# Table of Contents

# List of Tables

# List of Figures

**Abstract**

Recent studies of the human genome have uncovered a block-like pattern of SNP variation. Haplotype blocks are defined as chromosomal stretches in which a small number of multi-marker variants cover most of the observed variation. It is believed that haplotype blocks are generated by hotspots of recombination, which account for the vast majority of crossovers during meiosis.

We formulated a statistical model of haplotype block variation which takes account of recombinations, mutations and population genetic effects. Our model is based on a Bayesian Network with a Markov chain at its core. We developed two heuristic learning algorithms to infer instances of our model which are most suitable for observed haplotype, genotype or trio data.

Our model and learning algorithms were applied to three biological problems, with promising results. The first application is haplotype resolution, which infers pairs of haplotypes underlying a set of genotype observations. The second application is linkage disequilibrium (LD) mapping, which searches for a hidden genetic factor causing phenotypic variation. The third application is the inference of recombination structure from a set of raw genomic sequences.

We also addressed two key questions by examining high density data from the International Haplotype Mapping (HapMap) project. First, we confirmed the role of recombination hotspots in generating haplotype blocks, which has been the subject of much debate. Second, we showed that a Markov model over haplotype blocks is uniquely accurate for representing high density SNP variation.

Our statistical model and algorithms have been implemented as the HaploBlock software package, which is available online at `http://bioinfo.cs.technion.ac.il/haploblock/`.

# Common Abbreviations and Symbols

| | |
|---|---|
| bp | base pairs |
| CEPH | Centre d'Etude du Polymorphisme Humain |
| DL | description length |
| DNA | deoxyribonucleic acid |
| EM | expectation maximization |
| HapMap | International Haplotype Mapping Project |
| htSNP | haplotype tagging SNPs |
| kb | kilobases |
| LD | linkage disequilibrium |
| MAP | maximum a priori |
| Mb | megabases |
| MCMC | Monte Carlo Markov chain |
| MDL | minimum description length |
| PCR | polymerase chain reaction |
| RNA | ribonucleic acid |
| SNP | single nucleotide polymorphism |

| | |
|---|---|
| $\hat{a}_{k,c}$ | sequence of ancestor $c$ of block $k$ |
| $\mathcal{D}$ | haplotype and/or genotype data |
| $e_k$ | index of last SNP in block $k$ |
| $\mathcal{G}$ | genotype data |
| $\mathcal{H}$ | haplotype data |
| $M$ | statistical model |
| $\mathcal{M}$ | ensemble of statistical models |
| $\mathcal{P}$ | phenotype data |
| $q_k$ | number of ancestors of block $k$ |
| $s_k$ | index of first SNP in block $k$ |
| $\theta_{1,c}$ | probability of ancestor $c$ in first block |
| $\theta_{k,c' \to c}$ | probability of ancestor $c$ in block $k$ given $c'$ in block $k-1$ |
| $\mu_{j,a \to h}$ | probability of ancestral allele $a$ mutating to $h$ at site $j$ |
| $\Psi_M$ | statistical model parameters |

# Chapter 1

# Introduction

Genetic mapping is the task of discovering the genetic differences which affect susceptibility to a particular disease. Finding these differences is the first step towards understanding the biological mechanism which is malfunctioning in those suffering from the disease. Suitable drugs can then be developed which compensate by performing the function required. It is also hoped that some diseases will be cured by direct modification of their underlying genetic causes, without the need for ongoing treatment and medication.

A *phenotype* is defined as the properties which an individual exhibits in relation to a disease under study. For example, high blood pressure and cholesterol are two phenotypes which are related to heart disease. The process of genetic mapping begins by collecting a set of individuals who exhibit variation in the phenotype, and defining a genomic region which is suspected to cause this variation. The study focuses on a set of points called *markers* in the region, at which there is known to be genetic variability.

The mapping study proceeds by measuring the genetic variant (*allele*) present at each marker in the individuals studied. A statistical technique is then applied to correlate these measurements with the phenotypic differences. If the individuals are related by a known family pedigree, a technique called *linkage analysis* is applied, which is based on a model of recombination and inheritance. If the individuals are not related, a different technique called *linkage disequilibrium (LD) mapping* is used, which looks for direct correlations in the data observed.

The statistical analysis identifies the markers which are closest to the genetic cause of the phenotypic differences, so that the region of interest is narrowed. In an early-stage mapping study, this region might cover an entire chromosome, with markers spread far apart. Later on, the region might contain just a few genes, with markers spaced every few kilobases. When the region is sufficiently small, it can be completely resequenced in each individual chromosome, allowing the genetic cause to be identified by direct comparison.

Several different types of marker are available for mapping in humans. For example, microsatellite markers contain a short nucleotide sequence which is repeated a different number of times in different chromosomes. Until recently microsatellites were commonly used for genetic mapping due to their high variability and ease of measurement. However, with the advent of the Human Genome Project, researchers have now focused on single nucleotide polymorphism (SNP) markers, at which chromosomes differ by a single base pair [121]. SNP markers are far more common than other types, and so pave the way for mapping studies at a very high resolution. For example, the January 2005 build of the dbSNP database contains information on 9,348,745 human SNP markers, leading to an average spacing of one SNP marker per 320 base pairs in the genome [119].

The *haplotype* for a set of markers is defined as the sequence of alleles present at those markers in a single chromosome. Many recent studies of human genetic variation have demonstrated the presence of *haplotype blocks*, defined as regions in which a small number of multi-marker haplotypes cover the observed variation [16, 94, 29, 33, 140, 139]. The low level of variation within haplotype blocks can be explained by bottleneck effects and genetic drift. Bottlenecks occur when a local population is descended from a small group of individuals, for example due to migration or strong selection, resulting in a sharp reduction in genetic variation. Genetic drift refers to the gradual decrease in variation due to repeated random sampling of the alleles in a population from those in the previous generation. Since genetic drift is strongest when a population is small, the early generations following a bottleneck event will undergo the greatest reduction in diversity, leaving behind a small number of ancestral haplotypes upon which the future population is built.

The presence of haplotype blocks in the genome has many implications for genetic mapping. As a result, many different computational methods have been developed to infer haplotype blocks from raw SNP data. Daly *et al.* [16] identify stretches which have significantly less heterogeneity than would be expected considering the frequencies of the constituent SNPs. Patil *et al.* [94] and Zhang *et al.* [149, 151] examine the ratio between the number of SNPs in a region and the size of the smallest subset of these which is sufficient to uniquely identify all of its haplotypes. Gabriel *et al.* [29] look for areas within which the allelic correlation between most pairs of SNPs is high. Wang *et al.* look for regions in which no two sites exhibit all four possible combinations of alleles

[141]. Several different statistical approaches have also been applied to this problem, with varying degrees of similarity to our work [2, 59, 56].

We developed a new statistical model of haplotype block variation, suitable for high density SNP data. Chapter 2 describes the model using a Bayesian Network and explains how its parameters relate to the underlying biological processes. This chapter also explains how the model is extended to represent marker information from haplotype pairs or father-mother-child trios. Chapter 3 describes the criterion we use for assessing how well a particular model fits a set of observed data, and explains our algorithm for optimizing this criterion over the space of possible models.

Our use of a statistical model enables a large range of problems to be addressed by querying the model in different ways. Consequently, our work has broader application than most existing block identification methods. One such application is *haplotype resolution*, a problem that has been extensively studied in the literature. In diploid organisms such as humans, ordinary cells contain two copies of each chromosome, one of which was inherited from each parent. Standard measuring processes examine both chromosomes simultaneously, yielding an unordered pair of alleles for each marker. The *genotype* for a set of markers is defined as a series of such measurements, with no information on which alleles are co-located on the same chromosome. Molecular laboratory techniques to measure chromosomal haplotypes have been developed but their cost remains prohibitive in many cases [84, 144, 72, 20, 134, 19]. As a result, in silico *haplotype resolution* is performed to infer the haplotypes underlying the genotype observations for a group of individuals, based on some assumptions about how populations behave. Chapter 4 shows how our model is applied to the haplotype resolution problem, obtaining a high degree of accuracy in comparison with several well-known approaches.

The greatest potential impact of our work is in linkage disequilibrium (LD) mapping. The standard approach to LD mapping looks for correlations between phenotypic status and the alleles at each marker. The haplotype block structure of a chromosomal region can be incorporated into this method, by testing the haplotypes of each block for correlations instead of individual SNPs. This has the potential to dramatically increase the chance of detecting associations and reduce the probability of false positives [7, 133, 11]. Chapter 5 shows how our statistical model is applied for block-based LD mapping with haplotype or genotype marker data, enabling a reduction in the resequencing required.

