3
LD, Haplotyping and Genotype Imputation

3.1 Introduction

Non-random assortment of marker alleles into haplotypes (due to linkage) means that many fewer haplotypes will exist in a population than would be possible with free recombination between loci. Such a pattern of non-random co-occurrence of alleles on chromosomes (haplotypes) is referred to as linkage disequilibrium (LD). Linkage equilibrium, on the other hand, means that alleles at separate loci are combined at random to form haplotypes so that the haplotype frequencies are equal to the product of the marginal allele frequencies at each locus.

LD is widespead in the human genome and is a loose function of the physical distance between markers (because recombination rates generally increase with physical distance). However, the magnitude and extent of LD also depends on the population history. Roughly speaking, the younger the relationship between the individuals (chromosomes) in a population the greater the LD. This is because there have been fewer meioses (and thus fewer opportunities for recombination) in a younger population. Populations that have recently expanded from a small number of founding individuals, such as the populations of Quebec, or Finland, show greater LD than say the population of Ethiopia for this reason.

Linkage disequilibrium between loci on human chromosomes typically decays rapidly when their physical distance exceeds about 50 kb, on average (see e.g., Reich et al., 2001). LD is therefore most useful for fine-mapping of disease loci on small intervals of chromosomes. With sufficiently dense markers LD provides a way of indirectly identifying disease loci by first detecting an association with a neutral marker in LD with a disease locus that has not been genotyped (Goldstein et al.,

2001). A denser set of markers (or sequencing of all variants in the region) is then used to attempt to find the causal locus.

In this chapter we describe statistics for quantifying the magnitude of LD between pairs of marker loci, methods for inferring haplotype phase, methods for finding regions (blocks) of chromosomes with limited haplotype diversity, and methods for identifying "tagging SNPs" for use in GWASs. Finally, we consider recent methods for inferring the state of genotypes at SNP loci that are not directly typed in a particular study, so-called "genotype imputation." These methods together provide a suite of tools for preliminary analysis of GWAS data. Specific methods for detecting disease associations using multilocus data are considered in Chapter 4.

3.2 Linkage disequilibrium statistics

In the absence of recombination, alleles located on the same chromosome are co-transmitted to offspring. Here we consider a simple statistic that is often used to quantify the non-random assortment of alleles onto chromosomes (e.g., non-random frequencies of haplotypes). If two alleles are sufficiently far apart on a chromosome, recombination occurs at each generation (meiosis) and the alleles assort independently to form genotypes in offspring (e.g., Mendel's second law applies). Alleles at markers that are near one another on a chromosome will undergo recombination (and independent assortment) during only some fraction of the meioses and therefore have the potential for LD. Consider a pair of biallelic genetic markers located on the same chromosome. Marker locus 1 has alleles A and a, marker locus 2 has alleles B and b. The possible haplotypes are

```
A-B
a-B
a-b
A-b
```

We denote the frequency of haplotype A-B as p_{AB} and so on. The marginal (or total) frequency of allele A is obtained by summing up the frequencies of all haplotypes that contain allele A at locus 1,

$$p_A = p_{AB} + p_{Ab},$$

and the marginal frequencies of other alleles are obtained similarly. Free recombination between neutral loci (and random mating) would result in frequencies of the four haplotypes that are the product of the marginal allele frequencies at each locus. With less recombination the haplotype frequencies will evolve toward these "equilibrium" frequencies but at any point in time some level of LD may persist in the population.

3.2.1 Pairwise disequilibrium coefficient

The pairwise coefficient of disequilibrium, D, between two biallelic loci (with alleles A and a at locus 1 and alleles B and b at locus 2) is defined as (see e.g., Li, 1955)

$$D = p_{AB} - p_A p_B. \tag{3.1}$$

The departure of frequencies from equilibrium values for each of the four distinct haplotypes can be concisely summarized in terms of D as follows,

$$\begin{pmatrix} p_A p_B + D & p_A p_b - D \\ p_a p_B - D & p_a p_b + D \end{pmatrix}.$$

To obtain a statistic that varies between -1 and +1, equation 3.1 above can be normalized by dividing by its maximum possible value to obtain (Lewontin, 1964),

$$D' = D/D_{max},$$

where

$$D_{max} = \begin{cases} \min(p_A[1-p_B], [1-p_A]p_B) & \text{if} \quad D \geq 0 \\ \min(p_A p_B, [1-p_A][1-p_B]) & \text{if} \quad D < 0 \end{cases}$$

Because the sign of the statistic D' depends on the labeling of alleles, which is arbitrary, the absolute value $|D'|$ is most often used. To illustrate the use of this statistic, consider a sample with $p_{AB} = 0.6$, $p_{aB} = 0.1$, $p_{ab} = 0.1$ and $p_{Ab} = 0.2$. The marginal allele frequencies are

$$\begin{aligned} p_A &= 0.6 + 0.2 = 0.8 \\ p_B &= 0.6 + 0.1 = 0.7 \\ p_a &= 0.1 + 0.1 = 0.2 \\ p_b &= 0.1 + 0.2 = 0.3, \end{aligned}$$

the disequilibrium coefficient D is

$$D = 0.6 - 0.8 \times 0.7 = 0.04,$$

and the normalized disequilibrium coefficient D' is

$$D' = \frac{0.04}{\min(0.24, 0.14)} = \frac{0.04}{0.14} = 0.286.$$

In this case, the observed frequency ($p_{AB} = 0.6$) of A-B is greater than expected under random assortment ($p_A \times p_B = 0.8 \times 0.7 = 0.56$) and there is positive linkage disequilibrium of the alleles. Two extreme cases of D' are worth considering. First, there is the case of complete disequilibrium. An example is $p_{AB} = p_{ab} = 0.5$ and $p_{aB} = p_{Ab} = 0$, so only two of the four possible haplotypes are observed. In this case, $p_A = p_B = p_b = p_a = 0.5$, and

$$D' = \frac{0.5 - 0.5^2}{0.5^2} = \frac{1 - 0.5}{0.5} = 1.$$

The other extreme is complete equilibrium. An example is $P_{AB} = p_{Ab} = p_{aB} = p_{ab} = 0.25$. In this case, $p_A = p_B = p_b = p_a = 0.5$, and

$$D = p_{AB} - p_A p_B = 0.25 - 0.5 \times 0.5 = 0,$$

and therefore $D' = 0$. The statistic $|D'|$ is often used to summarize linkage disequilibrium across the genome in human population samples. Overall, there is a negative relationship between $|D'|$ and physical distance between markers in humans because markers spaced at greater intervals along a chromosome experience more recombination events, on average, per generation.

3.2.2 EM algorithm for estimating disequilibrium

Note that haplotype frequencies (e.g., p_{AB}, p_{Ab}, etc) are needed to apply equation 3.1 to genomic data and calculate D'. However, as noted in Chapter 1, SNP genotyping studies generate counts of individual multilocus genotypes which provide information about genotype frequencies at a pair of loci and not necessarily haplotype frequencies. The problem is that the phase is uncertain for individuals that are heterozygous at both loci. For example, an individual with the two-locus genotype $A/a, B/b$ might have either of the diplotypes $A - B, a - b$ or $A - b, a - B$.

Hill (1974) provided a solution to the problem of inferring haplotype frequencies from two-locus genotype counts. His solution exploits

a technique now known as the "Expectation-Maximization (EM) algorithm." Hill's paper extends an EM approach that was developed previously for allele frequency estimation, the so-called "gene counting method" of Ceppellini et al. (1955). The basic data from a two locus genotyping study (with two alleles at each locus) are the counts of numbers of individuals with each of the 9 possible combinations of 2 locus genotypes shown in Table 3.1. The population haplotype frequency pa-

	BB	Bb	bb	Total
AA	N_{11}	N_{12}	N_{13}	$N_{1\cdot}$
Aa	N_{21}	N_{22}	N_{23}	$N_{2\cdot}$
aa	N_{31}	N_{32}	N_{33}	$N_{3\cdot}$
Total	$N_{\cdot 1}$	$N_{\cdot 2}$	$N_{\cdot 3}$	N

Table 3.1 *SNP genotype counts for 9 possible genotype combinations at two biallelic loci.*

rameters and (unobservable) haplotype counts are shown in Table 3.2. If we were able to directly observe the haplotypes (rather than the geno-

Haplotype	$A-B$	$A-b$	$a-B$	$a-b$
Frequency	p_{AB}	p_{Ab}	p_{aB}	p_{ab}
Count	n_{AB}	n_{Ab}	n_{aB}	n_{ab}

