2(1 - f)p_1(p_1 - p_2)] + n_{22}^C \log[f p_2 + (1 - f) p_2^2]
\]

\[
+ n_{23}^C \log[2(1 - f)p_2(p_1 - p_2)] + n_{33}^C \log[f (1 - p_1 - p_2)] + n_{33}^C \log[(1 - f)(1 - p_1 - p_2)^2].
\]

Can you find the conditional expectation, given \((f^{(k)}, \hat{p}_1^{(k)}, \hat{p}_2^{(k)}) \) and the observed data, of (18) and create an EM algorithm?
Other maximization methods: Newton’s method

We’ll get back to the inbreeding coefficient example soon. First, I’ll introduce the maximization method known as Newton’s method.

The goal is to find a point where the loglikelihood function is flat (a maximum is one such point). Mathematically, saying that the loglikelihood is flat is the same as saying that the first derivative of the loglikelihood is zero. So we’re looking for a point where the first derivative of the loglikelihood (called the score) is zero.

Newton’s method, sometimes referred to as the Newton-Raphson method, is an iterative procedure with a nice geometric intuition. If there is only a single one-dimensional parameter, say \( \theta \), we update \( \theta^{(k)} \) at the \( k + 1 \)th iteration as follows:

\[
\theta^{(k+1)} = \theta^{(k)} - \frac{f(\theta^{(k)})}{f''(\theta^{(k)})}.
\]

If we define the score function \( S(\theta) \) to be the first derivative of the loglikelihood and the observed information \( I_o(\theta) \) to be the negative second derivative, Newton’s method becomes

\[
\theta^{(k+1)} = \theta^{(k)} + [I_o(\theta^{(k)})]^{-1}S(\theta^{(k)}), \tag{19}
\]

The situation becomes slightly more complicated if there is more than one parameter. Then \( \theta \) becomes a vector, which means that the score is a vector of partial first derivatives of the loglikelihood and the observed information is a matrix of partial second derivatives of the loglikelihood. However, Newton’s method may still be expressed by equation (19).

Consider the inbreeding coefficient example. The parameter vector is \( \theta = (f, p_1, p_2)^t \). Note that I’ll follow the convention that vectors are assumed to be column vectors unless otherwise specified; that’s why, for example, we put a transpose indicator in \( \theta = (f, p_1, p_2)^t \). The score vector and observed information matrix are defined as

\[
S(\theta) = \begin{bmatrix}
\frac{\partial f}{\partial f} \\
\frac{\partial f}{\partial p_1} \\
\frac{\partial f}{\partial p_2}
\end{bmatrix}
\quad \text{and} \quad
I_o(\theta) = -\begin{bmatrix}
\frac{\partial^2 f}{\partial f^2} & \frac{\partial^2 f}{\partial f \partial p_1} & \frac{\partial^2 f}{\partial f \partial p_2} \\
\frac{\partial^2 f}{\partial f \partial p_1} & \frac{\partial^2 f}{\partial p_1^2} & \frac{\partial^2 f}{\partial f \partial p_2} \\
\frac{\partial^2 f}{\partial f \partial p_2} & \frac{\partial^2 f}{\partial f \partial p_2} & \frac{\partial^2 f}{\partial p_2^2}
\end{bmatrix}.
\]
Newton’s method cont’d

To see how Newton’s method works in the inbreeding example, let’s consider the following data on haptoglobin genotypes from 1,948 people in northeast Brazil. The haptoglobin locus has three codominant alleles, $G_1$, $G_2$, and $G_3$. The slight excess of homozygotes in these data suggests inbreeding.

<table>
<thead>
<tr>
<th>Genotype</th>
<th>Observed Number</th>
<th>Expected number with no inbreeding</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1/G_1$</td>
<td>108</td>
<td>90.77</td>
</tr>
<tr>
<td>$G_1/G_2$</td>
<td>196</td>
<td>214.78</td>
</tr>
<tr>
<td>$G_1/G_3$</td>
<td>429</td>
<td>444.68</td>
</tr>
<tr>
<td>$G_2/G_2$</td>
<td>143</td>
<td>127.06</td>
</tr>
<tr>
<td>$G_2/G_3$</td>
<td>513</td>
<td>526.10</td>
</tr>
<tr>
<td>$G_3/G_3$</td>
<td>559</td>
<td>544.61</td>
</tr>
</tbody>
</table>

Rewrite the loglikelihood (17) as a function of $\theta = (f, p_1, p_2)$ and insert the known values of the data:

$$l(\theta) = C + 108 \log[f p_1 + (1 - f)p_1^2] + 196 \log[2(1 - f)p_1 p_2] + 429 \log[2(1 - f)p_1 (1 - p_1 - p_2)] + 143 \log[p_2 + (1 - f)p_2^2] + 513 \log[2(1 - f)p_2 (1 - p_1 - p_2)] + 559 \log[f (1 - p_1 - p_2) + (1 - f)(1 - p_1 - p_2)^2]$$

To implement Newton’s method, we must take the derivates of (17) with respect to $f$, $p_1$, and $p_2$ to form the score vector, then take all the second derivatives to form the observed information. As you can see, this will be a mess! (By the way, computational complexity for many problems is one of the drawbacks of Newton’s method.) Rather than write out the derivative for general $(f, p_1, p_2)$ here, I’ll simply give the starting values of the score and observed information given the starting point $(f^{(0)}, p_1^{(0)}, p_2^{(0)}) = (0.05, 0.25, 0.25)$:

$$S(\theta^{(0)}) = \begin{bmatrix} -10.73 \\ -724.65 \\ -133.00 \end{bmatrix} \quad \text{and} \quad I_o(\theta^{(0)}) = \begin{bmatrix} 3476.1 & -721.5 & -298.1 \\ -721.5 & 20741.6 & 7834.4 \\ -298.1 & 7834.4 & 23027.8 \end{bmatrix}$$

Here is the progress of Newton’s method:

<table>
<thead>
<tr>
<th>Iteration, $k$</th>
<th>$\theta^{(k)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(.0500,.2500,.2500)</td>
</tr>
<tr>
<td>1</td>
<td>(.0396,.2121,.2570)</td>
</tr>
<tr>
<td>2</td>
<td>(.0431,.2157,.2554)</td>
</tr>
<tr>
<td>3</td>
<td>(.0431,.2157,.2554)</td>
</tr>
</tbody>
</table>

One of the hallmarks of Newton’s method is that once it gets close to the final answer, convergence occurs very quickly. On the other hand, unlike the EM algorithm, Newton’s method is not guaranteed to increase the loglikelihood at each iteration. In fact, Newton’s method sometimes exhibits very bad behavior and fails to converge at all.
**Scoring and covariance estimates**

The method of **scoring** is nearly identical to Newton’s method except that the observed information $I_o(\theta^{(k)})$ is replaced by $I_e(\theta^{(k)})$:

$$
\theta^{(k+1)} = \theta^{(k)} + [I_e(\theta^{(k)})]^{-1} S(\theta^{(k)}),
$$

where $I_e(\theta)$ is the **expected information**, defined as $-E_{\theta}[d^2 l(\theta)]$. The notation $d^2 l(\theta)$ stands for the matrix of second partial derivatives of $l(\theta)$, sometimes called the **Hessian** matrix. In order to take the expectation, we must treat the data that occurs in the expression $l(\theta)$ as random, instead of fixed, with its random behavior governed by the parameter $\theta$.

The behavior of scoring is usually not qualitatively very different from that of Newton’s method. In fact, it sometimes happens that Newton’s method and scoring are identical because the observed information matrix is the same as the expected information.

Newton’s method and scoring share one big advantage over EM algorithms—the inverse of the observed or expected information matrix, evaluated at the MLE $\hat{\theta}$, is an estimate of the covariance matrix of the MLE. Therefore, it is possible to quantify the precision of the maximum likelihood estimate if we have one of these inverses at hand. It is certainly possible to run an EM algorithm until it converges, then evaluate the inverse of the information matrix to obtain a covariance estimate. However, this involves extra work, whereas the Newton and scoring schemes already use this inverse so it is automatically computed. As an example, the inverse of the Hessian matrix after convergence of the Newton method in the inbreeding example is

$$
[I_o(\theta^{(3)})]^{-1} = \begin{bmatrix}
271.802 & .970 & 1.228 \\
.970 & 45.244 & -14.714 \\
1.228 & -14.714 & 50.857
\end{bmatrix} \times 10^{-6}
$$

We may estimate the standard errors of $\hat{f}$, $\hat{p}_1$, and $\hat{p}_2$ by taking the square roots of the diagonal elements of $[I_o(\theta^{(3)})]^{-1}$, which gives .0165, .0067, and .0071, respectively. Furthermore, the off-diagonal covariances may be used to approximate the correlations among the parameter estimates, as follows:

$$
\text{Corr}(\hat{f}, \hat{p}_1) \approx \frac{.97}{\sqrt{271.8(45.2)}} = .0088, \text{ etc.}
$$

In summary, Newton’s method and scoring have advantages and disadvantages relative to EM algorithms. Advantages are fast convergence and automatic covariance estimates. Disadvantages are computational complexity and the potential for unpredictable behavior.
Information in the multinomial model

Typically, when the word “information” is used in statistics, it refers to the expected information \( I_e(\theta) \) and not the observed information \( I_o(\theta) \). (Please note that the \( I_e \) and \( I_o \) notation is my own; typically, \( I(\theta) \) with no subscript denotes the expected information.)

Because it will be used later on, we now derive the expected information for a multinomial model. A multinomial model is one in which each individual in a random sample of size \( n \) can fall into one of \( m \) categories, where category \( i \) occurs with probability \( \pi_i \) for each individual, independent of all other individuals \((i = 1, 2, \ldots, m)\). We have already seen this model several times; for example, for a locus with three codominant alleles, there are 6 possible phenotypes. So if we’re modeling the numbers of the various phenotypes observed in a random sample, we have \( m = 6 \). Under Hardy-Weinberg equilibrium, we have \( \pi_1 = p_A^2 \), \( \pi_2 = 2p_Ap_B \), \( \pi_3 = 2p_Ap_C \), \( \pi_4 = p_B^2 \), \( \pi_5 = 2p_Bp_C \), and \( \pi_6 = p_C^2 \).

If we let \( x_i \) denote the number of individuals in the sample observed to fall in category \( i \), we have seen repeatedly that the loglikelihood looks like

\[
\log \left( \prod_{i=1}^{m} x_i \right) + \log \pi_1 + \log \pi_2 + \cdots + \log \pi_m, \quad \text{or} \quad \sum_{i=1}^{m} x_i \log \pi_i. \tag{22}
\]

Note that each \( \pi_i \) is in general a function of the parameters of interest; for example, we could have \( \pi_1 = p_A^2 \), \( \pi_2 = 2p_Ap_B \), etc., as mentioned above.