Our model and algorithms were developed for the purpose of analyzing human SNP data. However with minor modifications the same techniques can be used to analyze raw genomic sequence data from other organisms to detect and characterize uneven recombination structure. Chapter 6 describes such an analysis performed on two sets of viruses – Kaposi's sarcoma-associated herpesvirus, in which recombination is suspected to play a role in creating diversity, and some newly-identified oceanic picornaviruses, for which the mechanism for generating diversity is unknown. This chapter shows how the parameters learned for a model allow inferences about selection pressures, mutation rates and recombination structure.

It is widely believed that haplotype blocks are created by *recombination hotspots*, defined as small genomic regions in which the probability of recombination is far higher than in the surrounding area [127, 112, 135, 101, 141, 4]. Since recombination is rare in the area between hotspots, the SNPs within segregate together from one generation to the next, acting as a multi-marker block allele. For a few loci, sperm-typing measurements from Jeffreys and others have shown that hotspots exist and explain some block boundaries [48, 49]. However some recent studies suggest that block boundaries can arise simply as a result of genetic drift [147, 98]. Chapter 7 analyzes a rich SNP data set in order to address this question, and shows a highly significant correlation between recombination probability and haplotype block boundaries.

The core of our model is a Markov chain, which expresses the haplotype block distribution of a population in terms of the pairwise correlations between adjacent blocks. This distinguishes our work from most other approaches to block identification, which consider the distribution for each block independently. Chapter 8 examines the Markov property in depth, showing that a Markov model of haplotype blocks provides a uniquely accurate way to model high density marker data.

This chapter also compares the performance of blocks and individual SNPs under the Markov and independent models and provides a theoretical explanation for the properties observed.

The empirical studies in Chapters 4, 5, 7 and 8 are based on four sets of publicly available human SNP data. Rieder *et al.* studied the gene ACE located on chromosome 17, thought to be related to cardiovascular disease, examining variation at 52 biallelic markers which extend over a genomic region of 24 kb [107]. They obtained 22 haplotypes from 11 subjects using allele-specific PCR to ensure that ambiguous genotypes were resolved correctly [84]. Daly *et al.* examined the variation in the 5q31 region of chromosome 5, as part of a study on the IBD5 locus related to Crohn's disease, examining variation at 103 SNPs over 500 kb [16]. A total of 258 transmitted and 258 untransmitted haplotypes were obtained from 129 trios in a European-derived population. Patil *et al.* undertook a full study of chromosome 21, examining variation at 24,047 SNPs over a total length of 21.7 Mb [94]. They obtained 20 haplotypes from 10 subjects by separating the two copies of each subject's chromosome using a somatic cell hybrid technique [20]. Finally, the International Haplotype Mapping Project (HapMap) recently began producing a high density haplotype map of the entire human genome [44]. The October 2004 data release of the HapMap included data for 693,114 SNPs spread over all 22 autosomes for 30 European trios.

The field of linkage disequilbrium mapping remains in its infancy, since high density genotypes over thousands of SNPs have only recently become feasible to obtain. Chapter 9 briefly discusses some of the ways in which our work might be improved or extended in future, now that a sufficient quantity of data is becoming available.

Our algorithms for model inference, haplotype resolution and linkage disequilibrium mapping have been implemented as the HaploBlock software package. HaploBlock provides many other functions, such as generating a model by simulation, generating data from a model, comparing models, and so on. HaploBlock is written in ANSI C code and is freely available as a command-line executable for Linux, Mac OS X and Sun OS. Appendix A is based on the HaploBlock user manual and describes its features and operation in detail.

# Chapter 2

# Data Modeling

# Introduction

The core of our work is a statistical model for representing high density SNP data in a genomic region. Our model contains both observed variables to represent the haplotype data and hidden variables to represent the processes which generated the haplotypes. The model is based on a Bayesian Network, which makes the dependencies and independencies between these variables explicit, allowing probability calculations to be performed efficiently.

Section 2.1 describes the model in detail, and provides some background on Bayesian Networks. This section also outlines some of the biological assumptions which underlie the model's design. Section 2.2 explains how we calculate the probability of a set of observed data given an instance of our model. For haplotype data, this calculation is simple and flows naturally from the model's definition. For genotype data, we use an extended model and sum over the different possible haplotype pairs which are compatible with the observations. Finally, Section 2.3 briefly compares our statistical model with two others that were recently published.

Different versions of this material were included in several publications [34, 36, 38, 35].

## 2.1 Statistical Model

Our model for the haplotype block variation in a genomic region is defined by: (a) a partition of the region into blocks, (b) one or more ancestor haplotypes for each block, (c) a Markov chain over the blocks defining the ancestor distributions and (d) site-specific mutation rates reflecting the mutations accumulated since the ancestors were alive.

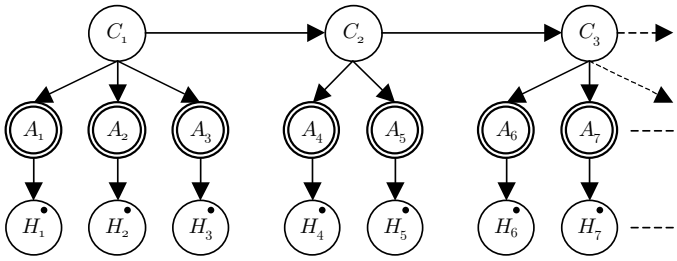
We partition a genomic region containing $l$ SNPs into adjacent and contiguous blocks, numbered $1 \ldots b$, with the indices of the first and last SNP of block $k$ defined by $s_k$ and $e_k$ respectively. Each block $k$ has $q_k$ ancestor haplotypes, numbered $1 \ldots q_k$. The sequence of ancestor haplotype $c$ of block $k$ is given by $\hat{a}_{k,c}$, a string of $e_k - s_k + 1$ symbols from the set $B = \{A, C, G, T, -\}$ of SNP alleles, which contains the four nucleic acids and a deletion. The probability that an observed haplotype is descended from ancestor $c$ in the first block is defined by the parameter $\theta_{1,c}$. For subsequent blocks, $\theta_{k,c' \to c}$ defines the probability that a haplotype is descended from ancestor $c$ in block $k$, given that it is descended from ancestor $c'$ in the previous block $k-1$. The mutation parameter $\mu_{j,a \to h}$ denotes the probability that ancestral allele $a$ at site $j$ is observed today as allele $h$.

The joint distribution defined by our model can be concisely depicted using a Bayesian Network. A Bayesian Network is a directed acyclic graph, where each vertex $v = 1 \ldots n$ corresponds to a discrete variable $X_v$ [95, 50]. The distribution for each variable $X_v$ is conditional upon the variables in $Pa_v$, which is defined as the set of vertices from which there are edges leading to $v$ in the graph. The joint probability of a full assignment $x_1, \ldots, x_n$ to variables $X_1, \ldots, X_n$ is the product of these conditional probabilities. Thus, $Pr(X_1 = x_1, \ldots, X_n = x_n) = \prod_v Pr(X_v = x_v | Pa_v = pa_v)$, where $pa_v$ is the joint assignment $\{x_i | X_i \in Pa_v\}$ to the variables in $Pa_v$. We will use the notation $Pr(y|z)$ as an abbreviated form of $Pr(Y = y | Z = z)$ for any sets of variables $Y$ and $Z$. For example, the joint probability could be rewritten as $Pr(x_1, \ldots, x_n) = \prod_v Pr(x_v | pa_v)$.

An important query is to compute the probability of a partial assignment $x_s$ to variables $X_s \subseteq \{X_1, \ldots, X_n\}$. This is defined as the sum of $Pr(x_1, \ldots, x_n)$ over all full assignments $x_1, \ldots, x_n$ which are compatible with $x_s$, i.e. $Pr(x_s) = \sum_{x_1} \cdots \sum_{x_n} Pr(x_1, \ldots, x_n | x_s)$. The independence assumptions embedded in the Bayesian Network allow such computations to be performed efficiently, for example by bucket variable elimination, a technique applied extensively in our work [17]. Also, suitable parameters for the conditional distributions in a Bayesian Network can be learned from observed data by the Expectation Maximization (EM) algorithm, which we use extensively as described in Chapter 3 [68].

The Bayesian Network corresponding to an instance of our model is shown in Figure 2.1. It contains a random variable $C_k$ for each block $k = 1 \ldots b$ and two random variables $A_j$ and $H_j$ for

Figure 2.1: Bayesian Network for haplotype data




each SNP $j = 1 \dots l$. Each variable $C_k$ defines the ancestor from which a haplotype is descended in block $k$. For the first block, $Pr(C_1 = c) = \theta_{1,c}$ and for subsequent blocks, $Pr(C_k = c|C_{k-1} = c') = \theta_{k,c' \to c}$. For each block $k$, variables $A_{s_k} \dots A_{e_k}$ define the sequence of the ancestor indicated by the value of $C_k$. For SNP $j$ in block $k$, $Pr(A_j = a|C_k = c) = 1$ if $\hat{a}_{k,c,j} = a$ and 0 otherwise. Variables $H_1 \dots H_l$ define the observed haplotype data over loci $1 \dots l$, where $Pr(H_j = h|A_j = a) = \mu_{j,a \to h}$ for each SNP $j$. The double borders in Figure 5.1 denote that variables $A_j$ are deterministic and the black dots indicate that variables $H_j$ are observed. On this point, it is worth noting the similarities between our model and a Hidden Markov Model (HMM), since in each case there is a Markov chain of distributions over unobserved variables upon which the observed data is conditional [32].