Table 3.2 *Population haplotype frequency parameters and and counts for 4 possible allele combinations at two biallelic loci.*

type combinations) the probability of the observed counts would follow a multinomial distribution,

$$\Pr(n_{AB}, n_{Ab}, n_{aB}, n_{ab}) = \binom{2N}{n_{AB} n_{Ab} n_{aB} n_{ab}} p_{AB}^{n_{AB}} p_{Ab}^{n_{Ab}} p_{aB}^{n_{aB}} p_{ab}^{n_{ab}}, \quad (3.2)$$

where $2N = n_{AB} + n_{Ab} + n_{aB} + n_{ab}$ and $p_{AB} + p_{Ab} + p_{aB} + p_{ab} = 1$, The log-likelihood (score) function is

$$L = C + \sum_{i,j} n_{ij} \log(p_{ij}). \quad (3.3)$$

where C is an irrelevant constant that is not a function of the frequency parameters. By differentiating the log-likelihood function with respect to the frequency parameters (using Lagrangian multipliers to constrain

the frequency parameters to sum to 1), setting the partial derivatives to equal zero and simultaneously solving the resulting equations the maximum likelihood estimates of the frequency parameters are found to be (see e.g., Rice, 1995),

$$\hat{p}_{ij} = \frac{n_{ij}}{2N}. \tag{3.4}$$

Assuming HWE the expected haplotype counts (as a function of the genotype counts) are

$$\begin{aligned}
n_{AB} &= 2N_{11} + N_{12} + N_{21} + \left(\frac{p_{AB}p_{ab}}{p_{Ab}p_{aB} + p_{AB}p_{ab}} \right) N_{22}, \\
n_{ab} &= 2N_{33} + N_{23} + N_{32} + \left(\frac{p_{Ab}p_{aB}}{p_{Ab}p_{aB} + p_{AB}p_{ab}} \right) N_{22}, \\
n_{Ab} &= 2N_{13} + N_{12} + N_{23} + \left(\frac{p_{Ab}p_{aB}}{p_{Ab}p_{aB} + p_{AB}p_{ab}} \right) N_{22}, \\
n_{aB} &= 2N_{31} + N_{21} + N_{32} + \left(\frac{p_{Ab}p_{aB}}{p_{Ab}p_{aB} + p_{AB}p_{ab}} \right) N_{22},
\end{aligned}$$

Note that $p_{AB}p_{ab}/(p_{Ab}p_{aB} + p_{AB}p_{ab})$ is the probability that an individual with heterozygous genotype Aa/Bb has diplotype $A-B/a-b$ assuming that haplotypes combine in individuals according to HWE proportions (random mating). We can reduce the number of parameters in these equations by noting that the maximum likelihood estimates of allele frequencies at each locus are

$$\begin{aligned}
\hat{p}_A &= \hat{p}_{AB} + \hat{p}_{Ab} \\
&= (2N_{11} + N_{12} + N_{21} + 2N_{13} + N_{12} + N_{23} + N_{22})/2N \\
&= \frac{N_{1\cdot} + N_{2\cdot}/2}{N},
\end{aligned}$$

and

$$\begin{aligned}
\hat{p}_B &= \hat{p}_{AB} + \hat{p}_{aB} \\
&= (2N_{11} + N_{12} + N_{21} + 2N_{31} + N_{21} + N_{32} + N_{22})/2N \\
&= \frac{N_{\cdot 1} + N_{\cdot 2}/2}{N},
\end{aligned}$$

and that

$$\hat{p}_{Ab} = \hat{p}_A - \hat{p}_{AB},\ \hat{p}_{aB} = \hat{p}_B - \hat{p}_{AB},\ \text{and}\ \hat{p}_{ab} = 1 - \hat{p}_{AB} - \hat{p}_A - \hat{p}_B.$$

We now substitute these expressions for the 3 haplotype frequency parameters (those other than p_{AB}) in terms of allele frequencies and $\hat{p}_{AB}$

into the formula for n_{AB} to obtain a predictor of the expectation of n_{AB} given $\hat{p}_{AB}$,

$$\mathbb{E}(n_{AB}|\hat{p}_{AB}) = 2N_{11} + N_{12} + N_{21} + \frac{\hat{p}_{AB}(1 + \hat{p}_{AB} - \hat{p}_A - \hat{p}_B)N_{22}}{(\hat{p}_A - \hat{p}_{AB})(\hat{p}_B - \hat{p}_{AB}) + \hat{p}_{AB}(1 + \hat{p}_{AB} - \hat{p}_A - \hat{p}_B)}. \quad (3.5)$$

The only unspecified variable in this formula is $\hat{p}_{AB}$ which is the MLE of the parameter that we wish to obtain. We now make use of the EM algorithm to obtain the MLE. The Expectation step in the EM algorithm calculates the expected value of the count n_{AB} given a specified value of $\hat{p}_{AB}$ (using equation 3.5 above) and the Maximization step uses the maximum likelihood estimator to obtain a revised estimate of $\hat{p}_{AB}$ using $\mathbb{E}(n_{AB}|\hat{p}_{AB})$ in place of n_{AB} in the MLE formula. The procedure is started with an arbitrary initial value for $\hat{p}_{AB}$. Th expectation-maximization steps are performed repeatedly, each time replacing the current value of $\hat{p}_{AB}$ with the MLE value obtained at the previous iteration, until the estimate converges to the MLE. At the ith iteration, the maximization step is

$$\hat{p}_{AB}(i) = \frac{\mathbb{E}(n_{AB}|\hat{p}_{AB}(i-1))}{2N}.$$

Once the MLE $\hat{p}_{AB}$ has been calculated the MLE of the disequilibrium coefficient is calculated as

$$\hat{D} = \hat{p}_{AB} - \hat{p}_A\hat{p}_B,$$

which follows from the invariance property of maximum likelihood estimators (Rice, 1995).

Another possible approach to the problem of estimating pairwise disequilibrium is to infer estimates of haplotype phase for individuals over a region of the genome using a multilocus genotype sample and then calculate the disequilibrium coefficent directly for pairs of loci from the inferred haplotype frequencies. Haplotype phase for an individual can potentially be inferred either by using genotype information from relatives, or by using a population sample of genotypes and an inference model based on the assumption of random mating (HWE). Statistical methods for haplotype inference will be discussed in Section 3.3.

3.3 Haplotype phase inference

3.3.1 Deterministic algorithms

One of the earliest multilocus haplotype inference algorithms is a deterministic procedure proposed by Clark (1990). The algorithm is "deteministic" in the sense that it does not model uncertainties of inferred haplotypes (uncertainties due to sampling and uncertainties inherent to the genotype data). For example, several haplotype resolutions may have very similar probability, given the genotype data, and a deterministic method will typically infer one haplotype resolution in such cases. Alternative haplotype inference methods that model such statistical uncertainty will be described below. Let $\mathbf{G} = \{G_{il}\}$ be a matrix of dimension $N \times L$ where G_{il} is the genotype of individual i at locus l and N individuals are sampled at random from a population and genotyped at L loci. Define $F_i = \{H_{i1}, H_{i2}\}$ to be the (unobservable) diplotype of individual i, where H_{i1} and H_{i2} are the haplotypes of maternal and paternal chromosomes. Let H_{i1j} be the allele present at locus j on haplotype 1 of individual i.

The first step of the algorithm identifies all individuals that are fully homozygous (e.g., homozygous at all loci) or that have a single heterozygous locus and therefore have unambiguous diplotypes. Suppose that there are n_0 individuals whose diplotype is fully resolved in this way and let $F^{(0)} = \{F_i^{(0)}\}$ be the set of known diplotypes for those individuals. The next step of the algorithm chooses an individual j with 2 or more heterozygous loci and and determines whether its genotype, G_j, is compatible with a haplotype in $F^{(0)}$. The genotype G_j is compatible with haplotype H_{il} if $H_{ilk} \subset G_{jk}$ for all $k = 1, \ldots, L$ (e.g., the allele found at every locus of the haplotype is present in one or more copies in the genotype). When a compatible haplotype, H_{il}, is found, one haplotype for individual j is resolved as $H_{jl} = H_{il}$ and the other haplotype is then completely specified. We add the newly resolved diplotype to the set, $F^{(1)} = F^{(0)} \bigcup F_j$, $n_1 = n_0 + 1$, and repeat the procedure.