To obtain \( I_e(\theta) \), where as usual \( \theta \) denotes the vector of parameters of interest, we first take minus the matrix of partial second derivatives. The \( j, k \) entry of this matrix is obtained by differentiating expression (22) with respect to \( \theta_j \) and then with respect to \( \theta_k \), then multiplying by \(-1\), which gives

\[
I_e(\theta)_{jk} = \sum_{i=1}^{m} x_i \frac{\partial^2 \pi_i}{\partial \theta_j \partial \theta_k} - \sum_{i=1}^{m} \frac{x_i}{\pi_i} \frac{\partial^2 \pi_i}{\partial \theta_j \partial \theta_k}.
\]

The expected information is obtained by treating the data \( x_i \) as random and taking the expectation of (23) under the assumption that the random behavior of \( X_i \) is governed by the parameter vector \( \theta \). For a multinomial model, we have \( \mathbb{E}_{\theta} X_i = n \pi_i \)—recall that \( \pi_i \) is a function of \( \theta \)—and thus we obtain

\[
I_e(\theta)_{jk} = n \sum_{i=1}^{m} \frac{1}{\pi_i} \frac{\partial \pi_i}{\partial \theta_j} \frac{\partial \pi_i}{\partial \theta_k} - n \sum_{i=1}^{m} \frac{\partial^2 \pi_i}{\partial \theta_j \partial \theta_k} = n \sum_{i=1}^{m} \frac{1}{\pi_i} \frac{\partial \pi_i}{\partial \theta_j} \frac{\partial \pi_i}{\partial \theta_k}. \tag{24}
\]

The second equality above follows because \( \sum_{i=1}^{m} \pi_i = 1 \), which of course has derivative zero, and because we may interchange the order of differentiation and summation.

If \( \theta \) is a scalar instead of a vector, then of course \( I_e(\theta) \) is also a scalar (instead of a matrix), and the subscripts \( j \) and \( k \) become irrelevant—they are both one—and may be dropped. We’ll see an example of this in a couple pages.
Meiosis and Mendel’s second law

As we’ve already stated, an individual receives two copies of each chromosome, one from each parent. We now discuss in slightly more detail how these two copies are obtained.

Take human reproduction as an example, recalling that humans have 23 pairs of chromosomes. At mating, each parent contributes one gamete (egg or sperm cell) that contains one of each of the 23 pairs of chromosomes. The formation of the gametes in the parents takes place in a process called meiosis.

In meiosis, a cell (called a meiocyte) with the full complement of 23 pairs of chromosomes undergoes the following steps: First, each of the 46 chromosomes is duplicated, forming the familiar X shape consisting of two identical strands of DNA, called sister chromatids. Later, at a stage we’ll call prophase I for future reference, the 46 chromosomes line up in pairs. These pairs are then pulled apart to form two separate daughter cells, each with only 23 chromosomes—but each of these chromosomes has been copied, so it still consists of two chromatids joined together (like an X). Finally, each of these 23 chromosomes splits into its two strands of DNA, forming a total of four gametes. Each of these four meiotic products contains, therefore, 23 chromosomes—one representative from each of the original 23 pairs of the parent. This is not to say that each of the 23 chromatids in each gamete is the same as one of the 2 corresponding chromatids in the parent, since recombination could occur—more on that later.

From each pair at prophase I, the chromosome that goes into each daughter cell is equally likely to be either of the pair, independent of each of the other pairs. (The two chromosomes that form a pair are called homologues, or homologous chromosomes.) If two genes reside on different chromosomes, therefore, we see the truth of Mendel’s second law: During gamete formation, the segregation of alleles of one gene is independent of the segregation of alleles of another gene.

As a simple example, consider two loci on different chromosomes. Suppose each locus has two alleles: At one locus, A is dominant over a, while at the second locus, B is dominant over b. If (for the sake of simplicity) we assume that \( p_A = p_a = p_B = p_b = 1/2 \), what are the frequencies of the four possible phenotypes AB, Ab, aB, and ab under Hardy-Weinberg equilibrium? The answers are given in the table below:

<table>
<thead>
<tr>
<th>Phenytype</th>
<th>Possible ordered genotypes</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB</td>
<td>AB/AB, AB/Ab, AB/aB, AB/ab, AB/Ab, aB/Ab, aB/AB</td>
<td>9/16</td>
</tr>
<tr>
<td>Ab</td>
<td>Ab/Ab, Ab/ab, ab/Ab</td>
<td>3/16</td>
</tr>
<tr>
<td>aB</td>
<td>aB/Ab, aB/ab, ab/aB</td>
<td>3/16</td>
</tr>
<tr>
<td>ab</td>
<td>ab/ab</td>
<td>1/16</td>
</tr>
</tbody>
</table>
Violations of Mendel’s second law

Suppose two genes are found on the same chromosome. Then, according to the simplistic explanation of meiosis on the preceding page, Mendel’s second law is completely false: The alleles do not segregate at all independently at meiosis. For example, if a parent has genotype AB/ab for two diallelic loci on the same chromosome, then the simplistic explanation of meiosis suggests that each gamete formed must contain one of the two homologues of this chromosome. Thus, the child could inherit either AB (i.e., the A and B alleles) or ab from that parent, but not Ab or aB.

However, the simplistic description missed an important facet of meiosis: The possibility of crossovers. In prophase I, it is possible for segments of nonsister chromatids to trade places with one another. In the previous example of an AB/ab parent, the two loci are separated by some physical distance on the chromosome. If it happens that a segment of a chromatid containing A but not B is exchanged with the corresponding segment of one of its nonsister chromatids (i.e., one containing a but not b), then gametes containing the Ab and aB combinations will be formed. Such an event is called a recombination.

We define the recombination fraction \( \rho \) between two loci to be the probability of observing a recombination between them. Note that if a parent were homozygous at each locus, a recombination would be undetectable; nonetheless, if the two alleles that wind up in a gamete originate on nonsister chromatids, we still consider a recombination to have occurred. Since intuitively, loci are more likely to recombine the further apart they lie physically on the chromosome, the size of the recombination fraction is a reflection of the physical distance separating the two loci: The larger the physical distance, the greater the recombination fraction. Since Mendel’s second law states in effect that \( \rho = 1/2 \), we have seen that \( \rho = 1/2 \) for genes on different nonhomologous chromosomes.

The residing of genes on the same chromosome pair is called linkage. In an ideal population (see p. 4), Mendel’s second law predicts that the frequencies of AB, Ab, aB, and ab gametes for two diallelic loci should be \( p_Ap_B, p_Ap_b, p_ap_B, \) and \( p_ap_b \) respectively (these probabilities must sum to one, a fact that follows from \( p_A + p_a = p_B + p_b = 1 \)). A population in which the frequencies of the types of gametes inherited by individuals correspond to those predicted by Mendel’s second law is said to be in linkage equilibrium.

Assuming that all individuals exhibit a recombination frequency \( \rho \) between the two alleles in question, then if \( f_n \) denotes the frequency of the AB gametes produced at generation \( n \), we have for the ideal population

\[
\begin{align*}
f_n & = (1-\rho)f_{n-1} + \rho p_Ap_B, \quad \text{or} \quad f_n - p_Ap_B = (1-\rho)(f_{n-1} - p_Ap_B) \\
& = (1-\rho)^n(f_0 - p_Ap_B).
\end{align*}
\]

Thus, linkage equilibrium is approached at the geometric rate \( 1 - \rho \). Note that males and females typically have different recombination frequencies, but the above argument can be easily modified using separate values \( \rho_f \) and \( \rho_m \).
Design of linkage experiments

Suppose that as before, we have two diallelic loci, but this time assume they are codominant with alleles \( A_1 \) and \( A_2 \) at the first locus and \( B_1 \) and \( B_2 \) at the second. We wish to estimate the recombination fraction \( \rho \) between these loci.

Of course, one may not ethically carry out breeding experiments on humans in which certain types of individuals are mated with certain other types at the behest of the researcher; for laboratory animals, however, it is possible to design such experiments.

We now compare two possible experimental designs on the basis of the (expected) information produced by each. The intuitively appealing notion that more information about the parameter is better is justified as follows: Since we’ve seen that the variance of an MLE may be approximated by the inverse of the information, and since smaller variances imply more precise estimates, it stands to reason that the greater the information, the more precise the estimate. In fact, information is proportional to sample size, so if Experiment 1 gives numerically twice the information of Experiment 2, we may interpret this by saying that Experiment 2 requires a sample size roughly twice as large as Experiment 1 to achieve the same precision of estimates.

These are the two experiments:

1. Breed \( A_1 B_1 / A_2 B_2 \) animals with \( A_1 B_1 / A_1 B_1 \) animals.

2. Breed \( A_1 B_1 / A_2 B_2 \) animals with \( A_1 B_1 / A_2 B_2 \) animals.

By \( A_1 B_1 / A_2 B_2 \) animals, we mean animals that received an \( A_1 B_1 \) gamete from one parent and an \( A_2 B_2 \) gamete from the other parent. The genetic contribution of each gamete is called a **haplotype**, so we may also say that an \( A_1 B_1 / A_2 B_2 \) animal has \( A_1 B_1 \) and \( A_2 B_2 \) haplotypes. Note that \( A_1 B_1 / A_2 B_2 \) animals must be heterozygous at these two loci, but not conversely; thus, knowing the haplotypes of a given individual is more than knowing the individual’s genotype at each locus.