An assignment of values to the variables in the Bayesian Network reflects the history of a single observed haplotype. The value of each variable $C_k$ is the index of the ancestor for block $k$ from which the observed haplotype is descended. The sequence of that ancestor is specified by the values of $A_{s_k} \dots A_{e_k}$, where $A_{s_k}$ and $A_{e_k}$ are the first and last variables descended from $C_k$ respectively. The observed haplotype is specified by the values of variables $H_1 \dots H_l$. Clearly, $H_j = A_j$ unless a mutation has taken place at site $j$ in one of the generations since the ancestor was alive.

Let $\delta(x, y) = 1$ if $x = y$ and 0 otherwise. The Bayesian Network defines the joint distribution $Pr(c_1, \dots, c_b, a_1, \dots, a_l, h_1, \dots h_l)$ as:

$$\theta_{1,c_1} \prod_{k=2}^{b} \theta_{k,c_{k-1} \to c_k} \prod_{k=1}^{b} \prod_{j=s_k}^{e_k} \delta(\hat{a}_{k,c_k,j}, a_j) \cdot \mu_{j,a_j \to h_j} \tag{2.1}$$

Many biological assumptions underlie our model's design. Most fundamentally, we assume our population is in Hardy-Weinberg equilibrium, so we define our distribution over individual haplotypes instead of genotypes [45]. The model represents a series of multiple star genealogies, one for each haplotype block. Each block ancestor corresponds to the center of one star, while the haplotypes descended from that ancestor correspond to the star's points. The parameter independence of each conditional distribution $Pr(A_j|C_k)$ lifts all constraints on the phylogenetic relationship between each block's ancestors, since we are only interested in tracing ancestry as far back as the formative bottleneck event.

The Markov chain expresses the dependencies between the block genealogies, reflecting the fact that linkage disequilibrium exists between blocks as well as within them. The Markov chain implies that the probability of a haplotype being descended from a particular ancestor for block $k$ depends on its ancestor for block $k - 1$, an assumption which we examine in depth in Chapter 8. The parameter independence of each conditional distribution $Pr(H_j|A_j)$ allows for both site- and allele-specific mutation rates, justified by evidence for mutation hotspots [127, 28]. The values of $q_k$ for each block $k$ are allowed to differ, since the processes of drift and selection can act independently on each block.

The mutation rates in a model are constrained in several ways. First, if either $a$ or $h$ are not observed alleles of site $j$, we fix $\mu_{j,a \to h} = 0$, since such mutations are assumed either never to occur or to be deleterious. For other alleles $a \neq h$, the range of possible mutation rates is set by

parameters $\mu_{min}$ and $\mu_{max}$, so that $\mu_{min} \leq \mu_{j,a\to h} \leq \mu_{max}$. The values of $\mu_{min}$ and $\mu_{max}$ should ideally be based on the mutability and history of the chromosomal region being studied. However, since we generally lack such knowledge, suitable guideline values are $\mu_{min} = 10^{-6}$ and $\mu_{max} = 10^{-3}$, based on mutation rates of $1.6 \times 10^{-7}$ to $5.5 \times 10^{-9}$ per generation, a generation length of 20 years and a most recent bottleneck event between 100,000 and 5,000 years ago [90, 60, 120, 63].

The Markov chain parameters $\theta$ determine some additional values of interest. For the first block, the prior distribution for each ancestor $c$ is clearly given by $\pi_{1,c} = \theta_{1,c}$. For subsequent blocks $k > 1$, we obtain the prior distribution from that of the previous block and the transition parameters, where $\pi_{k,c} = \sum_{c'} \left(\pi_{k-1,c'} \cdot \theta_{k,c'\to c}\right)$. The conditional entropy $\xi_{(k-1)\to k}$ across each recombination hotspot provides a measure of the degree of recombination between blocks $k-1$ and $k$ and is given by $\xi_{(k-1)\to k} = -\sum_{c'} \pi_{k-1,c'} \sum_{c} (\theta_{k,c'\to c} \cdot \log \theta_{k,c'\to c})$.

## 2.2 Likelihood Calculations

### 2.2.1 Haplotype Data

Under a particular model $M$ with parameters $\Psi_M = (\hat{a}, \theta, \mu)$, the likelihood $Pr(h|M, \Psi_M)$ of a haplotype $h = h_1, \ldots, h_l$ is obtained by calculating the probability of the corresponding partial assignment in the Bayesian Network. This is given by the summation of the joint probability function over all variables which have not been assigned, i.e. $Pr(h_1, \ldots, h_l|M, \Psi_M) = \sum_{c_1} \cdots \sum_{c_b} \sum_{a_1} \cdots \sum_{a_l} Pr(c_1, \ldots, c_b, a_1, \ldots, a_l, h_1, \ldots h_l|M, \Psi_M)$, calculated efficiently by bucket variable elimination [17]. In some cases, we lack observations for particular sites due to failed measurements in the laboratory, in which case the variables $H_j$ corresponding to those sites are unassigned and included in the summation.

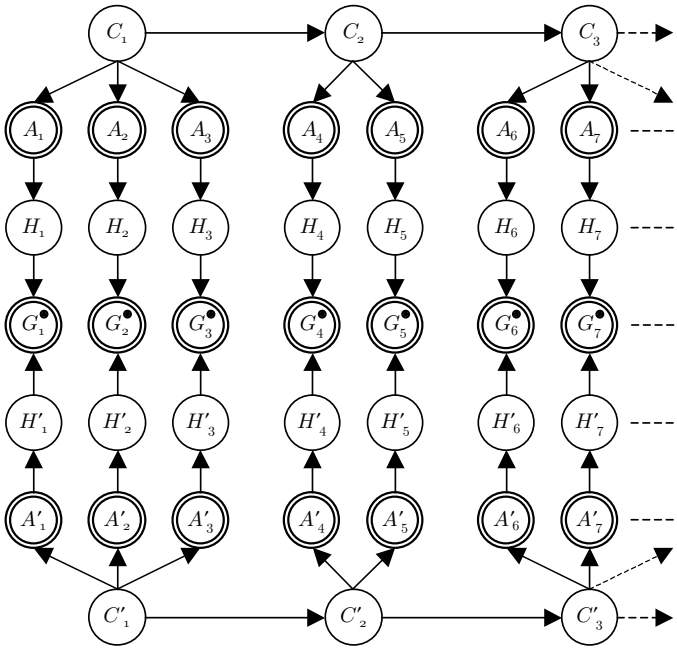
### 2.2.2 Genotype Data

The likelihood $Pr(g|M, \Psi_M)$ of a genotype $g = g_1, \ldots, g_l$ is calculated using the Bayesian Network shown in Figure 2.2. This contains two identical copies of the corresponding haplotype Bayesian Network in Figure 2.1 for $M$ and $\Psi_M$, where the mirrored copy has variables renamed to $C'_k$, $A'_j$ and $H'_j$. This defines the probability of a genotype as the product of the probabilities of its two constituent haplotypes, following our assumption of Hardy-Weinberg equilibrium. The new discrete variables $G_j$ in Figure 2.2 represent the joint genotype observations at each site $j$. As with the haplotype model, we calculate the likelihood $Pr(g_1, \ldots, g_l|M, \Psi_M)$ by summing the joint probability $Pr(c_1, c'_1, \ldots, c_b, c'_b, a_1, a'_1, \ldots, a_l, a'_l, h_1, h'_1, \ldots h_l, h'_l, g_1, \ldots, g_l|M, \Psi_M)$ over all unassigned variables.

Each variable $G_j$ takes values from the set $D$ of possible unordered pairs of SNP alleles, given by $D = \{[b_1, b_2] : b_1, b_2 \in B\}$. The conditional distribution for each $G_j$ is deterministic, since it is fixed by the alleles present on each chromosome at site $j$, i.e. $Pr(g_j|h_j, h'_j) = 1$ if $g_j = [h_j, h'_j]$ and 0 otherwise. If the genotype site $g_j$ is unknown, we define its conditional probability as 1 given any $h_j$ and $h'_j$. If $g_j$ is partially unknown, for example if only one allele was successfully measured, we assign its conditional probability according to the combinations of haplotype alleles which are compatible. For example, the conditional probability for the allele pair $[T, ?]$ is defined by $Pr([T, ?]|h_j, h'_j) = 1$ if $h_j = T$ or $h'_j = T$ and 0 otherwise.