To illustrate Clark's algorithm consider a situation in which 3 individuals are genotyped at 3 loci. The gentotypes are as follows:

$$
\begin{aligned}
G_1 &= \{\mathrm{A/A},\ \mathrm{B/b},\ \mathrm{C/c}\}, \\
G_2 &= \{\mathrm{A/A},\ \mathrm{B/B},\ \mathrm{C/C}\}, \\
G_3 &= \{\mathrm{A/a},\ \mathrm{b/b},\ \mathrm{C/c}\}.
\end{aligned}
$$

Genotype G_2 is fully homozygous and can therefore be resolved,

$$F^{(0)} = \left\{ \begin{array}{rcl} F_2 & = & [A-B-C]/[A-B-C] \end{array} \right.$$

Next we choose genotype G_1 which is compatible with haplotype $F_1 = F_2$ and infer its phase,

$$F^{(1)} = \left\{ \begin{array}{rcl} F_2 & = & [A-B-C]/[A-B-C] \\ F_1 & = & [A-B-C]/[A-b-c] \end{array} \right.$$

Finally, we choose genotype G_3 which is compatible with haplotype $F_{12} = [A-b-c]$ and resolve its phase,

$$F^{(2)} = \left\{ \begin{array}{rcl} F_2 & = & [A-B-C]/[A-B-C] \\ F_1 & = & [A-B-C]/[A-b-c] \\ F_3 & = & [A-b-c]/[a-b-C] \end{array} \right.$$

Here again we choose the phase of the first haplotype for individual 3 to be that of the compatible haplotype from individual 1, $H_{12} = [A-b-c]$ and the phase of the other chromosome is then determined.

The deterministic algorithm has an intuitive appeal. However, there are several disadvantages inherent to deterministic algorithms. First, the haplotype resolutions can depend on the order in which the genotypes are solved (this is the case for Clark's algorithms). Clark deals with this problem by permuting the order of the genotype resolutions in separate applications of the algorithm to obtain multiple solutions; a concensus of the solutions if used to obtain the final estimate. Second, a deterministic algorithm provides no measure of the statistical uncertainty of the inferred diplotypes. In many cases, multiple diplotypes exist that have very similar probability given the genotypes and choosing one resolution as a point estimate ignores this source of uncertainty. This is particularly important when the resulting haplotypes are used in a subsequent hypothesis test (a test of association, for example) as the test will have an incorrect type I error rate (typically the significance will be overestimated).

Several recent deterministic algorithms attempt to improve on Clark's method in various ways (Song et al., 2005; Liang and Wang, 2008). One advantage of deterministic algorithms is that they often require less computation time than probabilitic methods. However, given current computing resources (and the likelihood of future parallelization for use on multiple processors) the use of probabilitistic algorithms for the analysis of most datasets is now practical (even for datasets composed

of a large number of markers) suggesting that deterministic algorithms should be avoided.

3.3.2 EM algorithm

The EM algorithm of Hill (1974), presented in the previous section as a method for estimating D using multilocus genotypes, also provides estimates of population haplotype frequencies because MLEs are available for all four haplotype frequencies. Namely, $\hat{p}_{AB}$, $\hat{p}_{Ab} = \hat{D} - \hat{p}_A(1 - \hat{p}_B)$, and so on. In 1995, several papers appeared that extended Hill's method to allow haplotype frequency inference using several linked loci (Excoffier and Slatkin, 1995; Hawley and Kidd, 1995; Long et al., 1995). The methods of Excoffier and Slatkin (1995) and Long et al. (1995) are essentially similar, while the precise methodology of Hawley and Kidd (1995) is unknown because the method was published as a program note that lacked essential details of the implementation.

For simplicity, a description of the EM algorithm for multilocus haplotype inference will be given here that attempts to follow the notation of the two locus example in section 3.2.2, which differs from that of either Excoffier and Slatkin (1995) or Long et al. (1995). As noted by Excoffier and Slatkin (1995), there are c_i possible distinct diplotype combinations for genotype G_i where

$$c_i = \begin{cases} 2^{s_i - 1} & \text{if} \quad s_i > 0 \\ 1 & \text{if} \quad s_i = 0, \end{cases}$$

and s_i denotes the number of heterozygous loci for genotype G_i. Let

$$n_i = \sum_{j=1}^{N} I_i(F_j), \tag{3.6}$$

be the counts of haplotypes, where

$$I_i(F_j) = \begin{cases} 2 & \text{if} \quad H_{j1} = H_{j2} = i \\ 1 & \text{if} \quad H_{j1} = i \text{ and } H_{j2} \neq i \text{ or } H_{j1} \neq i \text{ and } H_{j2} = i \\ 0 & \text{if} \quad H_{j1} \neq i \text{ and } H_{j2} \neq i \end{cases}$$

The log-likelihood of the (unobservable) haplotype counts (assuming random mating) is

$$\log L = C \sum_{i=1}^{h} n_i \log(p_i),$$

where h is the number of distinct haplotypes, p_i is the population frequency of haplotype i, and C is an irrelevant constant. As noted in the previous section, the MLEs of the frequency parameters of this multinomial distribution, given the haplotype counts, are

$$\hat{p}_i = \frac{n_i}{2N}.$$

This formula defines the maximization step. Let $\hat{p} = \{\hat{p}_i\}$ be a vector of length h of the MLEs of population haplotype frequencies. The expectation step involves calculating the expected haplotype counts, n_i, given the estimated haplotype frequencies and individual genotypes. This expectation is given by

$$\mathbb{E}(n_i|\hat{p}, G) = \sum_{l=1}^{N} \sum_{j=1}^{c_l} I_i(F_j) \Pr(F_j|\hat{p}) \Pr(G_l|F_j), \tag{3.7}$$

where

$$\Pr(G_l|F_j) = \begin{cases} 1 & \text{if } F_j \text{ is compatible with } G_l \\ 0 & \text{otherwise} \end{cases},$$

and we assume HWE to obtain,

$$\Pr(F_j|\hat{p}) = \begin{cases} 1 & \text{if} \quad c_l = 1 \\ \hat{p}_k^2 & \text{if} \quad H_{j1} = H_{j2} = k \text{ and } c_l > 1 \\ 2\hat{p}_k\hat{p}_l & \text{if} \quad H_{j1} = k, H_{j2} = l \text{ or } H_{j2} = l, H_{j1} = k \text{ and } c_l > 1. \end{cases}$$

The EM algorithm then proceeds as follows: (1) determine all possible diplotypes (and haplotypes) for each multilocus genotype and the total number of possible unique haplotypes h; (2) assign arbitrary initial haplotype frequency parameter values $\hat{p}[0]$; (3) for each of $i = 1, \ldots, h$ propose a new value for the frequency estimate using EM:

$$\hat{p}_i(1) = \mathbb{E}(n_i|\hat{p}[0], G)/(2N).$$

(4) repeat step (3) until the estimates converge to a stable value.

Example: Haplotype phase for 3 loci via EM algorithm

To illustrate the application of the EM algorithm for 3 biallelic loci in a simple situation, we consider a case in which only 4 of the $2^3 = 8$ possible haplotypes have population frequencies greater than zero. Arbitrarily labelling the two alleles at each locus as 0 and 1 we can then specify the ordered alleles of the haplotypes using a binary string. For example, 001 represents the haplotype with allele 0 at loci 1 and 2, and allele 1 at locus 3. The 8 possible haplotypes and their population frequencies

are shown in Table 3.3. Given the 4 haplotypes present in the popula-

Index	Haplotype	Frequency
1	000	0.3
2	001	0.1
3	010	0.1
4	100	0.5
5	110	0.0
6	101	0.0
7	011	0.0
8	111	0.0

Table 3.3 *Population haplotype frequencies used to simulate multilocus genotypes for EM algorithm example. An arbitrary index number is assigned to each haplotype.*

tion, there are $4 + 4(3)/2 = 10$ possible distinct multilocus genotype combinations. Multilocus genotypes were simulated for $N = 20$ randomly mating individuals using the haplotype frequencies specified in Table 3.3. Only 6 of the 10 possible multilocus genotypes were observed in the simulated data. The counts of simulated genotypes are shown in Table 3.4. Note that 3 of the 6 distinct genotypes observed in the sam-

Index	Diplotype	Genotype	c_i	N_i
1	$(1,1)$	$0/0,0/0,0/0$	$c_1 = 1$	$N_1 = 2$
2	$(1,2)$	$0/0,0/0,0/1$	$c_2 = 1$	$N_2 = 4$
3	$(1,3)$	$0/0,0/1,0/0$	$c_3 = 1$	$N_3 = 5$
4	$(2,3)$	$0/0,0/1,0/1$	$c_4 = 2$	$N_4 = 2$
5	$(2,4)$	$0/1,0/0,0/1$	$c_5 = 2$	$N_5 = 3$
6	$(3,4)$	$0/1,0/1,0/0$	$c_6 = 2$	$N_6 = 4$

Table 3.4 *Counts of genotypes and actual diplotype combinations (known from simulation) for simulated sample of 3 biallelic loci in $N = 20$ randomly mating individuals.*

ple each have $c = 2$ possible diplotype combinations. By enumerating all possible haplotypes for each distinct genotype we see that there are $h = 7$ possible haplotypes (e.g., the haplotypes in Table 3.3 with indexes 1 through 7) given the sample of genotypes. The possible diplotypes for the sampled genotypes with $c = 2$ (indexed 4-6) are shown in Table 3.5.