The following table summarizes the possible phenotypes of a child from each experiment:

<table>
<thead>
<tr>
<th>( i )</th>
<th>Locus A phenotype</th>
<th>Locus B phenotype</th>
<th>( \pi_i(\rho) ) for Experiment 1</th>
<th>( \pi_i(\rho) ) for Experiment 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( A_1 / A_1 )</td>
<td>( B_1 / B_1 )</td>
<td>( \frac{1}{2} ) ( (1 - \rho) )</td>
<td>( \frac{1}{2} ) ( (1 - \rho)^2 )</td>
</tr>
<tr>
<td>2</td>
<td>( A_1 / A_2 )</td>
<td>( B_1 / B_1 )</td>
<td>( \frac{1}{2} ) ( \rho )</td>
<td>( \frac{1}{2} ) ( \rho (1 - \rho) )</td>
</tr>
<tr>
<td>3</td>
<td>( A_2 / A_2 )</td>
<td>( B_1 / B_1 )</td>
<td>0</td>
<td>( \frac{1}{4} ) ( \rho^2 )</td>
</tr>
<tr>
<td>4</td>
<td>( A_1 / A_1 )</td>
<td>( B_1 / B_2 )</td>
<td>( \frac{1}{2} ) ( \rho )</td>
<td>( \frac{1}{2} ) ( \rho (1 - \rho) )</td>
</tr>
<tr>
<td>5</td>
<td>( A_1 / A_2 )</td>
<td>( B_1 / B_2 )</td>
<td>( \frac{1}{2} ) ( (1 - \rho) )</td>
<td>( \frac{1}{2} \left[ \rho^2 + (1 - \rho)^2 \right] )</td>
</tr>
<tr>
<td>6</td>
<td>( A_2 / A_2 )</td>
<td>( B_1 / B_2 )</td>
<td>0</td>
<td>( \frac{1}{2} \rho (1 - \rho) )</td>
</tr>
<tr>
<td>7</td>
<td>( A_1 / A_1 )</td>
<td>( B_2 / B_2 )</td>
<td>0</td>
<td>( \frac{1}{4} \rho^2 )</td>
</tr>
<tr>
<td>8</td>
<td>( A_1 / A_2 )</td>
<td>( B_2 / B_2 )</td>
<td>0</td>
<td>( \frac{1}{2} \rho (1 - \rho) )</td>
</tr>
<tr>
<td>9</td>
<td>( A_2 / A_2 )</td>
<td>( B_2 / B_2 )</td>
<td>0</td>
<td>( \frac{1}{4} ) ( (1 - \rho)^2 )</td>
</tr>
</tbody>
</table>
Design of linkage experiments cont’d

Simplifying equation (24) for the case in which $\theta$ is merely the scalar $\rho$, (hence $j = k = 1$), we obtain the formula

$$I_e(\rho) = n \sum_{i=1}^{m} \frac{1}{\pi_i(\rho)} [\pi'(\rho)]^2 \quad (25)$$

for the expected information. Setting $n = 1$ gives the information for a single individual. In the case of Experiment 1, we obtain

$$I_e(\rho) = \frac{1}{\rho} + \frac{1}{1 - \rho}. \quad (26)$$

For Experiment 2, we obtain

$$I_e(\rho) = 2 + \frac{2(1 - 2\rho)^2}{\rho(1 - \rho)} + 2 + \frac{2(1 - 2\rho)^2}{2\rho^2 - 2\rho + 1}. \quad (27)$$

We can see graphically how these two functions compare:

![Graph showing the comparison of information and recombination fraction for Experiments 1 and 2.](image)

We see that Experiment 2 is always at least as informative as Experiment 1, with its advantage growing as $\rho$ gets closer to 0.
Linkage Phase

Recall that the recombination fraction $\rho$ for two loci denotes the probability of a recombination event between those two loci. For loci on different chromosomes, $\rho = 1/2$. For loci on the same chromosome, we say that the loci are linked and $\rho \leq 1/2$.

Consider two diallelic loci, one with alleles A and a, the other with alleles B and b. Suppose an individual is heterozygous at each locus; that is, her genotypes are A/a and B/b. From this fact alone, it is impossible to tell which alleles came from which parent, or even the makeup of the two gametes from the parents: The gametes could have been Ab and aB, or they could have been AB and ab. The alleles that comprise one of the gametes (e.g., Ab) are called a haplotype, and alleles that are part of the same haplotype are said to be in phase. Thus, in our simple example, we cannot construct the haplotypes from the given information, which means we lack knowledge of phase. Notice, however, that any other combination of genotypes for these two loci would have made phase determinable, since only the doubly heterozygous genotype leads to any ambiguity.

Notice, for example, that in Experiment 1 on p. 23, it is possible to determine phase for all of the offspring of the $A_1B_1/A_2B_2 \times A_1B_1/A_1B_1$ mating since one of the haplotypes of each offspring is guaranteed to be $A_1B_1$. As a final example, consider the locus AK1 (adenylate kinase 1) in the vicinity of the ABO locus on chromosome 9. This locus has two codominant alleles, $A_1$ and $A_2$. Consider the following pedigree, with phenotypes for each individual (males are squares; females are circles).

From the pedigree, it is possible to determine the haplotypes of each of the two children. (The above pedigree is somewhat atypical in that it has complete information and because phase may be unambiguously determined for children.) Note that if individual 4 were $A_1/A_2$ instead of $A_2/A_2$, we could not determine phase for individual 5.

Phase is important in linkage studies because it determines the gamete frequencies an individual produces. For example, an AB/ab individual produces Ab gametes with probability $\rho/2$, whereas an Ab/aB individual produces Ab gametes with probability $(1 - \rho)/2$.  

25
Recombination models

As remarked earlier, small values of the recombination fraction $\rho$ correspond to loci that are close together on the chromosome. We now seek some way to relate $\rho$ to distances on a chromosome. Since a crossover can take place anywhere along a chromosome, it will occur rarely between two loci that are very close to each other. The further apart two loci are, the more likely it is that a crossover will occur between them. In general, an odd number of crossovers results in a recombination, while an even number does not.

Recall that in meiosis, each (autosomal) chromosome duplicates itself then lines up next to its homologue. At this stage, there are four separate chromatids together in a bundle, two copies of the maternal chromatid and two copies of the paternal chromatid. Crossing over occurs at points called **chiasmata** along the bundle. At each chiasma, one of the paternal chromatids and one of the maternal chromatids are cut then rejoined with each other, so that each of the resulting chromatids consists of paternal material on one side of the chiasma and maternal material on the other side. The selection of which paternal chromatid and which maternal chromatid will undergo this exchange for each chiasma is assumed to be made at random; there is substantial empirical evidence that this assumption is reasonable. Furthermore, the choices of chromatids appear more or less independent from chiasma to chiasma; that is, there is no **chromatid interference**.

Let $N_{AB}$ denote the number of chiasmata occurring on the bundle between two loci located at points $A$ and $B$ along the bundle. Of course, $N_{AB}$ is a random variable that takes nonnegative integer values. We now derive a result, called Mather’s formula, that proves among other things that the recombination fraction between the two loci is bounded above by 1/2 under these assumptions.

Let $q_n$ denote the probability that a randomly chosen chromatid is recombinant between locus $A$ and locus $B$ conditional on the fact that there are $n$ chiasmata on its chromatid bundle between $A$ and $B$. For $n > 0$, we may find a recurrence relation for $q_n$ by using the fact that the chromatid is recombinant after $n$ crossovers if it is recombinant after $n - 1$ crossovers and does not take place in the $n$th crossover or if it is nonrecombinant after $n - 1$ crossovers and does take place in the $n$th crossover. Thus, we obtain

$$q_n = \frac{1}{2}q_{n-1} + \frac{1}{2}(1 - q_{n-1}) = \frac{1}{2}$$

for $n > 0$. Clearly, $q_0 = 0$. We conclude that

$$\rho_{AB} = \sum_{n=1}^{\infty} r_n P(N_{AB} = n) = \frac{1}{2} P(N_{AB} > 0)$$  \hspace{1cm} (28)$$

Equation (28) is called Mather’s formula, and clearly it implies that $\rho_{AB} \leq 1/2$. The only assumption we made was that there is no chromatid interference, which appears to be a reasonable assumption.
Map distance

**Map distance** between two loci $A$ and $B$ is defined as the expected number of crossovers between $A$ and $B$ per chromatid produced at meiosis. If we denote this map distance by $x_{AB}$, we therefore have $x_{AB} = E(N_{AB})/2$. Suppose we consider several loci on a chromosome, as follows:

$$
\begin{array}{cccc}
A & B & C & D \\
\end{array}
$$

Then map distance is a nice way to measure distance because it is additive:

$$x_{AD} = x_{AB} + x_{BC} + x_{CD}. \quad (29)$$

This is certainly not true of the recombination fraction, which is bounded by $1/2$. The unit of length for map distance is the Morgan (M). One Morgan is one expected crossover (per gamete). Since one Morgan turns out to correspond to a rather large distance, a more convenient unit of measure is the centiMorgan (cM), 1/100 of a Morgan.

Map distance does not necessarily reflect physical distance—the number of base pairs between loci—due to the fact that the frequency of crossovers is not constant throughout the genome. Nonetheless, experimental results suggest a rough proportionality between map distance and physical chromosome length. On the other hand, there are certain “recombination hot spots”; for example, in one region of the short arm of chromosome 11 that controls insulin, analyses indicate that recombination in that region occurs over 30 times more frequently than expected.

Note that it is possible to measure the length of the entire human genome in Morgans. A figure of 33 Morgans is often given for the total autosomal length, although this is actually a sex-averaged figure: The length is roughly 2750 cM for males and 3850 cM for females.

Returning to equation (29) for a moment, we notice that it is implied by the linearity of the expectation operator:

$$
\frac{1}{2} E(N_{AD}) = \frac{1}{2} E(N_{AB} + N_{BC} + N_{CD}) = \frac{1}{2} E(N_{AB}) + \frac{1}{2} E(N_{BC}) + \frac{1}{2} E(N_{CD}). \quad (30)
$$

Equation (30) is true for any random variables $N_{AB}, N_{BC}, N_{CD}$, whether or not they are independent. If these chiasma counts are not independent, this phenomenon is called **chiasma interference** or sometimes simply **interference**.
Independence versus interference

\[
\begin{array}{c|c|c|c}
A & B & C & D \\
\hline
\end{array}
\]

Let \( AC \) signify the event that an odd number of crossovers occurs between \( A \) and \( C \), and \( \overline{AC} \) its complement. Then \( AC \) may be expressed as the union of the disjoint events \( AB \cap BC \) and \( \overline{AB} \cap BC \). Thus,

\[
\rho_{AC} = P(AC) = P(AB \cap BC) + P(\overline{AB} \cap BC).
\]

Therefore, if \( AB \) and \( BC \) are independent, we obtain Trow’s formula:

\[
\rho_{AC} = \rho_{AB}(1 - \rho_{BC}) + (1 - \rho_{AB})\rho_{BC} = \rho_{AB} + \rho_{BC} - 2\rho_{AB}\rho_{BC}.
\]

Independence is another way of saying that there is no chiasma interference. This assumption turns out to be inaccurate; in most cases, the occurrence of one crossover tends to inhibit the formation of other crossovers in its neighborhood (positive interference).

Looking at it another way, we see that under independence (lack of interference), \( P(AB \cap BC) = \rho_{AB}\rho_{BC} \). More generally, if we write

\[
P(AB \cap BC) = (1 - i)\rho_{AB}\rho_{BC},
\]

then we can think of \( i \) as a coefficient of interference. For \( i = 0 \), there is no interference. For \( i > 0 \), the occurrence of a recombination event between \( A \) and \( B \) decreases the probability that there will also be a recombination between \( B \) and \( C \).