The calculations in the genotype network in Figure 2.2 are quadratic in terms of the number of alleles permitted, since summations for each site $j$ must be performed simultaneously over both $H_j$ and $H'_j$. Recall that variables $H_j$ and $H'_j$ take values from the set $B$ of possible alleles, with a cardinality of $|B| = 5$. However the vast majority of SNP data is biallelic by nature, so only two of these alleles are required. We therefore use a simplified model which only allows two alleles, mapped to the observed alleles at each site as appropriate. This optimization was implemented in the biallelic version of the HaploBlock software package, as described in Appendix A.

Figure 2.2: Bayesian Network for genotype data



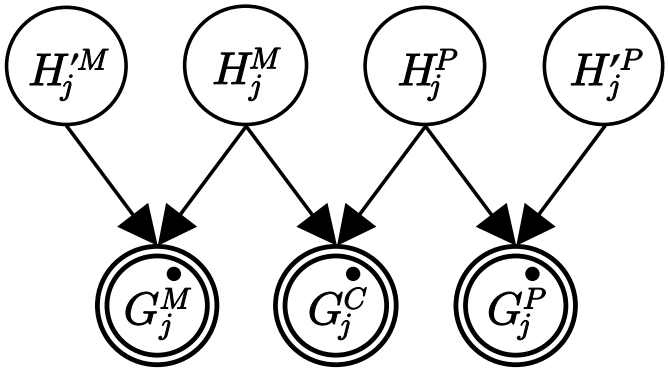

### 2.2.3 Trio Data

Let $(g^m, g^p, g^c)$ denote a set of trio genotypes, where $g^m = g^m_1, \ldots, g^m_l$ is the maternal genotype, $g^p = g^p_1, \ldots, g^p_l$ is the paternal genotype and $g^c = g^c_1, \ldots, g^c_l$ is the child genotype. As before, each genotype site takes values from the set $D$ of possible unordered pairs of SNP alleles. We define the probability $Pr(g^m, g^p, g^c|M, \Psi_M)$ of the trio as the product of the probabilities of the transmitted and untransmitted haplotypes of each of the two parents. In other words, we consider the two haplotypes transmitted to the child, as well as the two other haplotypes that would have formed the genetically complementary child. These four haplotypes contain the same genetic material as the parent haplotypes, possibly rearranged as a result of recombination.

The evidence from the trio is applied to two copies of the genotype network in Figure 2.2, one for each parent. The genotypes $g^m$ and $g^p$ are used to fix the values of variables $G_j$ in each model respectively, as with ordinary genotype data. We then use the technique described below to infer as much as possible of the maternal transmitted haplotype $h^m$ and untransmitted haplotype $h'^m$, as well as the corresponding paternal haplotypes $h^p$ and $h'^p$. At sites $j$ where these haplotypes could be determined, they fix the values of variables $H_j$ and $H'_j$ in the appropriate model. Where the haplotypes could not be determined, the respective variables are left unassigned and so included in the summation for the data likelihood. This can occur at sites where (a) all three individuals in the trio are heterozygous, (b) a Mendelian inheritance error is detected, (c) some or all of the genotypes are unknown.

We infer the parent haplotypes using the Bayesian Network shown in Figure 2.3. This model represents the inheritance relationships within the trio at site $j$, as well as the dependencies between the hidden haplotypes and the genotypes observed in each individual. Variable $H^M_j$ represents the transmitted maternal haplotype allele at site $j$ and $H'^M_j$ represents the corresponding untransmitted allele. Variables $H^P_j$ and $H'^P_j$ represent the respective paternal haplotype alleles. Each haplotype variable takes values from the set $B$ of SNP alleles. Variables $G^M_j$, $G^P_j$ and $G^C_j$ represent the maternal, paternal and child genotypes observed at site $j$, assigned respectively from the trio genotypes $g^m_j$, $g^p_j$ and $g^c_j$. The conditional distribution for each genotype variable is deterministic, given by same formula as for variables $G_j$ in Figure 2.2. For example, $Pr(g^C_j|h^M_j, H^P_j) = 1$ if

Figure 2.3: Bayesian Network for resolving trio data




$g_j^C = [h_j^M, h_j^P]$ and 0 otherwise. We infer the haplotype alleles from this model by calculating the posterior probability of each value of each haplotype variable given the genotype evidence, by bucket variable elimination. If any haplotype allele has a posterior probability of 1 given the evidence, it is considered fixed, otherwise it is left as unknown. We used this process instead of a more standard rule-based approach since the latter requires an unwieldy number of rules for all possible combinations of unknown and partially unknown measurements.

## 2.3 Discussion

Two other MDL approaches to modeling haplotype block variation have recently been published. Koivisto *et al.* [59] identify up to 10 haplotype clusters within each block using $k$-means clustering. Each haplotype cluster defines an independent distribution for the alleles at each SNP, with no constraint on the distribution's parameters. This contrasts with our ancestor plus mutation model, which expresses the variation within each cluster as the result of mutations since a founding bottleneck event. Koivisto *et al.* consider the ancestry for each block independently, allowing the optimal partition to be identified using dynamic programming [149]. In the language of our approach, their model conflates variables $A_j$ and $H_j$ in Figure 2.1 and removes the Markov chain connecting variables $C_k$.

Anderson and Novembre apply a different model, in which they enumerate the different haplotypes observed within each block without clustering by similarity or ancestry [2]. As in our technique, Anderson and Novembre represent the dependencies between adjacent haplotype blocks using a Markov chain. However, since their enumeration approach is liable to identify a large number of different haplotypes for each block, they save space in their model description by storing only selected parameters of this chain, setting the probability of the other haplotypes according to their marginal frequencies. Anderson and Novembre extend the dynamic programming algorithm of Zhang *et al.* to infer a globally optimal block partition in the presence of dependencies between adjacent blocks [149].

One clear advantage of our statistical model over these others is its ability to represent unphased genotype data, allowing it to be applied in the absence of phasing information. Another of its strengths is that missing data is dealt with naturally within the Bayesian Network framework, by summing over the variables for loci that are not observed. Both Koivisto *et al.* and Anderson and Novembre use dynamic programming to infer a single globally optimal partition for a genomic region. By contrast, we infer an ensemble of locally optimal models to allow for the ambiguity of block partitioning, as described in Chapter 3. Further research is required to determine which of these approaches is more fruitful.

# Chapter 3

# Model Learning

## Introduction

This chapter describes how we infer one or more statistical models from a set of observed data. This task can be formulated as a classical optimization problem, with three elements: (a) a definition of the search space, (b) a scoring function for each point in that space, and (c) an algorithm to search for points in the space with optimal score.

In our case, the search space is defined by the set of possible models, as described in Chapter 2. Clearly, for any non-trivial input, the set of possible models is vast – to begin with, there are $2^{l-1}$ possible block partitions for $l$ loci. Our scoring function assesses the suitability of a particular model for the observed data, using the minimum description length (MDL) criterion. Section 3.1 describes this criterion, which takes account of both the model's complexity and the probability of the data under the model.

We developed two related search algorithms. Section 3.2.1 describes the first algorithm, which infers a single model to explain the observed data using a heuristic search strategy. However recent research has suggested that it is over-simplistic to assume that a single 'true' block partition can be identified for a genomic region, due to the complexity of the patterns generated by recombination and mutation [113, 5, 143]. Section 3.2.2 describes our second algorithm, which expands the search strategy to infer an ensemble of models to explain the data observed.

Recall from Chapter 2 that the structure of the Bayesian Network for a statistical model is defined by the model's block partition, as well as the ancestor count $q_k$ for each block $k = 1 \ldots b$. Section 3.3 describes the steps used by our learning algorithms to explore these possible structures. Given a specific Bayesian Network structure, the conditional distributions of the variables within are defined by the ancestor haplotypes $\hat{a}_k$, Markov chain probabilities $\theta_k$, and cumulative mutation rates $\mu_j$ for each site $j = 1 \ldots l$. Section 3.4 describes how these parameters are inferred from the set of observed data by a number of different EM algorithms.

A shortened version of this material was presented at RECOMB 2003 and published in the *Journal of Computational Biology* [34, 36].

## 3.1 MDL Criterion

Our input data consists of a set of haplotype observations $\mathcal{H}$ and/or genotype observations $\mathcal{G}$. Input data consisting of trio observations is converted to pairs of genotypes with some additional haplotype constraints (see Section 2.2.3). Assuming independence, the likelihood $Pr(\mathcal{H}, \mathcal{G}|M, \Psi_M)$ of $\mathcal{H}$ and $\mathcal{G}$ under model $M$ with parameters $\Psi_M$ is given by $\prod_{h \in \mathcal{H}} Pr(h|M, \Psi_M) \prod_{g \in \mathcal{G}} Pr(g|M, \Psi_M)$, where the likelihoods $Pr(h|M, \Psi_M)$ and $Pr(g|M, \Psi_M)$ are calculated as in Section 2.2.