Our goal will be to obtain maximum likelihood estimates of the pop-

Index	Genotype	Possible Diplotypes
4	$0/0, 0/1, 0/1$	$(1,7)$ or $(\mathbf{2},\mathbf{3})$
5	$0/1, 0/0, 0/1$	$(1,6)$ or $(\mathbf{2},\mathbf{4})$
6	$0/1, 0/1, 0/0$	$(1,5)$ or $(\mathbf{3},\mathbf{4})$

Table 3.5 *The possible diplotypes (haplotype indexes in parentheses) for the 3 simulated genotype patterns with $c = 2$. The true diplotype is known for the simulated data and is shown in bold.*

ulation frequencies, $\hat{p}$, of these 7 possible haplotypes using the EM algorithm described above. Note that $\hat{p}_i$ denotes the MLE of the population frequency of the haplotype with index i. The key step is to work out formulae for the expected counts of each haplotype conditional on the frequencies. By applying equation 3.7 one finds the expected counts to be

$$\begin{aligned}
\mathbb{E}(n_1|\hat{p}, G) &= 5 + 4 + 2(2) + 2\left(\frac{\hat{p}_1\hat{p}_7}{\hat{p}_1\hat{p}_7 + \hat{p}_2\hat{p}_3}\right) \\
&\quad + 4\left(\frac{\hat{p}_1\hat{p}_5}{\hat{p}_1\hat{p}_5 + \hat{p}_3\hat{p}_4}\right) + 3\left(\frac{\hat{p}_1\hat{p}_6}{\hat{p}_1\hat{p}_6 + \hat{p}_2\hat{p}_4}\right), \\
\mathbb{E}(n_2|\hat{p}, G) &= 4 + 2\left(\frac{\hat{p}_2\hat{p}_3}{\hat{p}_1\hat{p}_7 + \hat{p}_2\hat{p}_3}\right) + 3\left(\frac{\hat{p}_2\hat{p}_4}{\hat{p}_1\hat{p}_6 + \hat{p}_2\hat{p}_4}\right), \\
\mathbb{E}(n_3|\hat{p}, G) &= 5 + 2\left(\frac{\hat{p}_2\hat{p}_3}{\hat{p}_1\hat{p}_7 + \hat{p}_2\hat{p}_3}\right) + 4\left(\frac{\hat{p}_3\hat{p}_4}{\hat{p}_3\hat{p}_4 + \hat{p}_1\hat{p}_5}\right), \\
\mathbb{E}(n_4|\hat{p}, G) &= 4\left(\frac{\hat{p}_3\hat{p}_4}{\hat{p}_3\hat{p}_4 + \hat{p}_1\hat{p}_5}\right) + 3\left(\frac{\hat{p}_2\hat{p}_4}{\hat{p}_1\hat{p}_6 + \hat{p}_2\hat{p}_4}\right), \\
\mathbb{E}(n_5|\hat{p}, G) &= 4\left(\frac{\hat{p}_1\hat{p}_5}{\hat{p}_3\hat{p}_4 + \hat{p}_1\hat{p}_5}\right), \\
\mathbb{E}(n_6|\hat{p}, G) &= 3\left(\frac{\hat{p}_1\hat{p}_6}{\hat{p}_1\hat{p}_6 + \hat{p}_2\hat{p}_4}\right), \\
\mathbb{E}(n_7|\hat{p}, G) &= 2\left(\frac{\hat{p}_1\hat{p}_7}{\hat{p}_1\hat{p}_7 + \hat{p}_2\hat{p}_3}\right).
\end{aligned}$$

Each iteration of the EM algorthm then involves calculating an updated expectation for counts of each of haplotypes 1-6 using the above formulae and then generating a new estimate of each $\hat{p}_i$ using the formula (for iteration j),

$$\hat{p}_i(j) = \mathbb{E}(n_i|\hat{p}(j-1), G)/(2N).$$

Note that because the MLEs of frequencies are constrained to sum to 1,

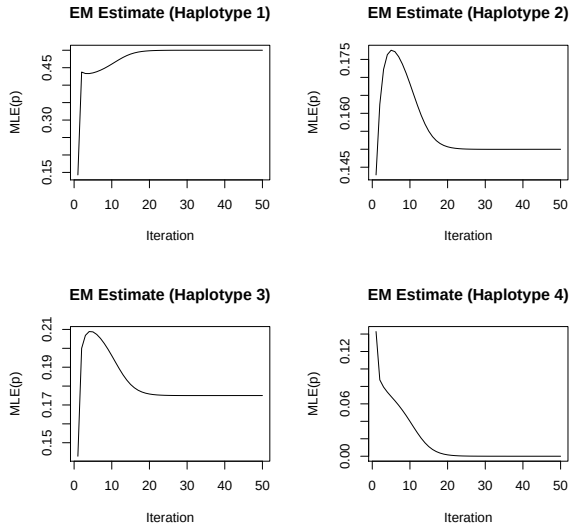

Figure 3.1 Estimates of first 4 haplotype frequency parameters obtained using the EM algorithm and plotted as a function of the iteration number. Clockwise from the top upper left panel the four figures show MLEs of $\hat{p}_1$, $\hat{p}_2$, $\hat{p}_4$, and $\hat{p}_3$.

only parameters indexed 1-6 need to be iterated and the estimate of the last frequency parameter is $\hat{p}_7 = 1 - \sum_{i=1}^{6} \hat{p}_i$.

The ML estimates of population frequency for the first 4 haplotypes (obtained via the EM algorithm) are plotted for each of 50 iterations in Figure 3.1. Note that all four frequencies converge to stable values after at most about 25 iterations of the algorithm. The MLEs of $\hat{p}$ for the first 6 haplotypes, obtained after 50 iterations of the EM algorithm, for the simulated genotype data are shown in Table 3.4, along with the estimates obtained using the actual haplotype frequencies in the sample (known for the simulated data) and the true haplotype frequencies used for the simulation. Note that the discrepancy between the true population haplotype frequencies and the frequencies inferred from sampled haplotypes is less than that for estimates obtained using the EM algorithm applied to the sampled genotypes, reflecting the uncertainty of statistical inference due to either population sampling effects alone, or due to both sampling effects and uncertain phase, respectively. In this example, the sample size is rather small and the resulting estimates are quite inaccurate.

	$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$	$\hat{p}_4$	$\hat{p}_5$	$\hat{p}_6$
Actual	0.30	0.10	0.10	0.50	0.00	0.00
Haplotypes	0.32	0.22	0.28	0.18	0.00	0.00
Genotypes (EM)	0.50	0.15	0.17	0.00	0.10	0.08

Table 3.6 *Actual haplotype frequencies used for simulation (row 1) and ML estimates of haplotype frequencies obtained either using diplotypes, known from simulation (row 2), or using multilocus genotypes and applying the EM algorithm (row 3).*

A major disadvantage of the EM algorithm for haplotype inference is that it requires all the frequencies of all possible haplotypes to be maintained in computer memory. If there are many loci the number of possible haplotypes can be enormous. Qin et al. (2002) have implemented a partition-ligation algorithm for EM inference that reduces the dimensionality of the problem and can greatly improve the performance of EM.

Inferring individual diplotypes

The EM algorithm outlined above aims to obtain MLEs of the population frequencies of haplotypes. In many situations, the primary interest is the most probable haplotype combination (diplotype) for each sampled individual. It is straightforward to construct an empirical Bayes estimator of the posterior probability of each diplotype combination for an individual given the MLEs of population haplotype frequencies and the individual's multilocus genotype. If G_i is the genotype of individual i and $\hat{p}$ is a vector of the MLEs of haplotype frequencies the empirical Bayes estimator of the posterior probability is given by

$$\Pr(F_i|\hat{p}, G_l) = \frac{1}{\Pr(G_i)}\Pr(G_i, F_i|\hat{p}) = \frac{1}{\Pr(G_i)}\Pr(G_i|F_i)\Pr(F_i|\hat{p}),$$

where,

$$\begin{aligned}\Pr(G_i) &= \sum_{j=1}^{c_i} \Pr(G_i|F_i)\Pr(F_i|\hat{p}). \\ &= \sum_{i=1}^{c_l} \Pr(F_i|\hat{p}),\end{aligned}$$

where the first term disappears because the sum is only over diplotypes compatible with the genotype.