We wish to relate recombination fraction \( \rho \) to map distance \( x \) by way of a map function. Many such functions have been proposed, and we will simply scratch the surface here. Consider first the simple map function \( x = \rho \) originally proposed by Morgan. This may be viewed as the complete interference \((i=1)\) model, since it allows for no more than one crossover per chromosome.

Returning to the independence or no interference case, we might propose that the chiasma are produced by a Poisson process. This means, for example, that \( N_{AB}, N_{BC}, \) and \( N_{CD} \) are independent Poisson random variables. Let’s consider \( N_{AB} \). We have already defined \( x_{AB} \) to be half of \( E(N_{AB}) \), so

\[
P(N_{AB} = r) = P(r \ \text{chiasmata between} \ A \ \text{and} \ B) = \frac{(2x_{AB})^r e^{-2x_{AB}}}{r!}.
\]

Using Mather’s formula (28), we obtain

\[
\rho_{AB} = \frac{1}{2} P(N_{AB} > 0) = \frac{1}{2} (1 - e^{-2x_{AB}}).
\]
Map functions

Formula (32) is **Haldane's map function**

\[ \rho = \frac{1}{2} (1 - e^{-2x}), \]

which may be inverted to give

\[ x = \frac{1}{2} \log(1 - 2\rho). \]

Note that for \( \rho \) and \( x \) small, we have \( e^{-2x} \approx 1 - 2x \) and \( \log(1 - 2\rho) \approx -2\rho \), which gives \( \rho \approx x \).

A more realistic map function than Haldane's, that takes chiasma interference into account, is the Kosambi map function

\[ \rho = \frac{1}{2} \frac{e^{2x} - e^{-2x}}{e^{2x} + e^{-2x}} = \frac{1}{2} \tanh(2x), \]

which gives

\[ x = \frac{1}{4} \log \frac{1 + 2\rho}{1 - 2\rho}. \]

The map functions of Morgan, Haldane, and Kosambi are plotted below:
Association studies

The purpose of such studies is to try to localize disease genes. The question we wish to answer is whether alleles (or genotypes) at a particular known locus, called a marker locus, occur at different frequencies in affected and unaffected individuals. If the answer is yes, there may be several explanations. First, the alleles in question might actually cause the disease (very interesting!). They could be merely linked to and in linkage disequilibrium with the disease gene, which would help to localize a gene locus that may have a causal relationship to the disease (since only loci on the same chromosome may be linked). Or the relationship could be caused by population stratification only, so it gives no indication of where a disease gene might be located.

The third case, a spurious result due to population stratification only, may be illustrated as follows: Suppose that in subpopulation 1, a disease is common and some marker allele A is also more common than in subpopulation 2, where the disease is rare. When we sample from a mixture of the two populations, the disease will be associated with allele A. The explanation is a lurking variable, namely the subpopulation from which an individual is drawn, which explains both incidence of the disease and frequency of the A allele.

Consider the following case/control study involving alcoholism and the dopamine $D_2$ receptor gene (Blum et al., 1990). The gene in question has two alleles, $A_1$ and $A_2$. A sample of 70 cadavers yielded the following count data:

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-alcoholic cadavers</td>
<td>9</td>
<td>61</td>
<td>70</td>
</tr>
<tr>
<td>Alcoholic cadavers</td>
<td>26</td>
<td>44</td>
<td>70</td>
</tr>
<tr>
<td>Total</td>
<td>35</td>
<td>105</td>
<td>140</td>
</tr>
</tbody>
</table>

From this table, we obtain a Pearson $\chi^2$ statistic of 11.01, yielding a p-value of .0009.

The problem with case/control studies is that the control sample (in this case, the non-alcoholic cadavers) must be chosen to resemble the case sample (in this case, the alcoholic cadavers) as closely as possible. For the example above, the association between alcoholism and the $A_1$ allele has been reassessed many times with conflicting results.

One way to correct for the effect of population stratification is to use relatives in the study. This is the idea of the so-called transmission/disequilibrium test.
Transmission/Disequilibrium test

The idea is to use parents of affected individuals to study association of the disease with various alleles. The TDT compares the frequency with which alleles are transmitted and not transmitted to affected offspring. This eliminates spurious associations due to population stratification.

In the TDT, the marker alleles potentially contributed by heterozygous parents to children with the disease are arranged in a $2 \times m$ contingency table, where $m$ is the number of alleles. The top row counts the number of alleles that are passed to affected children, while the bottom row counts alleles that are not passed. Note that homozygous parents do not provide any information about the association of the alleles in question to the disease, so they are excluded from the data.

Here is an example (Spielman et al., 1993). The disease in question is insulin-dependent diabetes mellitus (IDDM), and the marker is a locus near the insulin gene on chromosome 11. There are typically 3 alleles, but we combine two of them to consider “alleles” $A$ and $B$. The data are from 94 families with two or more IDDM children. Allele transmissions for 57 heterozygous parents are summarized below.

<table>
<thead>
<tr>
<th></th>
<th>$A$</th>
<th>$B$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitted</td>
<td>78</td>
<td>46</td>
<td>124</td>
</tr>
<tr>
<td>Not transmitted</td>
<td>46</td>
<td>78</td>
<td>124</td>
</tr>
<tr>
<td>Total</td>
<td>124</td>
<td>124</td>
<td>248</td>
</tr>
</tbody>
</table>

For a $2 \times 2$ table, the diagonal and off-diagonal entries will be equal—this is because homozygotes are excluded. One way of seeing this is the fact that the row totals MUST be equal, and in the $2 \times 2$ case the column totals MUST be equal as well because each parent-child combination contributes one allele to each column because all parents are heterozygous. The hypotheses being tested are

$H_0 : \rho = 1/2$ or linkage equilibrium

$H_1 : \rho < 1/2$ and linkage disequilibrium

Note that under $H_0$, the entries in the $i$th column are binomially $(n_i, 1/2)$ distributed, where $n_i$ is the $i$th column sum. In the $2 \times 2$ case, therefore, it is possible to compute exact p-values. For the example above, the p-value equals $P(|X - 62| \geq 16)$, where $X$ is a binomial $(124, 1/2)$ random variable. Mathematica gives this to be .00516.

Note that the usual Pearson chi-square test is equivalent to using a normal approximation to the binomial distribution (seeing this requires a bit of algebra).
Randomization tests

Next, we consider an example with more than 2 marker alleles at the locus in question. The table below summarizes marker data on 16 Costa Rican children afflicted with the disease ataxia-telangiectasia or AT (Uhrhammer et al., 1995).

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitted</td>
<td>3</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>28</td>
</tr>
<tr>
<td>Not transmitted</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>28</td>
</tr>
<tr>
<td>Total</td>
<td>3</td>
<td>4</td>
<td>22</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>11</td>
<td>56</td>
</tr>
</tbody>
</table>

We already noted that under $H_0$, the $i$th column tabulates the successes and failures for a binomial $(n_i, 1/2)$ random variable. Thus, we might consider the standardized residuals

$$Z_{ij} = \frac{n_{ij} - (n_i/2)}{\sqrt{(n_i/4)}}$$

and reject if they are in some sense too large. One way of obtaining a test statistic is to calculate the residual sum of squares; in this case, dividing this residual sum of squares by 2 gives the usual Pearson chi-square statistic

$$\chi^2 = \sum_{i=1}^{m} \sum_{j=1}^{2} Z_{ij}^2/2 = \sum_{i=1}^{m} Z_{i1}^2.$$ 

In the example above, we get $\chi^2 = 46.46$. Because of the small counts in some columns, the asymptotic chi-square approximation is not a good idea. However, it is possible to estimate a p-value using a technique called a randomization test.

The idea behind the randomization test is as follows: For the 28 heterozygous parents represented in the table, we know probabilistically how their alleles would segregate if the null hypothesis were true. Therefore, we can simulate many, many tables assuming the null hypothesis to be true and calculate a $\chi^2$ statistic for each of them. We then estimate the p-value as the proportion of those simulated $\chi^2$ statistics that are at least as large as the $\chi^2$ statistic we originally observed (in this case, 46.46).
Randomization tests cont’d

Let’s apply the ideas of the previous page to the following fabricated example. Suppose we’re studying the association between a particular marker locus and a disease. Assume the locus has three (codominant) alleles. We observe a group of children with the disease and the phenotypes of their parents, obtaining 65 (heterozygous parent, affected child) pairs. The data are summarized below.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmitted</td>
<td>18</td>
<td>23</td>
<td>24</td>
<td>65</td>
</tr>
<tr>
<td>Not transmitted</td>
<td>6</td>
<td>35</td>
<td>24</td>
<td>65</td>
</tr>
<tr>
<td>Total</td>
<td>24</td>
<td>58</td>
<td>48</td>
<td>130</td>
</tr>
</tbody>
</table>

The goal is to test the null hypothesis of linkage or linkage disequilibrium. Under the null hypothesis, the counts in each column of the table above are binomial with parameter 1/2. However, there is not enough given in the above table for us to determine the distribution of tables under the null hypothesis. We need to know the genotypes of the parents represented in the table. For this, we turn to the following data matrix:

<table>
<thead>
<tr>
<th></th>
<th>Transmitted</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Transmitted</td>
<td>0</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>Not transmitted</td>
<td>5</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>23</td>
<td>0</td>
</tr>
</tbody>
</table>

For example, we see that of the 17 A/B parents, 12 transmitted the A allele and 5 transmitted the B allele. It is not hard to verify that table (34) uniquely determines table (33), but not the other way around.

We need some statistic that measures in some sense the “closeness” of the observed data to the situation of perfect linkage disequilibrium or the absence of linkage. Examples are the Pearson $\chi^2$ statistic or the $Z_{max}$ statistic of the previous page. Another example is simply the probability of the observed table assuming the null hypothesis is true, namely

$$\begin{pmatrix} 17 \\ a_{12} \\ a_{13} \end{pmatrix} \begin{pmatrix} 7 \\ a_{21} \\ a_{23} \end{pmatrix} / 2^{65}.$$  

Let’s use the $\chi^2$ statistic to illustrate. The observed value of this statistic is 8.483. We would like to see how extreme this value is in relation to the null distrubution of the $\chi^2$ statistic (that is, the distribution of the $\chi^2$ statistic under the assumption that the null hypothesis is true). It is really hard to derive the exact distribution of this statistic. However, it is easy to draw a sample from the null distribution because the numbers $a_{12}$, $a_{13}$, and $a_{23}$ are binomial random variables with parameter 1/2 under the null distribution. Thus, we can simulate each of these three random variables, build the resulting table in the form of (33), and then compute the resulting $\chi^2$ statistic. I did this $10^4$ times; the number of $\chi^2$ values I got that were greater than or equal to the observed 8.483 is 623. Thus, we estimate a p-value of .0623.