Seeking a model which maximizes this likelihood leads to over-fitting, since any observed distribution is reproduced exactly by a simple model with many ancestors and no recombination or mutation. We therefore use the minimum description length (MDL) criterion, which penalizes models according to their complexity. The MDL criterion seeks to minimize the total number of bits required to represent data with a model, akin to finding the data's optimal compressed encoding [109, 43]. If $DL(M, \Psi_M)$ bits are required to represent a model $M$ with parameters $\Psi_M$, then the total description length for data $\mathcal{D} = (\mathcal{H}, \mathcal{G})$ using the model is $DL(\mathcal{D}, M, \Psi_M) = DL(M, \Psi_M) + DL(\mathcal{D}|M, \Psi_M)$, where $DL(\mathcal{D}|M, \Psi_M) = -\log_2 Pr(\mathcal{D}|M, \Psi_M)$. For general Bayesian Networks, the Bayesian Information Criterion (BIC) can be used to calculate $DL(M, \Psi_M)$ but we diverge somewhat from that formulation here [114].

Formally, the description length $DL(M, \Psi_M)$ is the number of bits required to represent $M$ and $\Psi_M$ with optimal efficiency. When comparing different models or parameters, we can ignore elements of this description whose lengths are fixed, for example the boolean vector describing the partition into blocks and the site mutation rates $\mu$. Therefore, we consider only an efficient representation of the ancestor sequences $\hat{a}$ and the parameters $\theta$ of the Markov chain.

Ancestor sequences are represented using a distribution-based optimal encoding scheme [118].

First, for each SNP $j$, the frequency $f_j(a)$ in the ancestors of each allele $a$ is calculated independently. If SNP $j$ falls in block $k$, this is given by $f_j(a) = \frac{1}{q_k} |\{c : \hat{a}_{k,c,j} = a\}|$. These independent frequencies are multiplied to form a distribution over the SNPs in block $k$, so that $Pr(\hat{a}_{k,c}) = \prod_{j=s_k}^{e_k} f_j(\hat{a}_{k,c,j})$. Using our scheme, the length of the sequence of ancestor $c$ of block $k$ is given by $L(\hat{a}_{k,c}) = -\log_2 Pr(\hat{a}_{k,c})$, so the representation length of all ancestor sequences for block $k$ is $S_k = \sum_c L(\hat{a}_{k,c})$. Note that we ignore the cost of representing the actual allele frequencies $f_j(a)$, since this is fixed for all models.

Since each parameter $\theta$ of the Markov chain is a continuous value with potentially infinite representation size, a limit must be placed on its accuracy. We apply Rissanen's result, which states that the optimal representation size for continuous parameters of a distribution from which $m$ samples are taken is $\frac{1}{2}\log_2 m$ bits [110]. Therefore, the cost $T_1$ to represent all $\theta_{1,c}$ parameters for the first block is given by $T_1 = \frac{q_1-1}{2}\log_2 n$, where $n = |\mathcal{H}| + 2|\mathcal{G}|$ is the total number of haplotypes represented by our data. Similarly, the cost $T_k$ to represent all $\theta_{k,c'\to c}$ parameters for subsequent blocks $k > 1$ is given by $T_k = \frac{q_k-1}{2} q_{k-1} \log_2 n$.

Thus, the total description length of a model $M$ with parameters $\Psi_M$ is given by $DL(M, \Psi_M) = \sum_k (S_k + T_k)$ and our aim is to find $M$ and $\Psi_M$ with minimal $DL(\mathcal{D}, M, \Psi_M) = DL(M, \Psi_M) - \log_2 Pr(\mathcal{D}|M, \Psi_M)$.

## 3.2 Search Algorithms

Our search algorithms take advantage of two features of the search space which were observed during development. First, it was noted that if the optimal model has several block boundaries, adding these one-by-one tends to incrementally improve the score. This means that boundaries may be examined individually and accumulated over several iterations. Second, even if the block boundaries in a model are not quite at their ideal locations, or the number of ancestors for each block is slightly sub-optimal, the model will nonetheless have a relatively strong score. This means that an initial quick scan can be used to assess regions of the search space, leading to further exploration in those areas which look most promising.

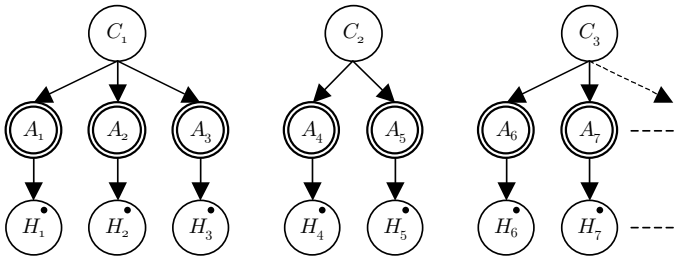
### 3.2.1 Finding one Model

Our first algorithm searches for a single explanation of the observed data with minimal MDL score. This algorithm uses a myopic strategy, retaining and attempting to improve only the best scoring model $M$ and parameters $\Psi_M$ found to date. We assign initial values to $M$ and $\Psi_M$ by learning the parameters of a model in which blocks are evenly spaced, as described in Section 3.3.1. Following this, we repeatedly execute a set of three stages, addition (Section 3.3.2), nudging (Section 3.3.3) and removal (Section 3.3.4), replacing $M$ and $\Psi_M$ as we go by any values which give a better $DL$ score. Each of these stage is applied to every block or boundary in the model in turn, before proceeding to the next stage. If two full rounds of these three stages produce no improvement, the algorithm outputs the final values of $M$ and $\Psi_M$.

For a given model structure, the parameters $\hat{a}$, $\mu$ and $\theta$ can be inferred from observed data by the EM algorithm [68]. To speed up our search, we perform EM for $\hat{a}$ and $\mu$ for each block independently, as described in Sections 3.4.1 and 3.4.2. Similarly, we learn $\theta$ for the Markov chain from adjacent pairs of blocks, as described in Section 3.4.3. For haplotypes, this is equivalent to performing EM on nodes $A_j$ and $H_j$ in the broken Bayesian Network shown in Figure 3.1, followed by EM on each node $C_k$ with just the single edge from $C_{k-1}$ to $C_k$ reintroduced.

Learning in this modular fashion means that during our model search, we need only recalculate parameters of blocks which are immediately affected by each adding, nudging or removing step. At the cost of losing some information, this shortcut introduces greater locality into our search space, reducing calculation time a great deal. For example, having added a boundary within block $k$ in an existing model $M$, we only relearn the ancestors $\hat{a}_k$ and $\hat{a}_{k+1}$, mutation rates $\mu_{s_k}, \ldots, \mu_{e_{k+1}}$

Figure 3.1: Broken Bayesian Network for haplotype data




and Markov transition probabilities $\theta_k$, $\theta_{k+1}$ and $\theta_{k+2}$. Parameters for unaffected blocks are copied from $M$, shifting indices appropriately. Furthermore, to calculate the new value of $DL(\mathcal{D}, M, \Psi_M)$, the elements $S_1, \ldots, S_{k-1}, S_{k+1}, \ldots, S_b$ and $T_1, \ldots, T_{k-1}, T_{k+2}, \ldots, T_b$ can be reused, along with cached forward probabilities such as $Pr(h_{s_1}, \ldots, h_{e_{k-1}}, c_{k-1}|M)$ and backward probabilities such as $Pr(h_{s_{k+2}}, \ldots, h_{e_b}|c_{k+1}, M)$ for each input haplotype $h$. As a result the complexity of each step in our learning algorithm is linear in the total number of SNPs, assuming that the number of SNPs per block can be bounded.

Note that the EM algorithm is only guaranteed to find values of parameters $\Psi_M$ which lead to a *local* maximum likelihood of the observed data. Therefore, for many of the models examined during our search, parameters will be independently learned several times, retaining the assignment which gives the best overall score. In so doing, we also partially address the concern that an assignment to $\Psi_M$ which gives a local maximum for the likelihood $Pr(\mathcal{D}|M, \Psi_M)$ does not necessarily give a local minimum for the description length $DL(\mathcal{D}, M, \Psi_M) = DL(M, \Psi_M) + DL(\mathcal{D}|M, \Psi_M)$, due to the complex dependence of $DL(M, \Psi_M)$ on parameters $\hat{a}$ and $\theta$. Nonetheless, it has been observed that the extrema usually coincide.

### 3.2.2 Ensemble Sampling

Our second algorithm samples an ensemble of locally optimal models for a set of observed data. This algorithm lies somewhere between the myopic search outlined in Section 3.2.1 and a fully-fledged Monte Carlo Markov Chain (MCMC) approach. The space of possible models is explored using Gibbs-style iterations, in which the existence and location of each block boundary is treated as the variable for resampling. The sampling begins with a model containing evenly spaced blocks and optimal parameters, as described in Section 3.3.1.