As an example, we consider the simulated genotypes presented in Table 3.4. If individual i has the 3-locus genotype $G_i = \{0/1, 0/1, 0/0\}$ then $c_i = 2$ and the possible diplotypes are $F_i \in \{1-1-0/0-0-0, 1-0-0/0-1-0\}$. Diplotype $[1,5] = 0-0-0/1-1-0$ is a combination of haplotypes 1 and 5, and diplotype $[3,4] = 1-0-0/0-1-0$ is a combination of haplotypes 3 and 4. Thus,

$$\Pr(F_i = [1,5]|\hat{p}) = 2\hat{p}_1\hat{p}_5 = 0.30 \times 0.00 = 0,$$
$$\Pr(F_i = [3,4]|\hat{p}) = 2\hat{p}_3\hat{p}_4 = 0.28 \times 0.18 = 0.05,$$

where the values for $\hat{p}$ come from Table 3.6. Thus, in this case the true diplotype of the individual has posterior probability,

$$\begin{aligned}\Pr(F_i = [3,4]|\hat{p}, G_l) &= \frac{\Pr(F_i = [3,4]|\hat{p})}{\Pr(F_i = [1,5]|\hat{p}) + \Pr(F_i = [3,4]|\hat{p})} \\ &= \frac{0.05}{0.05 + 0.00} = 1.0.\end{aligned}$$

If needed, a point estimate of the diplotype can be obtained from the posterior probability, for example one could use

$$\max_{F_i} \Pr(F_i|G_i, \hat{p}),$$

where F_i is maximized over the set of c_i haplotypes compatible with genotype G_i.

3.3.3 Bayesian algorithms

The Bayesian approach to haplotype inference aims to jointly infer the posterior probability of both the population haplotype frequencies and the individual diplotype resolutions. Using the same notation as above, the joint posterior probability is,

$$\begin{aligned}\Pr(F, p|G) &= \frac{1}{\Pr(G)}\Pr(G|F)\Pr(F|p)f(p), \\ &= \frac{1}{\Pr(G)}\prod_{i=1}^{N}\Pr(G_i|F_i)\Pr(F_i|p)f(p),\end{aligned}$$

where $f(p)$ is a prior probability distribution on population haplotype frequencies and

$$\Pr(G) = \sum_F \int_p \Pr(G|F)\Pr(F|p)f(p)dp.$$

In general, it is difficult to directly evaluate the above equations and the Bayesian haplotype inference methods that have thus far been proposed make use of Markov chain Monte Carlo (MCMC) methods that provide numerical estimates of posterior probabilities.

3.3.4 Gibbs sampler

The principle underlying MCMC is to construct an algorithm that generates samples from a Markov chain with a stationary distribution that is the probability distribution of interest (in Bayesian inference this is usually the posterior distribution of parameters). The Markov chain is simulated on a computer until it reaches a stationary distribution and samples are then collected from the chain to make inferences about the parameters (see Gamerman, 1997).

The Bayesian haplotype inference methods discussed below (Stephens et al., 2001; Niu et al., 2002) make use of a particularly simple MCMC method known as the Gibbs sampler (see Gamerman, 1997). Briefly, the Gibbs sampler generates samples from the k-dimensional joint probability distribution

$$f(\theta_1, \theta_2, \ldots, \theta_k),$$

by successively simulating values from the full-conditional distribution. The basic algorithm is as follows:

1. set $j = 1$ and assign arbitrary starting values $\theta^{(0)} = \{\theta_1^{(0)}, \ldots, \theta_k^{(0)}\}$.
2. generate a new $\theta^{(j)}$ by successively simulating $\theta_i^{(j)} \sim f(\theta_i | \theta_{-i}^{(j-1)})$ for $i = 1, \ldots, k$ where $\theta_{-i}^{(j-1)}$ is $\theta^{(j-1)}$ with the ith element removed.
3. set $j = j + 1$ and repeat step 2 until chain converges.

In this formulation, the variables are systematically updated (simulated) but it is also permissible to randomly choose variables for updates. The stationary chain is iterated on a computer and the variables sampled at intervals (to decrease autocorrelation). The frequency at which the variable takes particular values (discrete variable) or occurs in particular intervals (continuous variable) in the chain estimates the probability associated with those values in the target distribution.

The PHASE algorithm

One of the limitations of haplotype inference via the EM algorithm described above is that the number of possible haplotype resolutions,

given the genotypes, can become very large requiring much computer memory. The Bayesian algorithm of Stephens et al. (2001) outlined below can avoid the need to maintain a vector of the population haplotype frequencies by analytically integrating over haplotype frequencies and using approximations that only deal with the sample diplotypes in the algorithm.

Here we describe only algorithm 1 of Stephens et al. (2001) which they refer to as the niave Gibbs sampler. Let $F^{(j)}$ be the set of individual diplotypes at iteration j of a MCMC analysis. The Gibbs sampler method of Stephens et al. (2001) proceeds as follows:

1. set $j = 1$ and assign $F^{(0)} = \{F_1^{(0)}, \ldots, F_N^{(0)}\}$ compatible with G.
2. choose random individual i and simulate $F_i^{(j)} \sim \Pr(F_i|G, F_{-i}^{(j-1)})$.
3. set $F_l^{(j)} = F_l^{(j-1)}$ for all $l \neq i$.
4. set $j = j + 1$ and repeat from step 2 until chain converges.

Here, we need to simulate from the full conditional distribution which is proportional to

$$\Pr(F_i|G, F_{-i}^{(j-1)}) \propto \Pr(F_i|F_{-i}^{(j-1)}) \propto \pi(H_{i1}|F_{-i}^{(j-1)})\pi(H_{i2}|F_{-i}^{(j-1)}, H_{i1}),$$

where we have dropped G as we are only considering haplotypes compatible with genotypes. Taking the product of the conditional haplotype frequencies at the right of the above equation corresponds to an assumption of HWE (random mating) in combining haplotypes to form diplotypes. The proportionality constant is calculated as

$$\begin{aligned}\Pr(G_i) &= \sum_{F_i \in c_i} \Pr(F_i|F_{-i}^{(j-1)})\Pr(G_i|F_i), \\ &= \sum_{F_i \in c_i} \Pr(F_i|F_{i-1}^{(j-1)}). \end{aligned} \tag{3.8}$$

Note that

$$\Pr(F_i|F_{-i}) = \frac{\Pr(F_i, F_{-i})}{\Pr(F_{-i})} = \frac{\int_p \Pr(F_i|p)f(p)dp}{\int_p \prod_{j\neq i} \Pr(F_j|p)f(p)dp}.$$

Thus, the conditional distribution depends on the specific form of the prior since it is obtained by integrating over this prior. Stephens et al. (2001) chose a prior for haplotype frequencies that leads to a particularly simple result for the conditional distribution. The so-called parent-

independent mutation (PIM) model gives the probability

$$\pi(H_{ji}|F_{-i}) = \sum_{\alpha \in E} \sum_{s=0}^{\infty} \frac{r_\alpha}{r} \left(\frac{\theta}{r+\theta} \right)^s \frac{r}{r+\theta} (P^s)_{\alpha H_{ji}},$$

where r_α is the number of haplotypes of type α in the set F_{-i}, r is the total number of haplotypes in F_{-i}, and θ is a scaled mutation rate. Note that F_{-1} is updated to include the first haplotype sampled before generating the second haplotype. The mutation parameter determines the expected similarity between subsequent haplotypes in a sample; a larger θ produces a more diverse sample of haplotypes. For algorithm 1, the PIM model specifies the full conditional probability distribution up to a proportionality constant

$$\Pr(H_{ji} = h) = \frac{r_h + \theta v_h}{r + \theta},$$

where v_h is the probability that a mutant chromosome is of haplotype h. Although there is a rough correspondence of this prior to the distribution expected under certain population genetic models, the correspondence is not exact and the prior is heuristic, rather than explicitly model-based. In particular, the PIM prior does not explicitly model recombination which is an essential factor in any realistic model of the genesis of human haplotypes.