Incidentally, since the asymptotic $\chi^2$ test gives a p-value of .0144, this serves as empirical evidence that the usual $\chi^2$ test is inapplicable in this situation.
Covariance estimates via expected information

Let’s return to the ABO example on p. 11. Recall that the ABO locus has 3 (common) alleles, with A and B codominant and O recessive. We used a dataset collected in a study of peptic ulcers:

<table>
<thead>
<tr>
<th>Blood type</th>
<th>A</th>
<th>B</th>
<th>AB</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>186</td>
<td>38</td>
<td>13</td>
<td>284</td>
</tr>
</tbody>
</table>

Assuming Hardy-Weinberg equilibrium, we used an EM algorithm to obtain the MLE \((\hat{p}_A, \hat{p}_B) = (.2136, .0501)\). Note that \(\hat{p}_O\) is uniquely determined by \(\hat{p}_A\) and \(\hat{p}_B\).

Suppose that we now wish to estimate the covariance matrix for this MLE, perhaps in order to give confidence intervals for the true parameters. We have already seen one way to do this, namely, by using the inverse of the observed information matrix or the expected information matrix evaluated at the MLE (see p. 20). Let’s see how to use the expected information in this example.

First, we build the Hessian matrix. Remember that we’re dealing with a multinomial model here with 4 categories. Let’s denote the probabilities of the 4 categories by \(\pi_A\), \(\pi_B\), \(\pi_{AB}\), and \(\pi_O\). Remember that each of these categories is a function of \(p_A\) and \(p_B\). The loglikelihood is

\[
\log(p_A, p_B) = k + n_A \log \pi_A + n_B \log \pi_B + n_{AB} \log \pi_{AB} + n_O \log \pi_O.
\]

Letting \(dl\) denote the row vector of partial derivatives of \(\log(p_A, p_B)\) with respect to \(p_A\) and \(p_B\), we obtain

\[
dl = \begin{pmatrix}
\frac{n_A}{\pi_A} d\pi_A \\
\frac{n_B}{\pi_B} d\pi_B \\
\frac{n_{AB}}{\pi_{AB}} d\pi_{AB} \\
\frac{n_O}{\pi_O} d\pi_O
\end{pmatrix}.
\]

Letting \(d^2l\) denote the Hessian matrix of second partial derivatives, and noting that \(E(n_A, n_{AB}, n_B, n_O) = n(\pi_A, \pi_B, \pi_{AB}, \pi_O)\), we saw in equation (24) on p. 20 that

\[
E(-d^2l) = n \begin{pmatrix}
\frac{1}{\pi_A} d\pi_A d\pi_A + \frac{1}{\pi_B} d\pi_B d\pi_B + \frac{1}{\pi_{AB}} d\pi_{AB} d\pi_{AB} + \frac{1}{\pi_O} d\pi_O d\pi_O
\end{pmatrix}.
\]

Let’s examine just one of the terms above. In this model,

\[
\pi_A = p_A^2 + 2p_A(1 - p_A - p_B) = 2p_A - p_A^2 - 2p_A p_B.
\]

Therefore, \(d\pi_A = (2 - 2p_A - 2p_B, -2p_A)\), or, more succinctly, \(d\pi_A = 2(p_O, -p_A)\). This gives

\[
d\pi_A d\pi_A = 4 \begin{pmatrix}
p_O^2 & -2p_O p_A \\
-2p_O p_A & p_A^2
\end{pmatrix}.
\]

Repeating this calculation for each of the four terms and summing gives the expected information matrix

\[
I_e = 4n \begin{pmatrix}
\frac{\hat{p}_O}{p_A + 2p_A p_O} + \frac{p_O}{p_A + 2p_O} + \frac{p_A}{p_A + 2p_A} + 1 & -\frac{p_O}{p_A + 2p_O} - \frac{p_O}{p_A + 2p_O} + \frac{3}{2} \\
-\frac{p_O}{p_A + 2p_O} - \frac{p_O}{p_A + 2p_O} + \frac{3}{2} & \frac{p_A}{p_A + 2p_O} + \frac{p_A}{p_A + 2p_O} + \frac{p_A}{p_A} + 1
\end{pmatrix}.
\]

(35)
Covariance estimates via bootstrapping

If we plug the MLE \((\hat{p}_A, \hat{p}_B, \hat{p}_O) = (0.2136, 0.0501, 0.7363)\) along with the value \(n = 521\) into the information expression (35) on the preceding page, we obtain the expected information matrix

\[
\begin{pmatrix}
5533.84 & 1208.28 \\
1208.28 & 2160.5
\end{pmatrix}, \quad \text{with inverse} \quad 10^{-4} \times \begin{pmatrix}
1.829 & -0.1023 \\
-0.1023 & 0.4687
\end{pmatrix}. \tag{36}
\]

Therefore, we can obtain the asymptotic standard errors for \((\hat{p}_A, \hat{p}_B, \hat{p}_O)\) as .0135, .0068, and .0145, respectively.

However, this was a bit of a headache. And in some cases, such a closed-form solution will be impossible. Thus, we consider an alternative to find (in this case) a covariance matrix estimate known as bootstrapping.

The idea behind bootstrapping a covariance estimate is this: The covariance matrix may be viewed as a function of whatever distribution produces the data \(n_A, n_B, n_{AB}, n_O\) in a random sample of size \(n = 521\). We don’t know what that distribution is; however, it is reasonable to assume that the empirical distribution of the observed data is “close to” the true distribution. Furthermore, we can determine the desired covariance matrix corresponding to the empirical distribution to any degree of accuracy we wish! This is because we can obtain a very large number of samples from the empirical distribution, find the MLE for each sample, and compute the sample covariance matrix of all of the MLEs thus obtained. By the law of large numbers, this sample covariance matrix converges to the true covariance matrix for the MLE obtained via the empirical distribution. Thus, in some sense the bootstrap is a way of estimating an approximation. The estimate can be made arbitrarily precise, but the quality of the approximation depends on the original sample.

Here’s how we could go about obtaining a bootstrap estimate of the covariance matrix of the MLE in this problem:

1. Select a large number \(B\). This is the number of bootstrap estimates.

2. For all \(i, 1 \leq i \leq B\), select a random sample of size \(n = 521\) from the empirical distribution.
   In other words, select 521 times with replacement from 186 A, 38 B, 13 AB, and 284 O.

3. Run the EM algorithm on the \(i\)th sample and obtain the MLE, which we’ll denote by \((\hat{p}_{A,i}^*, \hat{p}_{B,i}^*, \hat{p}_{O,i}^*)\).

4. After doing all \(B\) iterations, we have a \(2 \times B\) matrix \(P^*\) of maximum likelihood estimates. Find the sample covariance matrix \(S\) of these estimates as follows: First, center each of the two rows of \(P^*\) by subtracting its mean. Let the resulting matrix be called \(X\). Then the sample covariance matrix is \(S = XX^T/(B - 1)\).

5. Take \(S\) as the bootstrap estimate of the covariance.

Using \(B = 10^4\) in the above procedure, I obtained a bootstrap covariance estimate of

\[
10^{-4} \times \begin{pmatrix}
1.804 & -0.0741 \\
-0.0741 & 0.4713
\end{pmatrix}.
\]
Covariance estimates via parametric bootstrapping

**Parametric bootstrapping** is similar to bootstrapping, except that the bootstrap samples are not drawn from the empirical distribution. Instead, they are drawn from a distribution determined by parameters that we could estimate from the sample data.

In our ABO example, the obvious choice of parameters to estimate is \((p_A, p_B, p_O)\). We obtained the MLE \((.2136, .0501, .7363)\). We now need to draw a large number of samples of size \(n = 521\) from the distribution determined by these estimates. Plugging in the MLE values, we obtain

\[
\hat{\pi}_A = .3602, \quad \hat{\pi}_B = .0763, \quad \hat{\pi}_{AB} = .0214, \quad \hat{\pi}_O = .5421.
\]  

(37)

Thus, the parametric bootstrap proceeds as follows:

1. Select a large number \(B\). This is the number of bootstrap estimates.

2. For all \(i, 1 \leq i \leq B\), select a random sample of size \(n = 521\) from the multinomial distribution with probabilities given in (37).

3. Run the EM algorithm on the \(i\)th sample and obtain the MLE, which we’ll denote by \((p_{A,i}^*, p_{B,i}^*, p_{O,i}^*)\).

4. After doing all \(B\) iterations, we have a \(2 \times B\) matrix \(P^*\) of maximum likelihood estimates. Find the sample covariance matrix \(S\) of these estimates as follows: First, center each of the two rows of \(P^*\) by subtracting its mean. Let the resulting matrix be called \(X\). Then the sample covariance matrix is \(S = XX^T/(B-1)\).

5. Take \(S\) as the parametric bootstrap estimate of covariance.

Using \(B = 10^4\) in the above procedure, I obtained a parametric bootstrap covariance estimate of

\[
10^{-4} \times \begin{pmatrix} 1.835 & -1.106 \\ -1.106 & .4654 \end{pmatrix}.
\]

Note that the only difference between the parametric bootstrap and the empirical bootstrap in this example is that the former draws multinomial samples using the probabilities in (37), whereas the latter uses probabilities \((186/521, 38/521, 13/521, 284/521)\). In general, this similarity will not necessarily hold; it is possible that an empirical bootstrap procedure could look much different than a parametric bootstrap procedure.
Paternity testing

Consider the problem of attempting to determine whether a man is the father of a particular child. In this case the mother, child, and putative father are phenotyped at a number of different marker loci. It may be that the genetic information reveals that the man could not have been the father of the child, in which case the putative father is absolved. Otherwise, either the putative father is the actual father or a rare event has occurred. We wish to quantify how rare the event is.

One way to do this is by calculating a so-called exclusion probability, defined to be the probability that a random male would be excluded as the father. (On the next page we will see a second way to do it that involves calculating pedigree likelihoods.) Consider the following simple example involving two loci, the ABO locus and the ADA locus (the ADA locus is a codominant biallelic locus not on the same chromosome as ABO).

<table>
<thead>
<tr>
<th>ABO phenotype</th>
<th>ADA phenotype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mother</td>
<td>AB</td>
</tr>
<tr>
<td>Child</td>
<td>B</td>
</tr>
<tr>
<td>Putative father</td>
<td>B</td>
</tr>
</tbody>
</table>

The exclusion probabilities may be calculated for each locus individually or for all loci. Suppose we assume (based on other data) that the A, B, and O alleles have population frequencies .28, .06, and .66, respectively and that the 1 and 2 alleles at the ADA locus have frequencies .934 and .066, respectively.

For the ABO locus, we can see that the child inherited its mother’s B allele, and thus it must have inherited either a B or an O from its father. Thus, the only genotype that is impossible for the father of this child is AA. Therefore, the exclusion probability for the ABO locus is $P(AA) = p_A^2 = .28^2 = .078$.