During each sampling iteration, each of the boundaries in the current model $M$ is reexamined in turn, by forcibly removing it by the process described in Section 3.3.4 and then attempting to add new boundaries into the larger block created. We attempt to introduce the first new boundary into this larger block by the process described in Section 3.3.2, if this causes an improvement in the DL score of $M$ and $\Psi_M$. If this attempt was successful, the algorithm then attempts to add another new boundary into the two new blocks created on either side. Since each iteration has the potential to up to triple the number of boundaries in the current model, models containing thousands of blocks can be reached quickly within a few sampling rounds.

The running time of the sampling algorithm is highly dependent on the parameters of the models inferred. If a bound is placed on the maximum number of SNPs and ancestors in any block, the time complexity is $O(l \cdot n \cdot s)$, where $l$ is the number of SNPs, $n$ is the number of haplotypes or genotypes and $s$ is the number of models to be sampled. In practice, unphased genotypes take much longer to analyze than haplotypes, due to the extra complexity of the calculations involved.

## 3.3 Model Search

This section describes the four steps used by the search algorithms described in Section 3.2.

### 3.3.1 Initial Model

We search for an initial model with blocks spaced evenly every $l_0$ SNPs, and then seek to optimize the number of ancestors $q_k$ for each block $k$ in turn. For each $k$, the ancestors, mutation rates and Markov parameters of our model are learned with $q_k \leftarrow 1$, after which we repeatedly learn parameters for new models with $q_k \leftarrow \lfloor (1.4 \cdot q_k) + 1 \rfloor$, for as long as the target score improves. We then try removing ancestors from the retained model in ascending order of their prior probability $\pi$, in each case relearning parameters $\mu$ and $\theta$, keeping any improvements in score. At this point, the value $q_k$ of the retained model should be close to optimal, so we try learning parameters for new models with $q'_k \approx q_k$ ancestors, where the number of independent runs for each $q'_k$ is proportional to $2^{-|q'_k - q_k|}$. If no more improvements can be found after several attempts, the search is completed for block $k$ and moves on to block $k+1$, until all blocks have been examined. The final outputs of this step are the model $M$ and parameters $\Psi_M$ which produced the best score.

### 3.3.2 Addition

In the addition step, we attempt to insert a new boundary somewhere within a block of the current model, optimizing the number of ancestors for the new blocks generated on either side. For addition in block $k$, we only relearn the parameters in $\Psi_M$ which are directly affected ($\hat{a}_k$, $\hat{a}_{k+1}$, $\mu_{s_k}, \ldots, \mu_{e_{k+1}}$, $\theta_k$, $\theta_{k+1}$, $\theta_{k+2}$), retaining all others with appropriate shifts in index.

The addition process should thoroughly test every possible boundary location within a block, however doing so is very costly. Therefore, the search for a suitable addition takes place in two stages, called *scan* and *isolation*. We begin the scan stage by learning a vector $V_j$ of new models and parameters for each insertion site $j = s_k + 1 \ldots e_k - 1$, where the number of ancestors for the two new blocks is equal to $q_k$ in the original model $M$. Then, for each entry in the vector, we try removing ancestors from each of the two new blocks in ascending order of their prior probability $\pi$, in each case relearning parameters $\mu$ and $\theta$, keeping any improvements in score. Having done so, the score of each $V_j$ is a reasonable guide to the value of adding a block boundary at $j$.

We begin the isolation stage by discarding all entries in $V$ whose score is lower than that of either of their neighbors. This search for local minima is guaranteed to remove at least half (rounded down) of those remaining. Then, we try to improve each remaining entry $V_j$ by slightly moving the newly placed boundary and relearning parameters, as in the nudging step described in Section 3.3.3. Following this, the search for local minima is repeated, continuing the isolation process until a single entry remains. In each round of isolation, we double the search time expended on improving each remaining entry, leading to a constant cost per round.

The single entry in $V$ remaining after the isolation stage replaces the current $M$ and $\Psi_M$ if it improves their score.

### 3.3.3 Nudging

In the nudging step, we try moving an existing block boundary a small distance, also allowing small changes in the number of ancestors for the blocks on either side. For nudging of the boundary between blocks $k$ and $k+1$, we only relearn the parameters in $\Psi_M$ which are directly affected ($\hat{a}_k$, $\hat{a}_{k+1}$, $\mu_{s_k}, \ldots, \mu_{e_{k+1}}$, $\theta_k$, $\theta_{k+1}$, $\theta_{k+2}$), retaining all others.

Clearly, the boundary which lies between blocks $k$ and $k+1$ is located between SNPs $j = e_k$ and $j+1$. To nudge a boundary, we learn the parameters of many models with the boundary placed at $j' \approx j$, where the number of ancestors $q'_k \approx q_k$ and $q'_{k+1} \approx q_{k+1}$ are close to those in the current model $M$. The number of independent learning runs tested for each assignment to $j'$, $q'_k$ and $q'_{k+1}$

is proportional to $2^{-|j'-j|\cdot|q'_k-q_k|\cdot|q'_{k+1}-q_{k+1}|}$. As the nudging step proceeds, the current $M$ and $\Psi_M$ are replaced by any new model and parameters which improve their score.

### 3.3.4 Removal

In the removal step, we attempt to remove an existing block boundary, optimizing the number of ancestors for the newly reunited block. Throughout the removal step for the boundary between blocks $k$ and $k+1$, we only relearn the parameters in $\Psi_M$ which are directly affected ($\hat{a}_k$, $\mu_{s_k}, \ldots, \mu_{e_k}$, $\theta_k$, $\theta_{k+1}$), retaining all others with appropriate shifts in index. The number of ancestors $q_k$ for the new block $k$ is optimized exactly as in the search for an initial model, described in Section 3.3.1. The current $M$ and $\Psi_M$ are replaced by any new model and parameters which improve their score.

## 3.4 Parameter Inference

This section describes the EM algorithms which are used to learn the parameters $\Psi_M$ for a given model $M$. We describe the EM iterations in terms of a series of parameter sets $\Psi^0, \Psi^1, \ldots$, where $\Psi^0$ receives initial values and each $\Psi^{i+1}$ is calculated from the previous $\Psi^i$. Each $\Psi^i$ contains a set of vectors $\Psi^i_{Y|Z}$, one for each of the conditional distributions $Pr(Y|Z)$ whose parameters we are learning. For example, when learning the mutation rates for block $k = 1$ of the broken Bayesian Network in Figure 3.1, $\Psi^i = \{\Psi^i_{H_1|A_1}, \Psi^i_{H_2|A_2}, \Psi^i_{H_3|A_3}\}$. The entries of the vector for each distribution define its conditional probabilities, so that $Pr(Y = y|Z = z, \Psi^i) \equiv \Psi^i_{y|z}$.

For all EM algorithms, iterations are stopped after round $i$ if the following convergence criterion is fulfilled: (a) at least 3 rounds have been performed ($i \geq 3$) and (b) for all conditional distributions $Pr(Y|Z)$ whose parameters are being learned, $\forall_z \sum_y |\Psi^i_{y|z} - \Psi^{i-1}_{y|z}| < \frac{1}{n}$, where $n = |\mathcal{H}| + 2|\mathcal{G}|$. Condition (a) is included in order to allow the EM algorithm to escape from an initial saddle-like region. When condition (b) is fulfilled, the expected frequency of each assignment $Y = y$ conditional on a particular $Z = z$ in a sample of size $n$ is guaranteed to have changed by less than 1, indicating a suitable degree of stability for our data.

Note that in all cases, we give double weight to genotypes since they represent evidence for two haplotype strands. Also, in our formulations, the symbol ? denotes an unknown measurement.

### 3.4.1 Ancestor Sequences

To learn the parameters $\hat{a}_k$ containing the sequences of ancestors for block $k$, we perform EM for the variables $A_{s_k}, \ldots, A_{e_k}$ in a broken Bayesian Network (shown for haplotypes in Figure 3.1). Note that we require a deterministic conditional distribution for each ancestor variable $A_j$, but the EM algorithm will rarely produce this, due to mutations which have occurred. Therefore, during the EM iterations, we fix the conditional distribution for each $H_{s_k} \ldots H_{e_k}$ as if no mutations have taken place, effectively clustering the observed sequences into $q_k$ self-similar clades. Since the edges from $C_{k-1}$ to $C_k$ and $C_k$ to $C_{k+1}$ have been removed, the distribution $Pr(c_k)$ acts as a simple prior for each ancestor, so the parameters we learn are $\Psi_{c_k}$ and $\Psi_{a_{s_k}|c_k}, \ldots, \Psi_{a_{e_k}|c_k}$.

The EM algorithm is initialized as follows. The prior over ancestors $c_k = 1 \ldots q_k$ is set to the uniform distribution, so that $\Psi^0_{c_k} \leftarrow \frac{1}{q_k}$. The conditional distribution over alleles $a_j$ given ancestors $c_k$ for each site $j = s_k \ldots e_k$ is set to the uniform distribution with significant random perturbation, so that $\Psi^0_{a_j|c_k} \propto 1 + r$ where each $r$ is distributed uniformly and independently over $[0, 1]$.