The HAPLOTYPER algorithm

Niu et al. (2002) developed a Bayesian haplotype inference algorithm that is closely related to the algorithm of Stephens et al. (2001) presented above. The main difference between the methods is the prior on haplotype frequencies, p. Niu et al. (2002) use a Dirichlet prior for p which has the form,

$$f(p_1, \ldots, p_n) = \Gamma \left(\sum_{i=1}^{n} \beta_i \right) \prod_{i=1}^{n} \frac{p_i^{b_i - 1}}{\Gamma(\beta_i)}, \text{ where } \sum_{i=1}^{n} p_i = 1, \qquad (3.9)$$

and $\Gamma(\cdot)$ denotes the Gamma function. The marginal expectation of the ith haplotype frequency is

$$\mathbb{E}(p_i) = \beta_i / \sum_j \beta_j,$$

where the parameters β can be adjusted to incorporate prior information regarding haplotype frequencies. If no prior information is available, one of several neutral Dirichlet parameterizations can be used

with differing levels of variance. The basic Gibbs sampling implementation of their algorithm simulates both the individual diplotypes, F, and population haplotype frequencies, p, from the full conditional probability distribution. The full conditional distribution of F_i given G_i and p is

$$\begin{aligned}\Pr(F_i|F_{-i}, G_i, p) &= \frac{\Pr(F_i|p)\Pr(G_i|F_i)}{\sum_{j=1}^{c_i}\Pr(F_i = D_j|p)\Pr(G_i|F_i = D_j)}, \\ &= \frac{\Pr(F_i|p)}{\sum_{j=1}^{c_i}\Pr(F_i = D_j|p)},\end{aligned}$$

where D_j is the jth element of the set of c_i haplotypes that are compatible with the ith individual genotype G_i. The term at right only allows diplotypes compatible with genotypes (eliminating the terms $\Pr(G_i|F_i = D_j)$). The full conditional probability distribution of p is the posterior probability density of p conditional on the haplotype configuration. Because the Dirichlet distribution is a conjugate prior (see Hartigan, 1983) for the multinomial distribution the posterior density has the same distributional form as the prior but with a change of parameters,

$$f(p_1, \ldots, p_n|F, G) = \Gamma\left(\sum_{i=1}^{n}\gamma_i\right)\prod_{i=1}^{n}\frac{p_i^{\gamma_i - 1}}{\Gamma(\gamma_i)},$$

where n is the total number of possible haplotypes compatible with the sampled genotypes and

$$\gamma_i = \beta_i + n_i[F],$$

where $n_i[F]$ is the count of the number of haplotypes of type i given the sample of diplotypes, F. This algorithm requires that a vector of population haplotype frequencies is maintained; as mentioned in reference to the EM algorithm this can demand a large amount of computer memory when many heterozygous SNP loci are present in a sample. Niu et al. (2002) therefore introduced a second version of the algorithm, the so-called "predictive updating" method, that is analogous to the procedure of Stephens et al. (2001) in which they analytically integrate over the haplotype frequency prior and only integrate over diplotypes using the Gibbs sampler. In that case, the full conditional probability is proportional to

$$\Pr(F_i = [g, h]|F_{-i}, G_i) \propto (n_g[F_{-i}] + \beta_g)(n_h[F_{-i}] + \beta_h),$$

where g and h are haplotypes compatible with G_i, $n_j[F_{-i}]$ denotes the

number of copies of haplotype j in the set of diplotypes F_{-i}, and β_j is the prior Dirichlet parameter for haplotype j. Niu et al. (2002) introduce other inovations to improve efficiency of the MCMC including a partition-ligation algorithm to improve mixing.

3.3.5 Performance of haplotyping algorithms

The likelihood of each possible diplotype configuration given the sample of multilocus genotypes is obtained from a multinomial sampling distribution (assuming HWE). This likelihood function underlies all probabilistic models used for deriving haplotype inference methods. Thus, the asymptotic statistical performance of all methods (EM algorithms, Bayesian inference, etc) should be similar. Any differences in statistical performance among methods will be due to the different priors they use for modeling population haplotype frequencies. In terms of computational efficiency, scalability, etc, the implementation will matter, however, and there may be great differences among methods. Most of the more recently proposed haplotype inference methods consider different priors on haplotype frequencies or use different MCMC implementations to try to improve computational efficiency.

Several recent simulation studies (Stephens and Donnelly, 2003; Zhang et al., 2006) have shown that population-based haplotype inferences can be highly accurate and that the differences in performance among the Bayesian and likelihood-based methods are relatively minor. For the Bayesian algorithms, if a particular method performs better in a given simulation study it is likely due to the fact that the prior used in the inference method is more similar to the model used to simulate the samples; this has limited relevance on the likely performance of the methods in analyzing empirical data since the models used to formulate priors are all quite unrealistic. Improved priors can be obtained by using a more realistic model of the process of lineage coalescence and recombination within populations (see section 4.3.2) although the computational burden is greatly increased.

3.4 Haplotype tagging SNPs

Several influential papers were published in 2001 promoting the use of haplotypes, rather than genotypes, for genome-wide disease association analysis (Patil et al., 2001; Johnson et al., 2001; Daly et al., 2001;

Reich et al., 2001). The studies provided empirical evidence (from large-scale genotyping of populations, family-based phase inference, and experimental haplotyping via hybrid cell lines) for the existence of a relatively small number of common haplotypes in humans. The main reason for this is that human populations have undergone a recent expansion and most haplotypes are therefore descended from a single chromosome that existed in the recent past (typically within about the last 200,000 years) and so few recurrent mutations have occurred on each chromosome. As well, it appears that few recombinations have occurred within distances smaller than about 30 kb in many regions of the genome (Reich et al., 2001). This recent origin for variation also explains why most human SNP loci have only two alleles segregating in the population.

One of the principle ideas driving recent haplotyping projects, such as the International HapMap project (Consortium, 2005), is that by cataloguing haplotype variants in human populations one can choose a sparser set of "haplotype tagging" SNPs (htSNPs) to carry out association studies and yet still capture all the important haplotypes present in the population that may be associated with common diseases. With this goal in mind, a large number of statistical procedures have been proposed for identifying htSNPs, and for carrying out association studies using htSNPs. Here, we present the most widely-used approaches and discuss their utility. As the technology is rapidly moving toward whole-genome resequencing in human disease studies, these techniques may well become obsolete within a decade or less.

3.4.1 Identifying redundant SNPs

Efforts to identify htSNPs have been largely data-driven and the methods heuristic. This has created much confusion and a plethora of proposed methods based on various ad hoc optimality criteria. However, the basic concept of a redundant SNP is very simple when viewed from a population genetics perspective. One or more SNPs located within a genomic region are redundant if they occur on the same branch of the ancestral recombination graph underlying that region for the population. The ancestral recombination graph (Griffiths and Marjoram, 1996) is a representation of the genealogical history of the population including both coalescence events (joining of lineages with a shared common ancestor) and recombination events (splitting of adjacent chromosomal segments into separate lineages). An ARG is illustrated in Fig-

ure ?. Clearly, if n mutations arise on a branch, then all chromosomes descended from the branch will carry the n mutations. Thus, any one of the n markers is sufficient to document the shared ancestry among chromosomes due to that branch and the remaining $n - 1$ markers are redundant. Strictly speaking, this notion of SNP redundancy is defined for the ARG underlying the entire population. For any given sample of chromosomes, the sample ARG will be a subgraph of the population ARG. This means that some SNPs that are redundant for the sample ARG will not be redundant for the population ARG. Moreover, sampling another set of chromosomes may lead to a different set of redundant SNPs from those defined for the first sample. Such issues limit the utility and generality of htSNPs identified using existing algorithms.

The distribution of haplotypes across the human genome is a product of ancestral processes of recombination, mutation, selection, genetic drift, etc. Although it is possible to model many of these processes using a parametric approach and theory from population genetics, the problem is computationally challenging and the field is still in its infancy. Ideally, algorithms for identifying htSNPs should do so by integrating over the unobserved ARG given the SNP data and inferring the probability distribution of mutation locations on branches. However, such approaches remain to be developed.

3.4.2 Haplotype block-finding algorithms

Haplotype block-finding algorithms attempt to identify genomic regions (referred to as "blocks") that contain extended haplotypes such that subsets of redundant markers that are highly correlated within a haplotype are identified. The basic premise is that a smaller number of markers representing these subsets can then be used as tagging SNPs without a significant loss of information.