By a similar argument, the only excluded ADA genotype is 1/1 since the child must have inherited the 2 allele from the father. Therefore, the exclusion probability for the ADA locus is $P(1/1) = p_1^2 = .934^2 = .872$.

Since the two loci are not linked, we can invoke independence to compute the overall exclusion probability

$$P(AA \text{ or } 1/1) = P(AA) + P(1/1) - P(AA \text{ and } 1/1) = .078 + .872 - (.078)(.872) = .882.$$ 

If the loci were linked, then the above calculation would still be valid as long as the loci were in linkage equilibrium because $P(AA \text{ and } 1/1)$ would still be given by $(p_{AP1})^2$. 

37
Bayesian approach to paternity testing

Consider the previous example of paternity testing. A different approach from the exclusion probability approach is to compute a **paternity index**, which is a ratio of two pedigree likelihoods. The numerator is the likelihood for the pedigree consisting of the mother, the child, and the putative father as the father of the child. The denominator is the likelihood for the pedigree consisting of the mother, the child, and a random male as the father of the child (along with the putative father as an unattached individual). In other words,

\[
\text{paternity index} = \frac{L(\text{data|paternity})}{L(\text{data|not paternity})}.
\]

The reason this approach is Bayesian is that the posterior odds of paternity given the data are equal to the prior odds of paternity times the paternity index. In other words, if \( \beta = P(\text{paternity|data}) \), then

\[
\frac{\beta}{1 - \beta} = \frac{\alpha}{1 - \alpha} \frac{L(\text{data|paternity})}{L(\text{data|not paternity})},
\]

where \( \alpha \) is the prior probability of paternity (based on nongenetic evidence, for example).

In our simple example, we are comparing two pedigrees. The first, or paternity, pedigree, is as follows:

```
1
B 1/2
```

```
2
AB 1/1
```

```
3
B 1/2
```

For the ABO locus only, the likelihood for this pedigree is .001391. For the ADA locus only, the likelihood for this pedigree is .053776. Therefore, the overall likelihood is the product of these numbers, or \( 7.4802 \times 10^{-6} \). We wish to compare these numbers to the corresponding ones for the not paternity pedigree:

```
4
?
```

```
2
AB 1/1
```

```
1
B 1/2
```

```
3
B 1/2
```

For the not paternity pedigree, we obtain likelihoods of .001002 for the ABO locus alone, .007098 for the ADA locus alone, and \( 7.1122 \times 10^{-6} \) overall. These results lead to the following paternity indices:

<table>
<thead>
<tr>
<th>Locus</th>
<th>ABO</th>
<th>ADA</th>
<th>Both</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paternity index</td>
<td>1.39</td>
<td>7.58</td>
<td>10.5</td>
</tr>
</tbody>
</table>

Note that the above numbers are equal to the posterior odds of paternity if we assume a prior probability of paternity equal to 1/2.
Lod scores

Suppose we are interested in mapping a disease locus and we wish to check to see if it is linked to a particular marker locus. If the genetic information on linkage comes in the form of a pedigree, we might do this by computing the likelihood \( L(p) \) as a function of the recombination fraction \( p \) between the disease locus and the marker locus. In order to make this likelihood interpretable, we standardize by dividing it by \( L(1/2) \). It is standard practice to then take the logarithm base 10 of the result. The quantity \( \log_{10}[L(p)/L(1/2)] \) is called a **lod score**. The word “lod” is a contraction of “logarithm of the odds,” although the likelihood ratio is not, strictly speaking, the odds. As a general rule of thumb, geneticists provisionally accept linkage if the maximum lod score on the interval \([0, 1/2] \) is at least 3.

As an example, consider the following pedigree from Ken Lange’s book. It is a pedigree depicting the autosomal dominant disease episodic ataxia along with alleles at 9 marker loci on chromosome 12.

The above pedigree is completely idealized, since we never get data on individuals’ haplotypes, let alone which gamete is responsible for a disease (the alleles marked with dots are descended from the offending gamete). In fact, this pedigree is merely a “best-guess” reconstruction from the true data, which are unavailable. Nonetheless, it will serve to demonstrate lod scores. Despite the complicated appearance of the pedigree above, it is actually quite simple to work with because all genotypes are known.
Lod scores cont’d

To see why the complicated-looking pedigree on the preceding page is actually quite simple to compute likelihoods for, recall that we normally find the likelihood as

$$
\sum_{(g_1, \ldots, g_n)} \prod_{all\ individuals\ i} \text{Pen}(x_i|g_i) \prod_{all\ founders\ j} \text{Prior}(g_j) \prod_{all\ children\ k} \text{Trans}(g_k|g_k’s\ mom; g_k’s\ dad),
$$

where the sum is over all possible genotype vectors \((g_1, \ldots, g_n)\). In this case, however, there is only a single possible genotype vector, so the likelihood does not involve any summation.

Consider the D12S372 locus, the fourth on the list for each individual. We wish to compute the lod score as a function of \(\rho\), the recombination fraction between the D12S372 locus and the disease locus (assuming there is a disease locus, of course). First, we note that all penetrances are 1. Second, we note that \(\rho\) does not influence any of the priors, which means that when we find \(L(\rho)/L(1/2)\), all the priors cancel. Thus, we are left with

$$
L(\rho) = \prod_{all\ children\ k} \text{Trans}(g_k|g_k’s \ mom; g_k’s \ dad).
$$

Once again, we simplify the lod score by noting that \(\rho\) only influences those transmitting probabilities involving the disease; all other transmission probabilities cancel. We are left with only 13 terms in the product, one for each child who got the disease or part of the diseased gamete. Twelve of these children are nonrecombinant, but the 13th, number 9004, is recombinant between the disease locus and D12S372. Thus, we obtain

$$
L(\rho) = k\rho(1 - \rho)^{12},
$$

where \(k\) denotes the constant that will cancel upon division by \(L(1/2)\). The lod score is plotted as a function of \(\rho\) below.
Location scores

A location score is analogous to a lod score. However, instead of being a function of a recombination fraction, the location score is a function of the position of the disease locus on the chromosome, measured in morgans or centimorgans. This requires us to fix an origin somewhere along the chromosome; this may be done arbitrarily.

If the distances between the markers are all known, then for a given position $d$ of the disease locus, it is possible to find the distance between the disease locus and each marker. If these distances are then converted into recombination fractions, the likelihood may be computed for entire pedigree. For the sake of simplicity, assume Haldane’s model, with recombinations occurring independently on disjoint intervals and with $\rho = (1 - e^{-2|d-x|})/2$, where $|d-x|$ represents map distance in morgans between the disease locus and position $x$.

The location score as a function of the position $d$ of the disease locus is defined to be $\log_{10}[L(d)/L(\infty)]$, where we abuse notation slightly by changing $L(d)$ from a function of the recombination fraction to a function of the map distance.

In the episodic ataxia example, four of the markers, pY2/1, pY21/1, KCNA5, and D12S99, are very tightly clustered, so we only use one of them, pY2/1. The map distances in centimorgans for the remaining 6 markers are

$$S91\overset{1\text{cm}}{\rightarrow}S100\overset{1\text{cm}}{\rightarrow}CACNL1A1\overset{3\text{cm}}{\rightarrow}S372\overset{3\text{cm}}{\rightarrow}pY2/1\overset{4\text{cm}}{\rightarrow}S93$$

You may want to see if you can reconstruct the following location score plot based on the pedigree. The zero on the horizontal scale is taken to be the S91 locus, and the units on that axis are centimorgans.

Notice that the location score does rise above the level of 3, indicating that the disease locus probably resides between D12S372 and pY2/1.
Markov chains

Suppose that $Z_1, Z_2, Z_3, \ldots$ is a sequence of random variables each of which can take finitely many (say $m$) possible values, called states. If the probability distribution of $Z_n$ is determined completely by the value of $Z_{n-1}$, the sequence is called a Markov chain. In this case, because $Z_t$ can take only finitely many values, this Markov chain is a finite state Markov chain. Also, since the random variables $Z$ are indexed by the counting numbers $1, 2, 3, \ldots$, this Markov chain is called a discrete time Markov chain. It is also possible to define a continuous time Markov chain, in which the random variables $Z$ are indexed by real numbers $t$ and we write the chain as $\{Z_t \mid t \geq 0\}$ or $\{Z(t) \mid t \geq 0\}$. We will not discuss continuous time Markov chains. (Note: a continuous time Markov chain can still be a finite state Markov chain.)

A bit of clarification is probably in order. The Markovian condition for our discrete time Markov chain requires that the distribution of $Z_n$ (for all $n > 1$) may depend on the past, but any dependency is completely specified by the value of $Z_{n-1}$:

$$P(Z_n = i \mid Z_1, Z_2, \ldots, Z_{n-1}) = P(Z_n = i \mid Z_{n-1}).$$

(38)

Although many textbooks define a finite state discrete time Markov chain to be any sequence satisfying equation (38), typically they also assume that the probabilities in (38) do not depend on $n$. We shall make this assumption here.

Consider the following simple examples:

1. Let $Z_n$ be the indicator of heads for the $n$th coin flip in a sequence of flips. Then the $Z_n$ are iid (independent and identically distributed) and they trivially form a Markov chain; the Markovian condition (38) is satisfied because both sides are equal to $P(Z_n = i)$ by independence.

2. Suppose there are 20 marbles in a box, some of them red and some of them white. We run an experiment in which we draw one marble at random from the box. If a white marble is drawn, then we replace it by a red marble; if a red marble is drawn, we replace it by a white marble. Then we repeat the process. Let $Z_n$ denote the number of white marbles in the box before the $n$th draw. Then $Z_1, Z_2, \ldots$ is a Markov chain. In fact, we have

$$P(Z_n = j \mid Z_{n-1} = i) = \begin{cases} i/20 & \text{if } j = i - 1 \text{ and } j \geq 0 \\ (20 - i)/20 & \text{if } j = i + 1 \text{ and } j \leq 20 \\ 0 & \text{otherwise.} \end{cases}$$

Since the probability in equation (38) does not depend on $n$, we may define

$$t_{ij} = P(Z_n = j \mid Z_{n-1} = i).$$

(39)

The value $t_{ij}$ is called a transition probability since it equals the probability of a transition from state $i$ to state $j$. Define the $m \times m$ matrix $M$ to be the matrix with its entry in the $i$th row and $j$th column equal to $t_{ij}$. Then $M$ is called the transition matrix for the Markov chain. Notice that conditional on $Z_{n-1} = i$, we know that $Z_n$ must equal something, which means that if we fix $i$ and sum equation (39) over all possible $j$, we get 1. Equivalently, we see that each row of the transition matrix $M$ must sum to 1.
Simplistic model for a DNA sequence

As a concrete example of a finite state discrete time Markov chain, let’s consider the sequence of letters A, C, G, T that make up a strand of DNA. Suppose that we assume that the letters, in order, form a Markov chain with 4 states. Swartz et al. (1962) give the following transition matrix for this Markov chain:

$$M = \begin{pmatrix}
    A & C & G & T \\
    A & .32 & .18 & .23 & .27 \\
    C & .37 & .23 & .05 & .35 \\
    G & .30 & .21 & .25 & .24 \\
    T & .23 & .19 & .25 & .33
\end{pmatrix}.$$  \hspace{1cm} \hspace{1cm} (40)

This transition matrix tells us, for example, that if the chain is currently at state C, it will make a transition to state T with probability .35. Note that each row of the transition matrix sums to 1.