Following this, we repeatedly perform E and M steps, with $i = 1, 2, \ldots$ denoting the iteration. In the E step, for each ancestor $c_k = 1 \ldots q_k$, we calculate $Pr(c_k|h, \Psi^i)$ for each haplotype $h \in \mathcal{H}$ and $Pr(c_k|g, \Psi^i)$ for each genotype $g \in \mathcal{G}$. Similarly, for each site $j = s_k \ldots e_k$, allele $a_j \in B$ and ancestor $c_k$, we calculate $Pr(a_j, c_k|h, \Psi^i)$ for each $h$ and $Pr(a_j, c_k|g, \Psi^i)$ for each $g$. In the M step, we set $\Psi^{i+1}_{c_k}$ and $\Psi^{i+1}_{a_j|c_k}$ to their multinomial maximum likelihood values, based on the expected frequency of each variable value obtained during the E step.

The details for the E step are shown below. The probability $Pr(c_k|h,\Psi^i)$ of ancestor $c_k$ for haplotype $h$ is obtained by:

$$\begin{aligned} Pr(c_k|h,\Psi^i) &= \frac{Pr(h|c_k,\Psi^i)Pr(c_k|\Psi^i)}{\sum_{c_k} Pr(h|c_k,\Psi^i)Pr(c_k|\Psi^i)} \\ \text{where } Pr(h|c_k,\Psi^i) &= \prod_{j=s_k}^{e_k} Pr(h_j|c_k,\Psi^i) \\ Pr(h_j|c_k,\Psi^i) &= \sum_{a_j} Pr(h_j|a_j)Pr(a_j|c_k,\Psi^i) \\ Pr(h_j|a_j) &= \begin{array}{ll} 1 & \text{if } h_j \in \{a_j,?\} \\ 0 & \text{otherwise} \end{array} \end{aligned} \tag{3.1}$$

Similarly, the joint probability $Pr(a_j,c_k|h,\Psi^i)$ of ancestor $c_k$ and allele $a_j$ at site $j$ is:

$$\begin{aligned} Pr(a_j,c_k|h,\Psi^i) &= Pr(c_k|h,\Psi^i)Pr(a_j|h,c_k,\Psi^i) \\ \text{where } Pr(a_j|h,c_k,\Psi^i) &= \frac{Pr(h_j,a_j|c_k,\Psi^i)}{\sum_{a_j} Pr(h_j,a_j|c_k,\Psi^i)} \\ Pr(h_j,a_j|c_k,\Psi^i) &= Pr(a_j|c_k,\Psi^i)Pr(h_j|a_j) \end{aligned} \tag{3.2}$$

Parameters for genotype computations apply for both chromosomes symmetrically, so that $Pr(C'_k = c|\Psi^i) = Pr(C_k = c|\Psi^i)$ and $Pr(A'_j = a|C'_k = c,\Psi^i) = Pr(A_j = a|C_k = c,\Psi^i)$. The probability $Pr(c_k|g,\Psi^i)$ of ancestor $c_k$ for genotype $g$ is obtained by:

$$\begin{aligned} Pr(c_k|g,\Psi^i) &= \frac{Pr(g|c_k,\Psi^i)Pr(c_k|\Psi^i)}{\sum_{c_k} Pr(g|c_k,\Psi^i)Pr(c_k|\Psi^i)} \\ \text{where } Pr(g|c_k,\Psi^i) &= \sum_{c'_k} Pr(c'_k|\Psi^i) \prod_{j=s_k}^{e_k} Pr(g_j|c_k,c'_k,\Psi^i) \\ Pr(g_j|c_k,c'_k,\Psi^i) &= \sum_{a_j}\sum_{a'_j} Pr(g_j|a_j,a'_j)Pr(a_j|c_k,\Psi^i)Pr(a'_j|c'_k,\Psi^i) \\ Pr(g_j|a_j,a'_j) &= \begin{array}{ll} 1 & \text{if } g_j \in \left\{[a_j,a'_j],[a_j,?],[a'_j,?],[?,?]\right\} \\ 0 & \end{array} \end{aligned} \tag{3.3}$$

Similarly, the joint probability $Pr(a_j,c_k|g,\Psi^i)$ of ancestor $c_k$ and allele $a_j$ at site $j$ is:

$$\begin{aligned} Pr(a_j,c_k|g,\Psi^i) &= \frac{Pr(g,a_j,c_k|\Psi^i)}{\sum_{c_k}\sum_{a_j} Pr(g,a_j,c_k|\Psi^i)} \\ Pr(g,a_j,c_k|\Psi^i) &= Pr(c_k|\Psi^i)Pr(a_j|c_k,\Psi^i)Pr(g|a_j,c_k,\Psi^i) \\ Pr(g|a_j,c_k,\Psi^i) &= \sum_{c'_k} Pr(c'_k|\Psi^i)Pr(g|a_j,c_k,c'_k,\Psi^i) \\ Pr(g|a_j,c_k,c'_k,\Psi^i) &= Pr(g_j|a_j,c'_k,\Psi^i) \prod_{i=s_k,i\neq j}^{e_k} Pr(g_i|c_k,c'_k,\Psi^i) \\ Pr(g_j|a_j,c'_k,\Psi^i) &= \sum_{a'_j} Pr(a'_j|c'_k,\Psi^i)Pr(g_j|a_j,a'_j) \end{aligned} \tag{3.4}$$

When using genotypes derived from trio observations, some genotypes $g_j$ come with additional haplotype constraints $h_j$ and $h'_j$ (see Section 2.2). For these sites, the summations in Equations 3.3 and 3.4 over $a_j$ and $a'_j$ are replaced with the fixed values $h_j$ and $h'_j$ respectively. For other sites at which haplotypes could not be inferred from the trio, the probability calculations are unchanged.

The details for the M step are shown below, where all quantities on the right hand side of the equations are computed during the E step:

$$\Psi^{i+1}_{c_k} \leftarrow \frac{1}{n}\sum_{h\in\mathcal{H}} Pr(c_k|h,\Psi^i) + \frac{2}{n}\sum_{g\in\mathcal{G}} Pr(c_k|g,\Psi^i) \tag{3.5}$$

$$\Psi^{i+1}_{a_j|c_k} \leftarrow \frac{Pr(a_j,c_k|\mathcal{H},\mathcal{G},\Psi^i)}{\sum_{a_j} Pr(a_j,c_k|\mathcal{H},\mathcal{G},\Psi^i)} \tag{3.6}$$

$$\text{where } Pr(a_j,c_k|\mathcal{H},\mathcal{G},\Psi^i) = \frac{1}{n}\sum_{h\in\mathcal{H}} Pr(a_j,c_k|h,\Psi^i) + \frac{2}{n}\sum_{g\in\mathcal{G}} Pr(a_j,c_k|g,\Psi^i)$$

After $\Psi^{i+1}$ is computed, if the convergence criterion described in Section 3.4 is fulfilled, EM iterations are stopped and ancestor sequences are extracted by setting $\hat{a}_{k,c,j} \leftarrow \arg\max_{a_j} \Psi^{i+1}_{a_j|c_k}$. Otherwise, we repeat another E step, incrementing $i$ accordingly.

### 3.4.2 Mutation Rates

To learn the parameters $\mu_{s_k},\ldots,\mu_{e_k}$ containing the mutation rates of sites in block $k$, we perform EM for the variables $H_{s_k},\ldots,H_{e_k}$ in a broken Bayesian Network (shown for haplotypes in Figure 3.1). Note that this will be performed after ancestor sequences have been learned by the EM algorithm in Section 3.4.1, fixing the deterministic conditional distribution for each $A_{s_k}\ldots A_{e_k}$. As in Section 3.4.1, the absence of edges from $C_{k-1}$ to $C_k$ and $C_k$ to $C_{k+1}$ means that the distribution $Pr(c_k)$ acts as a simple prior, so the parameters we learn are $\Psi_{c_k}$ and $\Psi_{h_{s_k}|a_{s_k}},\ldots,\Psi_{h_{e_k}|a_{e_k}}$.

The EM algorithm is initialized as follows. The prior over ancestors $c_k = 1\ldots q_k$ is set to the uniform distribution, so that $\Psi^0_{c_k} \leftarrow \frac{1}{q_k}$. The conditional distribution over alleles $h_j \neq a_j$ for each site $j = s_k\ldots e_k$ are based on $\mu_{min}$ and $\mu_{max}$ as follows: if both $h_j$ and $a_j$ are observed at site $j$, then $\Psi^0_{h_j|a_j} \leftarrow \max(\sqrt{\mu_{min}\cdot\mu_{max}}, \mu^2_{max})$, otherwise $\Psi^0_{h_j|a_j} \leftarrow 0$. For each non-mutation of allele $a_j$, we initialize $\Psi^0_{a_j|a_j} \leftarrow 1 - \sum_{h_j\neq a_j}\Psi^0_{h_j|a_j}$.