Patil et al. (2001) stated the objective of their computational analysis as being to define contiguous non-overlapping blocks of SNP haplotypes covering the entire 32.4 Mb region of chromosome 21 that was surveyed. This was to be done in such a way so as to minimize the total number of SNPs needed to identify all the common haplotype blocks. They developed a so-called "greedy" algorithm for block-finding defined as follows: (1) enumerate all possible blocks comprised of one or more adjacent SNPs; (2) exclude all blocks for which fewer than 80% of chromosomes are defined by haplotypes with more than 1 copy in the sample; (3) from the remaining overlapping blocks, select the set of

non-overlapping block estimates that maximize the ratio,

$$\max\left(\frac{\text{total SNPs}}{\text{minimal SNPs needed}}\right),$$

where "total SNPs" is the total number of SNPs spanned by the block and "minimal SNPs needed" is the minimum number of SNP loci needed to unambiguously identify the block. This third step intentionally biases the block choices in favor of blocks that can be identified using fewer htSNPs. Once the blocks have been defined in this way, a subset of htSNPs are identified that are sufficient to unambiguously identify the blocks.

Daly et al. (2001) used a somewhat different strategy for identifying haplotype blocks. An index of haplotype diversity is first calculated for successive 5 SNP windows along the chromosome. The diversity index was defined as a ratio of the observed haplotype heterozygosity, H_{obs}, to the expected haplotype heterozygosity, H_{exp}, under HWE,

$$S_5 = H_{obs}/H_{exp},$$

where the observed heterozygosity is

$$H_{obs} = \frac{1}{n}\sum_{i=1}^{n} I(F_i).$$

Here, n is the number of individuals sampled, F_i is the diplotype of individual i, and

$$I(F_i) = \begin{cases} 1 & \text{if} \quad H_{i1} \neq H_{i2}, \\ 0 & \quad\ \text{otherwise}, \end{cases}$$

where H_{il} is the maternal (or paternal) haplotype $(l = 1, 2)$ of individual i. The expected heterozygosity, H_{exp}, is calculated as

$$H_{exp} = 1 - \prod_{j=1}^{5}\prod_{i=1}^{2} p_{ij}^2,$$

where p_{ij} is the inferred frequency of allele i at SNP locus j. Blocks are created by expanding windows with locally minimal S_5 scores. It is difficult to predict whether this ad hoc procedure will identify all biologically relevant blocks under any given criterion. As well, the choice of window size (5 SNPs) is arbitrary and different block structures may be obtained by varying the window size. Moreover, increasing the marker density in a study will also lead to a different assigment of blocks which

is not very desirable if one wishes to obtain a stable block structure for identifying htSNPs.

Reich et al. (2001) used the pairwise linkage disequilibrium measure D' (see section 3.2.1) as a metric for evaluating the physical extent of haplotype blocks in the human genome. They defined the "half-length" of LD to be the physical distance from an arbitrarily chosen core SNP at which average $|D'|$ (averaged over all pairwise comparisons between markers in the interval) drops below 0.5. Using this criterion, they found that 19 randomly selected regions had half-lengths of about 60 kb on average. They did not attempt to develop a formal criterion for identifying haplotype blocks or choosing markers to tag blocks, although in principle one can do this using a criterion based on neighborhoods of high average pairwise LD (see e.g., Gabriel et al., 2002; Zhang and Jin, 2003).

Zhang et al. (2002) developed a dynamic programming algorithm for haplotype block partitioning aimed at minimizing the number of htSNPs needed to identify most common haplotypes. Let $\mathbf{r} = \{r_{ik}\}$, where $r_{ik} = 0, 1, 2$ is the allele at the ith contiguous SNP locus of the kth haplotype, where 0 is missing data and 1 and 2 are the alternative SNP alleles. A haplotype block is defined by an initiating locus i and a terminating locus j, where $i < j$, so that the block is

$$\text{Block}(i, j) = \{r_i, r_{i+1}, \ldots, r_{j-1}, r_j\}.$$

Two haplotypes, h and g, are designated as "compatible" within a block region (i, j) if the alleles at loci with no missing data are identical,

$$r_{lh} = r_{lg} \text{ for any } i \leq l \leq j \text{ such that } r_{lh} \times r_{lg} \neq 0.$$

A haplotype is designated as "ambiguous" for a block region (i, j) if it is compatible with two other haplotypes that are themselves incompatible over the block region. Zhang et al. (2002) provide the following example of an ambiguous haplotype. Let the three haplotypes be

$$\begin{aligned} r_1 &= (1, 1, 0, 2), \\ r_2 &= (1, 1, 2, 0), \\ r_3 &= (1, 1, 1, 2). \end{aligned}$$

In this case, r_1 is compatible with both r_2 and r_3, but r_2 is incompatible with r_3 (because they have different alleles at locus 3) and therefore r_1 is ambiguous, whereas r_2 and r_3 are unambiguous. The ambiguous

haplotypes are discarded and the unambiguous haplotypes are partitioned into disjoint groups where two or more unambiguous compatible haplotypes are placed into the same group and henceforth treated as identical. The remaining steps in the algorithm are then applied to groups, rather than individual haplotypes. A non-overlapping (contiguous) block partition of the genomic region is

$$\mathbf{B} = \{B_1, \ldots, B_I\},$$

where $B_w = \text{Block}(i, j)$ and $B_{w+1} = \text{Block}(j+1, l)$, etc, and I is the total number of block partitions. A boolean (0,1) operator function is defined such that $\text{Bool}(B_i) = 1$ if at least α-percent of the unambiguous groups in the block are represented by more than one haplotype, and $\text{Bool}(B_i) = 0$ otherwise.

Let $f(\cdot)$ be a function of a haplotype block that specifies the minimum number of SNP loci needed to uniquely distinguish at least α percent of the unambiguous haplotype groups within the block; this is referred to as the number of "representative" SNPs for the block. For a particular block partition, the total number of representative SNPs is

$$T = \sum_{i=1}^{I} f(B_i).$$

A dynamic programming algorithm (see e.g., White, 1969) is then used to identify two quantities: (1) the block partition requiring the minimum number of representative SNPs (e.g., htSNPs); and (2) among the block partitions satisfying 1, the partition requiring the fewest blocks. The dynamic programming algorithm is applied recursively. The algorithm for inferring block partitions that minimize the number of representative SNPs is as follows. Let S_j be the number of representative SNPs for the optimal block partition of the first j SNPs. Set $S_0 = 0$ and implement the recursion,

$$S_j = \min\{S_{i-1} + f(r_i, \ldots, r_j), \text{ if } 1 \leq i \leq j \text{ and } \text{Bool}(r_i, \ldots, r_j) = 1\}.$$

Similarly, dynamic programming can be applied to a recursion over block number given S_j. Let C_j be the minimum number of blocks for all partitions with S_j representative SNPs and set $C_0 = 0$. The recursion is

$$\begin{aligned} C_j = \min\{&C_{i-1} + 1, \text{ if } 1 \leq i \leq j \text{ and } \text{Bool}(r_i, \ldots, r_j) = 1 \\ &\text{and } S_j = S_{i-1} + f(r_i, \ldots, r_j)\}. \end{aligned}$$

Zhang et al. (2002) implemented an efficient algorithm for identifying

blocks and htSNPs under these criteria. They applied their program to the data of Patil et al. (2001), requiring that at least 80% of unambiguous haplotypes are represented more than once in each block, and found that only 3,982 SNPs (out of 24,047 SNPs in total) and 1,884 blocks are required to explain 95% of haplotype diversity. By contrast, the greedy algorithm of Patil et al. (2001) required 4,563 SNPs and 4,135 blocks to capture the same diversity.

Wang et al. (2002) attempted to incorporate a more biological motivation for haplotype block identification as determined by historical recombination events. The basic idea is to use a four gamete test for recombination (see section 4.3.1) to identify historical recombinations defining boundaries between haplotype blocks. The algorithm begins at a particular SNP locus and adds one additional flanking SNP locus at each iteration. If the newly added locus results in 4 gametes then a block is terminated, otherwise it extends to the next locus, and so on. As mentioned in section 4.3.1, the 4-gamete test for recombination assumes an infinite sites mutation model (which is probably reasonable for human SNPs) and only determines that at least one recombination occurred. Also, the test will miss a certain proportion of recombinations that (by chance) do not lead to 4 gametes. More general approaches for inferring locations (and rates) of recombination are described in section 4.3.2.