The transition matrix summarizes the random behavior of the Markov chain if we make one step forward in time. What if we wish to consider the random behavior if we make two steps forward in time? In other words, what is the distribution of $Z_n$ conditional on $Z_{n-2}$? Let’s answer this question for a single transition, say from C to T. There are 4 ways we could make a transition from C to T in 2 steps, depending on which state occurs in between. We can find the probability of each of these paths and then sum them up:

$$P(C \rightarrow A \rightarrow T) = P(C \rightarrow A)P(A \rightarrow T) = .37(.27) = .0999$$
$$P(C \rightarrow C \rightarrow T) = P(C \rightarrow C)P(C \rightarrow T) = .23(.35) = .0805$$
$$P(C \rightarrow G \rightarrow T) = P(C \rightarrow G)P(G \rightarrow T) = .05(.24) = .0120$$
$$P(C \rightarrow T \rightarrow T) = P(C \rightarrow T)P(T \rightarrow T) = .35(.33) = .1155$$
Total = .3079

Therefore, the transition matrix for a two-step transition would have .3079 in row C and column T. Think about how we got this number; in the calculations above, all we did was matrix multiply row C times column T. But this is exactly what is accomplished if we multiply the entire matrix $M$ by itself—the $i,j$ entry of $T^2$ is the $i$th row times the $j$th column. We conclude that the transition matrix for a two-step transition is just $M^2$. Using the same argument repeatedly, we conclude that the transition matrix for a $k$-step transition is $M^k$.

Suppose that the Markov chain eventually “forgets” its starting state. If $M^k$ has a limit as $k \to \infty$, then we reason that in this limit, all rows should be the same. Consider repeated squaring of the matrix $M$ in our example:

$$M^2 = \begin{pmatrix}
    A & C & G & T \\
    A & 0.3001 & 0.1986 & 0.2076 & 0.2937 \\
    C & 0.2990 & 0.1965 & 0.1966 & 0.3079 \\
    G & 0.3039 & 0.2004 & 0.2020 & 0.2937 \\
    T & 0.2948 & 0.2003 & 0.2074 & 0.2975
\end{pmatrix}$$
$$\quad \quad , \quad M^4 = \begin{pmatrix}
    A & C & G & T \\
    A & 0.2991 & 0.1991 & 0.2042 & 0.2976 \\
    C & 0.2990 & 0.1991 & 0.2043 & 0.2977 \\
    G & 0.2991 & 0.1990 & 0.2042 & 0.2977 \\
    T & 0.2991 & 0.1991 & 0.2042 & 0.2977
\end{pmatrix},$$
$$\quad \quad , \quad M^8 = \begin{pmatrix}
    A & C & G & T \\
    A & 0.2991 & 0.1991 & 0.2042 & 0.2977 \\
    C & 0.2991 & 0.1991 & 0.2042 & 0.2977 \\
    G & 0.2991 & 0.1991 & 0.2042 & 0.2977 \\
    T & 0.2991 & 0.1991 & 0.2042 & 0.2977
\end{pmatrix}.$$
Stationary distributions

On the preceding page we saw that in the limit as \( k \to \infty \), \( M^k \) converges to a matrix in which all rows are identical. Let \( \pi \) denote this common row. In our example,

\[
\pi = (.2991, .1991, .2042, .2977).
\]

Naturally, multiplying \( \lim_{k \to \infty} M^k \) by \( M \) does not change it. Therefore, we know that \( \pi M = \pi \). In words, this means that if we assume \( \pi \) gives a probability distribution on the states of the Markov chain, then the distribution on the states does not change as the chain moves forward in time. For this reason, any \( \pi \) satisfying \( \pi M = \pi \) is known as a stationary distribution or an equilibrium distribution of the Markov chain.

It is a fact (that we won’t prove) that any finite state Markov chain has at least one stationary distribution. It is also true that under certain conditions, which we will call ergodic conditions, the stationary distribution is unique. The two ergodic conditions are irreducibility and aperiodicity. Irreducibility means that all pairs of states communicate; i.e., given any two states \( i \) and \( j \), there is an integer \( k \) such that there is a nonzero probability of reaching \( j \) from \( i \) in \( k \) steps. Aperiodicity means that for any state \( j \), there exists \( N \) large enough such that there is a nonzero probability of getting from \( j \) back to \( j \) in exactly \( k \) steps for ALL \( k > N \). In our DNA example, it is easy to see that the ergodic conditions are satisfied since all entries in the transition matrix are positive.

Another way to view the stationary distribution arises from considering the equation \( \pi M = \pi \). If we take the transpose of both sides, we obtain \( M^T \pi^T = \pi^T \). Thus, \( \pi^T \) is an eigenvector of \( M^T \) corresponding to the eigenvalue 1. (Since as we noted previously, all finite state Markov chains have stationary distributions, the transpose of any transition matrix must have one of its eigenvalues equal to 1.) Here’s what Splus gives as the first eigenvalue and eigenvector of the transpose of our DNA transition matrix:

\[
> \text{eigen}(t(M))$val[1]
\]

\[
[1] \ 1+0i
\]

\[
> \text{eigen}(t(M))$vec[,1]
\]

\[
[1] \ -0.7039771+0i \ -0.4685396+0i \ -0.4806658+0i \ -0.7006305+0i
\]

Note: Because \( M^T \) is not a symmetric matrix, its eigenvalues and eigenvectors could be complex numbers, which is why Splus includes \(+0i\) with each number. Clearly the Splus output does not match our stationary distribution \( \pi \), but remember that \( \pi \) is supposed to be a probability distribution, which means that its entries must sum to one (which is not true of an eigenvector in general). Therefore, we must divide each entry by the sum of all entries in order to see the familiar form of \( \pi \):

\[
> \text{ev_eigen}(t(M))$vec[,1]
> \text{ev}/\text{sum}(ev)
\]

\[
[1] \ 0.2990794+0i \ 0.1990556+0i \ 0.2042073+0i \ 0.2976577+0i
\]
MCMC: Introduction

Markov chain Monte Carlo (MCMC) is a method of sampling from the stationary distribution of a Markov chain. The term “Monte Carlo” refers to simulating random behavior, and basically MCMC is just simulating a Markov chain.

We will see further examples of MCMC later, so I won’t say much about it here except to explain how it could be accomplished in our DNA Markov chain example. We have already seen two ways to establish the equilibrium distribution of that Markov chain: One was by successive matrix multiplication and another was by finding an eigenvector. Recall that each of these methods is just a way of answering one question: If we run the chain long enough so that it “forgets” its initial state, what is the probability that it will be in state $i$ for any $i$? One way to approximate this is by simulation: Simulate a Markov chain according to the transition matrix $M$ for a while. Then after enough steps to ensure that the initial state’s influence is negligible, known as a burnin period, observe which state you’re in. By definition, this final state is an observation from the (approximate) stationary distribution. Repeat this process many times, and keep a tally of the ending states. We may estimate the stationary distribution by the observed proportions.

We may implement this in Splus. I’ve written a short function that takes an initial state, a transition matrix, and a number $k$ as input and returns the final state of the Markov chain after $k$ repetitions:

```R
> function(state, M, rep = 1) {
  m <- dim(M)[1]
  for(i in 1:rep)
    state <- sample(m, size = 1, prob = m[state, ])
  state
}

Since our multiplications earlier revealed that the rows of $M^8$ are identical to at least 4 decimal places, 8 is a good number to use as a burnin period. We repeat the MCMC procedure described above 10000 times and tally the results:

```R
> for(i in 1:10000) x[i]_transition(1,M,rep=8)
> table(x)/10000
   1  2  3  4
0.302 0.2032 0.1999 0.2949
```

Thus we see that our MCMC procedure provides a reasonable estimate to $\pi$, which in this case is known to be (.2991, 1991, .2042, .2977). In fact, if we assume that our burnin period is long enough, it is possible to give confidence intervals for the true $\pi$ since the estimate is nothing but the sample mean for a multinomial sample with underlying probability vector $\pi$. 

45
Monte Carlo expectations

In the DNA sequence Markov chain model of page 43, suppose we wanted to know the probability that a randomly sampled nucleotide (letter) from this chain is C. We know from the analyses on pages 43 and 44 that the answer is .1991. However, those analyses used repeated multiplication of a matrix or eigenvector decomposition, neither of which is practical in problems with very large state spaces.

On the other hand, page 45 suggested a Monte Carlo approach: simulate many generations of this Markov chain, keep track of the results, and estimate the desired probability by the correct ratio. In other words, we obtain

$$\hat{\pi}_C = \frac{1}{N} \sum_{i=1}^{N} I\{Z_i \text{ is a C}\},$$

where $Z_i$ is the $i$th sampled state of the Markov chain. This estimates

$$\pi_C = \mathbb{E}\left[I\{Z_i \text{ is a C}\}\right].$$

Thus, the Monte Carlo scheme uses the sample mean $\hat{\pi}_C$ as an estimate of the corresponding population mean $\pi_C$.

In general, suppose that we wish to estimate a quantity that can be written as an expectation, say

$$\theta = \mathbb{E}[f(Z)], \quad (41)$$

where $Z$ is a random variable with some known distribution. If we draw a random sample $Z_1, \ldots, Z_N$ of size $N$ from the distribution, the sample mean

$$\hat{\theta} = \frac{1}{N} \sum_{i=1}^{N} f(Z_i) \quad (42)$$

is a reasonable estimate of $\theta$. In fact, by the law of large numbers, $\hat{\theta}$ may be shown to converge to $\theta$ as $N$ grows larger and larger.

The above argument was framed in terms of a random sample $Z_1, \ldots, Z_N$, which implies that the $Z_i$ are independent and identically distributed (iid). However, the iid assumption is not necessary in order for the sample mean (42) to converge to the expectation (41). In particular, the convergence is usually still true if $Z_1, \ldots, Z_N$ are successive states of a Markov chain. This means that they are not independent, though their dependence may be very weak.
An example: Monte Carlo integration

In the DNA example on the previous page, it was possible to find the expectation $\pi_C$ directly using, for example, an eigenvector analysis. This makes the Monte Carlo method a bit artificial. However, it is often the case that an expectation is so complicated that an analytic solution is infeasible.

As an example of this, suppose we are interested in evaluating a complicated integral such as

$$I = \int_0^1 \frac{\sin x}{x + \cos^2 x} \, dx.$$

A closed-form solution is infeasible (I think!), so we must instead resort to numerical methods. In order to use Monte Carlo integration, we first write the integral as an expectation. To see one way of doing this, suppose $U$ is a uniform random variable on $(0, 1)$. Then $\mathbb{E}[f(U)]$, where $f(U)$ is some function, equals $\int_0^1 f(u) \, du$. Therefore, for an appropriate choice of $f(U)$, we can easily rewrite $I$ as an expectation involving $U$:

$$I = \mathbb{E} \left[ \frac{\sin U}{U + \cos^2 U} \right].$$

Now that we have $I$ expressed as a mean, the Monte Carlo scheme is to draw a large sample $U_1, \ldots, U_N$ from the distribution of $U$, then evaluate the sample mean

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N f(U_i) = \frac{1}{N} \sum_{i=1}^N \frac{\sin U_i}{U_i + \cos^2 U_i}.$$

I did this for $N = 10000$ and got the following result:

> u_runif(10000)
> fu_sin(u)/(u+cos(u)^2)
> mean(fu)

[1] 0.3639998

Of course, we’d like to get an idea of how precise this estimate of $I$ is. The standard error of the estimator above is easy to compute:

> sqrt(var(fu)/10000)

[1] 0.00185107

Thus, we could take $0.3631 \pm 1.96(0.00185)$, or $0.3631 \pm 0.0036$, as an approximate 95% confidence interval for $I$.

Mathematica reports an exact value of $I$, obtained via numerical integration, of 0.363582. Because numerical integration is easy to do in this case, the utility of Monte Carlo integration for this problem is small—however, there are many examples in which even a numerical answer is infeasible, making Monte Carlo methods very important.
Importance sampling

Importance sampling is a method of reducing the variance of an estimator obtained by a Monte Carlo method like the integration example on the preceding page. Suppose that the random variable $U$ has density $g(u)$ on the interval $(0, 1)$. (On the previous page, where $U$ was uniform, $U$ had the constant density $g(u) = 1$.) Then the expectation of a function of $U$, say $h(U)$, is given by

$$
E[h(U)] = \int_0^1 h(u)g(u)\,du.
$$

(43)

In the Monte Carlo integration example, we wanted to evaluate $\int_0^1 f(u)\,du$, where $f(u) = \sin u/(u + \cos^2 u)$. Thus, we should take $h(u) = f(u)/g(u)$ if expectation (43) is to give the desired result.

What might be a sensible choice for $g(u)$? Ideally, we would like an estimator with small variance. Thus it makes sense to choose $g(u)$ in such a way that $h(u)$ is nearly constant, because the estimator we will use is the sample mean of many instances of $h(U)$. In other words, we should choose a density $g(u)$ that is roughly the same shape as $f(u)$. Here is a graph of $f(u)$:

![Graph of f(u)](image)

Since $f(u)$ is roughly linear, if we can choose $g(u)$ to be linear as well, we can reduce the variance of $h(U)$. It turns out that the square root of a uniform random variable has density $g(u) = 2u$ (we won’t discuss why here—perhaps you can verify it).

Thus, we sample $U_1, \ldots, U_N$ with density $g(u) = 2u$, then estimate the integral by

$$
\hat{I} = \frac{1}{N} \sum_{i=1}^{N} h(U_i) = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{\sin U_i}{U_i + \cos^2 U_i} \right) \left( \frac{1}{2U_i} \right).
$$

> u_sqrt(runif(10000))  # The square root of uniform has the desired density.
> hu_sin(u)/(u+cos(u)^2)/(2*u)
> mean(hu)  # Here is the estimator of I.

[1] 0.3634434

> sqrt(var(hu)/10000)  # The standard error is smaller than before.

[1] 0.000287837

Thus, the importance sampling scheme above gives a standard error of the estimator smaller than it was for the uniform sampling scheme on the preceding page.
The Metropolis-Hastings algorithm

We return now to the topic of finite state, discrete time Markov chains developed on page 42. Suppose we wish to sample from a multinomial distribution with probability vector $\pi$. The Metropolis-Hastings algorithm is a way to construct a Markov chain that has its stationary distribution equal to $\pi$. Suppose we start in state $i$. Conditional on $i$, generate some proposal state $j$ for the next time point. We may denote the distribution used to obtain this proposal by

$$q_{ij} = P(i \to j) = P(\text{moving to state } j \mid \text{ currently in state } i).$$

The Metropolis-Hastings algorithm proceeds by either moving next to state $j$ or remaining once more in state $i$, based on the flip of a biased coin. The correct probability of moving to state $j$ is

$$\min \left\{ 1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}} \right\}.$$

Thus, the Markov chain transition probability of moving from state $i$ to state $j$ is

$$t_{ij} = q_{ij} \min \left\{ 1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}} \right\}. \quad (44)$$

Note in particular that it is only necessary to know $\pi_j / \pi_i$ in order to use the Metropolis-Hastings algorithm; the individual values of $\pi_j$ and $\pi_i$ are unimportant. To see that $t_{ij}$ defines a Markov chain with the correct equilibrium distribution $\pi$, we need to check that for all $j$,

$$\pi_j = \sum_i \pi_i t_{ij}. \quad (45)$$

The right side of equation (45) is the probability of being in state $j$ at time $n + 1$ if the chain has distribution $\pi$ at time $n$, which should be equal to $\pi_j$ under equilibrium. To check (45), note that

$$\pi_i t_{ij} = \begin{cases} \pi_i q_{ij} & \text{if } \pi_j q_{ji} \geq \pi_i q_{ij} \\ \pi_j q_{ji} & \text{otherwise} \end{cases} \quad (46)$$

and

$$\pi_j t_{ji} = \begin{cases} \pi_j q_{ij} & \text{if } \pi_j q_{ji} \geq \pi_i q_{ji} \\ \pi_i q_{ij} & \text{otherwise} \end{cases}. \quad (47)$$

Combining (46) and (47), we see that $\pi_i t_{ij} = \pi_j t_{ji}$ for all $i$ and $j$. Summing this result over $i$, we obtain

$$\sum_i \pi_i t_{ij} = \pi_j \sum_i t_{ji} = \pi_j.$$

Thus, the Metropolis-Hastings scheme really does satisfy condition (45).

Here is a summary of the Metropolis-Hastings algorithm:

1. Let $n = 1$ and start the chain by choosing the initial value of $Z_1$.
2. Let $i$ be the state of the Markov chain at time $n$; that is, $Z_n = i$.
3. Pick the possible next state $j$ using transition probabilities $q_{ij}$.
4. Generate a uniform $(0,1)$ variable $U$. If $U > \pi_j q_{ji} / \pi_i q_{ij}$, stay at the current state (let $Z_{n+1} = i$); otherwise, move to $j$ (let $Z_{n+1} = j$).
5. Increment $n$ and go back to step 2.
Gibbs Sampling

Suppose that we are interested in sampling from a distribution on $R^2$. Let $(X, Y)$ denote a random vector with this distribution, and suppose that the joint distribution is complicated or unknown but that the conditional distributions $X \mid Y$ and $Y \mid X$ are simple. It is possible to implement a particular example of the Metropolis-Hastings algorithm called Gibbs sampling in order to sample from the desired joint distribution.

We define a Markov chain as follows. First, fix some initial value for $Y_0$. Then, alternate between sampling from $X \mid Y$ and sampling from $Y \mid X$ as follows:

1. Let $n = 1$.
2. Sample from the distribution $X \mid Y_{n-1}$ and call the result $X_n$.
3. Sample from the distribution $Y \mid X_n$ and call the result $Y_n$.
4. Increment $n$ and return to step 2.

The result of this scheme is the Markov chain $(X_1, Y_1), (X_2, Y_2), \ldots$. If $X$ and $Y$ take on finitely many values, we can show that the Gibbs sampler is a special case of Metropolis-Hastings. Consider step 2 in the above algorithm. Let $i$ and $j$ denote possible states of the vector $(X, Y)$. Since the value of $Y$ does not change in step 2, the proposal probability $q_{ij}$ is nonzero only if states $i$ and $j$ both have the same value of $Y$. Thus, if state $i$ is $(a_i, y)$ and state $j$ is $(a_j, y)$, then

$$q_{ij} = P(X = a_j \mid Y = y) = \frac{\pi_j}{P(Y = y)}.$$

Therefore, it is immediately obvious that $\pi_j q_{ij} = \pi_i q_{ji}$, which means that the Metropolis-Hastings acceptance probability is always 1. For step 3 in the above algorithm, there would be a different set of proposal probabilities, say $q_{ji}$, but the logic would be the same. Thus, the Gibbs sampler interweaves two Metropolis-Hastings algorithms, each with the correct equilibrium distribution. To see why this works, imagine two different Markov chain transition matrices $A$ and $B$ with the same equilibrium distribution $\pi$. Then $\pi = \pi A = \pi B$, so it is immediately clear that $\pi = \pi (AB)$. We conclude that the Markov chain obtained by the Gibbs sampler has the desired equilibrium distribution $\pi$.

As a simple example, suppose that $X$ and $Y$ are both Bernoulli random variables (that is, they take only the values 0 and 1), with conditional distributions as follows:

$$P(X = 1 \mid Y) = \begin{cases} 2/3 & \text{if } Y = 0 \\ 1/3 & \text{if } Y = 1 \end{cases} \quad P(Y = 1 \mid X) = \begin{cases} 3/4 & \text{if } X = 0 \\ 3/7 & \text{if } X = 1 \end{cases}$$

It is not immediately obvious (at least to me!) what the joint distribution of $X$ and $Y$ is. However, it is easy to sample from the joint distribution using Gibbs sampling. Ten thousand sampled values from the Gibbs sampler (after a burn-in period of 30) gives the following estimate of the joint distribution, along with the true joint distribution:

\[
\begin{array}{c|c|c}
Y = 0 & Y = 1 \\
\hline
X = 0 & 0.1324 & 0.2688 \\
X = 1 & 0.3961 & 0.2027 \\
\end{array}
\quad
\begin{array}{c|c|c}
Y = 0 & Y = 1 \\
\hline
X = 0 & 0.1333 & 0.2667 \\
X = 1 & 0.4000 & 0.2000 \\
\end{array}
\]

50