Another approach for finding haplotype block boundaries uses the concept of Minimum Description Length (MDL) borrowed from the field of information theory. Several researchers have pursued the application of a MDL criteria to block-finding (Anderson and Novembre, 2003; Greenspan and Geiger, 2004, 2006). A principle focus of information theory has been the development of efficient coding strategies for transmitting information. Let $\mathbf{w} = \{w_1, \ldots, w_m\}$ be a set of words and let Σ be a set of D code symbols. Let $\mathbf{p} = \{p_1, \ldots, p_m\}$, where p_i is the probability that w_i as the next word transmitted in a message. The objective is to encode $\mathbf{w}$ using symbols Σ in the most economical way. Intuitively, we want to use shorter codes for more frequent words. In most situations, a binary code, $\Sigma = (0, 1)$, is used. For example, if the words are $=\{a, b, c\}$, and the probabilities are $p_a = 0.9$, $p_b = 0.05$ and $p_c = 0.05$, then one possible prefix code using Σ is,

$$a \rightarrow 0,\ b \rightarrow 10,\ c \rightarrow 11.$$

For example, the sentence abbcab would be encoded as 0101011010 using this coding scheme. This coding has an average (expected) word

length of

$$0.9 \times 1 + 0.05 \times 2 + 0.05 \times 2 = 1.1.$$

Shannon's Source Coding Theorem places a lower bound on the code length, L_C, which is the entropy of the probability distribution of words,

$$L_C \geq H(\mathbf{p}) = -\sum_{i=1}^{m} p_i \log_2 p_i.$$

It can be shown that a binary code exists with average word length no greater than $1 + H(\mathbf{p})$. The better a code is, the smaller is its average word length. For the example word distribution given above the upper bound on best average code word length is

$$1 + -(0.9 \log_2 0.9 + 0.05 \log_2 0.05 + 0.05 \log_2 0.05) = 1.57$$

The MDL extends the concept of an optimal coding (which is defined for a specified probability distribution of words) to comparisons of probability distributions as descriptions of data. Different probability distributions correspond to different minimum code lengths; a shorter code length indicates a better fit of the probability distribution. The true probability distribution is unknown for any given data set $\mathbf{x} = \{x_1, \ldots, x_n\}$ and so the parameters, θ, of a parametric probability distribution are estimated from the data; conceptually, this adds a cost to the coding – the sender transmits the coded data to the receiver and then transmits the parameters needed for decoding. This can be formalized as a penalty $L(\theta)$ associated with inference of the model parameters,

$$-\log f_\theta(\mathbf{x}) + L(\theta).$$

More complex models receive a heavier penalty much as they do under conventional schemes for comparing non-nested models. Two important aspects to consider in the formulation of MDL methods for particular applications are the family of probability models that are considered and the penalty for model complexity (parameter inference).

Anderson and Novembre (2003) modeled the correlation of haplotypes between blocks along a chromosome using a first order Markov chain. This is a heuristic model in that it does not faithfully represent the correlations of alleles along chromosomes due to evolutionary processes. If there are R blocks, the parameters of the model include the haplotype frequencies at the first block (at left) as well as a matrix of transition probabilities for each possible pairing of haplotypes between

adjacent blocks. Because not all haplotype blocks are correlated, submodels with block frequencies being proportional to the product of the marginal haplotype frequencies are also considered. The distribution of SNP alleles within blocks (the haplotype) is modeled as a product of independent Bernoulli random variables. Again, this model is chosen for its simplicity, not because it has any relationship to the biological processes generating haplotypes. The MDL criterion is then used to choose among models with different distributions of blocks. The so-called two stage coding scheme is used. The first stage involves calculating the code length for transmission of each of the model parameters (e.g., number of blocks, physical end-points of blocks, number of haplotypes in each block and transition probabilities among blocks and/or marginal haplotype frequencies for blocks). The second stage calculates the code length for transmission of the haplotypes in each block (the binary strings of alleles). The best model minimizes the total description length (the sum of code lengths from stages one and two). The total cost involves a trade-off between the cost of coding the model parameters (which increases with more complex models) and the cost of coding the data (which decreases with more complex models).

The authors simulated data under a population genetic model with recombination hotspots (regions with enhanced recombination rates R_h times greater than the background recombination rate) and compared the numbers of blocks inferred using the MDL-based method versus several other block-finding methods. In general the MDL method inferred fewer blocks than the other methods and was less likely to infer the presence of a block boundary in a region where no hotspot was located. However, as noted by the authors, it is difficult to evaluate whether a method is performing well, or poorly, based on the distribution of blocks because a "block" is not a parameter of the simulation model, it is a property of the block-finding statistic. Thus, comparisons among statistics are not easily interpreted

3.4.3 Block-free tagging SNP selection algorithms

Johnson et al. (2001) focus specifically on the identification of optimal htSNPs, rather than block-finding, using the following procedure: (1) order haplotypes according to similarity using ClustalX alignment software; (2) identify a htSNP subset of a given size that "best" captures the full haplotype information. The optimality criterion is based on a haplotype diversity index. The haplotype diversity is defined to be the

total number of differences observed in all $N \times N$ comparisons among N haplotypes in the sample. The proportion of diversity explained by the htSNPs is then defined as

$$P = 1 - \left(\frac{\text{Residual diversity}}{\text{Total diversity}} \right),$$

where "residual diversity" is the diversity between haplotypes within groups defined by the htSNPs. A set of htSNPs are chosen that maximize P.

Sebastiani et al. (2003) focused on identifying a minimal set of tagging SNPs necessary and sufficient to characterize all haplotypes in a sample. Following their notation, let $S = \{s_1, \ldots, s_m\}$ denote a vector of m SNP loci surveyed for a sample of haplotypes. Let $S = H \cup D$, where $H = \{h_j\}$ is the minimal tagging set of k SNPs (e.g., the smallest set of SNPs both necessary and sufficient to derive all SNPs in the haplotype set) and $D = \{d_j\}$ is the set of $(m - k)$ SNPs that are derivable from H. A SNP d_j is derivable if it can be expressed as a Boolean function of the elements of H. The minimal tagging set may not be unique as the following example shows.

Suppose that in a sample of 4 SNP loci, the following 3 distinct haplotypes are observed:

A - T - C - G
A - G - G - T
C - T - C - G,

where A is the allele present at locus 1 of haplotype 1, T is the allele present at locus 2 of haplotype 1, and so on. In this case, the alleles at loci 2, 3 and 4 are prefectly correlated. Thus, the minimal set of tagging SNPs would be two and would include the first locus and either of the remaining three loci. For example, if we choose $H = (1, 4)$ as the tagging SNPs then (A,G) would imply haplotype A-T-C-G, (A,T) would imply haplotype A-G-G-T and (C,G) would imply haplotype C-T-C-G, so two loci are both necessary and sufficient to identify all 3 haplotypes.

Sebastiani et al. (2003) describe an efficient algorithm for applying this criterion to large datasets. They reanalyzed the dataset of Johnson et al. (2001) and in several cases found smaller sets of minimal sufficient SNPs than were identified in the original paper. They also found that in most instances, multiple optimal sets of tag SNPs existed, sometimes as many as 30 different sets (e.g., for the CASP8 region).

There are several potential drawbacks to the procedures described

above for identifying SNP tags. First, the minimal set of tagging SNPs is determined for a specific sample and choice of cut-off for population haplotype frequency. If the sample size is increased, or an alternative random sample is analyzed, the set of minimal tagging SNPs could change because the choice is only optimal for the specific sample analyzed. How stable the tag SNPs will be depends on factors such as sample size and population substructure.

Second, the method requires that tag SNPs be 100% diagnostic to form a minimal set. There may exist a considerably smaller set of tag SNPs that will identify common haplotypes with high probability (although there may still be some low error rate). A probabilistic method, rather than a deterministic method such as those described above, can potentially use many fewer tag SNPs with little loss of accuracy. It can also take account of the fact that the tag SNPs are chosen based on estimated haplotype frequencies from a sample (rather than assuming knowledge of population haplotype frequencies) and is therefore more applicable to samples other than the specific sample from which the tag SNPs were chosen.

Meng et al. (2003) proposed a block-free method for choosing htSNPs based on multivariate analysis of pairwise linkage disequilibrium. The basic approach is to construct a matrix of pairwise LD coefficients among SNP loci and then analyze the LD matrix using spectral decomposition. The weighting (eigenvalues) obtained for different linear combinations of markers (eigenvectors) is used to choose an informative subset of markers that contribute most of the information concerning pairwise LD. This method appears to work less well than methods based directly on haplotypes, likely because dependence among markers due to extended haplotypes may not be efficiently captured by the use of pairwise LD coefficients alone. Lin and Altman (2004) proposed a similar method based on principle components analysis that used a correlation matrix among SNPs, rather than a matrix of LD coefficients. Presumably their method, which is based on pairwise correlations among SNPs, will also fail to adequately model dependence between SNP loci that arises from extended haplotypes.

3.5 Genotype imputation