Following this, we repeatedly perform E and M steps, with $i = 1,2,\ldots$ denoting the iteration. In the E step, for each ancestor $c_k = 1\ldots q_k$, we calculate $Pr(c_k|h,\Psi^i)$ for each haplotype $h\in\mathcal{H}$ and $Pr(c_k|g,\Psi^i)$ for each genotype $g\in\mathcal{G}$. Similarly, for each site $j = s_k\ldots e_k$ and alleles $a_j, h_j \in B$, we calculate $Pr(h_j,a_j|h,\Psi^i)$ for each $h$ and $Pr(h_j,a_j|g,\Psi^i)$ for each $g$. In the M step, we set $\Psi^{i+1}_{c_k}$ and $\Psi^{i+1}_{h_j|a_j}$ to their multinomial maximum likelihood values, based on the expected frequency of each variable value obtained during the E step, constraining to $\mu_{min}$ and $\mu_{max}$ as appropriate.

The details for the E step are shown below. The probability $Pr(c_k|h,\Psi^i)$ of ancestor $c_k$ for haplotype $h$ is obtained by:

$$Pr(c_k|h,\Psi_i) = \frac{Pr(h|c_k,\Psi_i)Pr(c_k|\Psi_i)}{\sum_{c_k} Pr(h|c_k,\Psi_i)Pr(c_k|\Psi_i)} \tag{3.7}$$

$$\text{where } Pr(h|c_k,\Psi_i) = \prod_{j=s_k}^{e_k} Pr(h_j|c_k,\Psi_i)$$

$$Pr(h_j|c_k,\Psi_i) = \begin{cases} 1 & \text{if } h_j = ? \\ Pr(h_j|\hat{a}_{k,c_k,j},\Psi_i) & \text{otherwise} \end{cases}$$

Similarly, the joint probability $Pr(h_j,a_j|h,\Psi^i)$ of alleles $h_j$ and $a_j$ at site $j$ is:

$$
\begin{aligned}
Pr(h_j, a_j | h, \Psi_i) &= \sum_{c_k} Pr(h_j, a_j, c_k | h, \Psi_i) \qquad (3.8)\\
\text{where } Pr(h_j, a_j, c_k | h, \Psi_i) &= Pr(c_k | h, \Psi_i) Pr(a_j | c_k) Pr(H_j = h_j | h, a_j, \Psi_i)\\
Pr(a_j | c_k) &= \begin{array}{ll} 1 & \text{if } \hat{a}_{k,c_k,j} = a_j \\ 0 & \text{otherwise} \end{array}\\
Pr(H_j = x | h, a_j, \Psi_i) &= \begin{array}{ll} 1 & \text{if } h_j = x \\ Pr(H_j = x | a_j, \Psi_i) & \text{if } h_j = ? \\ 0 & \text{otherwise} \end{array}
\end{aligned}
$$

Parameters for genotype computations apply for both chromosomes symmetrically, so that $Pr(C'_k = c|\Psi^i) = Pr(C_k = c|\Psi^i)$, $Pr(A'_j = a|C'_k = c, \Psi^i) = Pr(A_j = a|C_k = c, \Psi^i)$ and $Pr(H'_j = h|A'_j = a, \Psi^i) = Pr(H_j = h|A_j = a, \Psi^i)$. The probability $Pr(c_k|g, \Psi^i)$ of ancestor $c_k$ for genotype $g$ is obtained by:

$$
\begin{aligned}
Pr(c_k | g, \Psi_i) &= \frac{Pr(g | c_k, \Psi_i) Pr(c_k | \Psi_i)}{\sum_{c_k} Pr(g | c_k, \Psi_i) Pr(c_k | \Psi_i)} \qquad (3.9)\\
\text{where } Pr(g | c_k, \Psi_i) &= \sum_{c'_k} Pr(c'_k | \Psi_i) Pr(g | c_k, c'_k, \Psi_i)\\
Pr(g | c_k, c'_k, \Psi_i) &= \prod_{j=s_k}^{e_k} Pr(g_j | c_k, c'_k, \Psi_i)\\
Pr(g_j | c_k, c'_k, \Psi_i) &= \sum_{a_j} \sum_{a'_j} Pr(g_j | a_j, a'_j, \Psi_i) Pr(a_j | c_k) Pr(a'_j | c'_k)\\
Pr(g_j | a_j, a'_j, \Psi_i) &= \sum_{h_j} \sum_{h'_j} Pr(g_j | h_j, h'_j) Pr(h_j | a_j, \Psi_i) Pr(h'_j | a'_j, \Psi_i)\\
Pr(g_j | h_j, h'_j) &= \begin{array}{ll} 1 & \text{if } g_j \in \left\{ [h_j, h'_j], [h_j, ?], [h'_j, ?], [?, ?] \right\} \\ 0 & \text{otherwise} \end{array}
\end{aligned}
$$

Similarly, the joint probability $Pr(h_j, a_j|g, \Psi^i)$ of alleles $h_j$ and $a_j$ at site $j$ is:

$$
\begin{aligned}
Pr(h_j, a_j | g, \Psi_i) &= \frac{Pr(g, h_j, a_j | \Psi_i)}{\sum_{a_j} \sum_{h_j} Pr(g, h_j, a_j | \Psi_i)} \qquad (3.10)\\
\text{where } Pr(g, h_j, a_j | \Psi_i) &= \sum_{c_k} \sum_{c'_k} Pr(c_k | \Psi_i) Pr(c'_k | \Psi_i) Pr(g, h_j, a_j | c_k, c'_k, \Psi_i)\\
Pr(g, h_j, a_j | c_k, c'_k, \Psi_i) &= Pr(g_j, h_j, a_j | c_k, c'_k, \Psi_i) \prod_{i=s_k, i \neq j}^{e_k} Pr(g_i | c_k, c'_k, \Psi_i)\\
Pr(g_j, h_j, a_j | c_k, c'_k, \Psi_i) &= Pr(a_j | c_k) \sum_{a'_j} Pr(a'_j | c'_k) Pr(g_j, h_j | a_j, a'_j, \Psi_i)\\
Pr(g_j, h_j | a_j, a'_j, \Psi_i) &= Pr(h_j | a_j, \Psi_i) \sum_{h'_j} Pr(h'_j | a'_j, \Psi_i) Pr(g_j | h_j, h'_j)
\end{aligned}
$$

For trio-derived genotype observations for which haplotype constraints $h_j$ and $h'_j$ are available, the summations over $h_j$ and $h'_j$ in Equations 3.9 and 3.10 are replaced with the fixed values of $h_j$ and $h'_j$.

The details for the M step are shown below, where all quantities on the right hand side of the equations are computed during the E step:

$$\Psi^{i+1}_{c_k} \leftarrow \frac{1}{n}\sum_{h\in\mathcal{H}} Pr(c_k|h,\Psi^i) + \frac{2}{n}\sum_{g\in\mathcal{G}} Pr(c_k|g,\Psi^i) \tag{3.11}$$

$$h_j \neq a_j : \Psi^{i+1}_{h_j|a_j} \leftarrow \min(\mu_{min}, \max(\mu_{max}, \Psi^{*i+1}_{h_j|a_j})) \tag{3.12}$$

$$\Psi^{i+1}_{a_j|a_j} \leftarrow 1 - \sum_{h_j\neq a_j} \Psi^{i+1}_{h_j|a_j} \tag{3.13}$$

$$\text{where } \Psi^{*i+1}_{h_j|a_j} = \frac{Pr(h_j,a_j|\mathcal{H},\mathcal{G},\Psi^i)}{\sum_{h_j} Pr(h_j,a_j|\mathcal{H},\mathcal{G},\Psi^i)}$$

$$Pr(h_j,a_j|\mathcal{H},\mathcal{G},\Psi^i) = \frac{1}{n}\sum_{h\in\mathcal{H}} Pr(h_j,a_j|h,\Psi^i) + \frac{2}{n}\sum_{g\in\mathcal{G}} Pr(h_j,a_j|g,\Psi^i)$$

Note that $\Psi^{*i+1}_{h_j|a_j}$ contains the maximum likelihood estimates for mutation rates, whereas the updated $\Psi^{i+1}_{h_j|a_j}$ is constrained by $\mu_{min}$ and $\mu_{max}$. After $\Psi^{i+1}$ is computed, if the convergence criterion described in Section 3.4 is fulfilled, EM iterations are stopped and mutation rates are assigned by setting $\mu_{j,a\to h} \leftarrow \Psi^{i+1}_{h_j|a_j}$. Otherwise, we repeat another E step, incrementing $i$ accordingly.

### 3.4.3 Markov Transition Probabilities

To learn the parameters $\theta_k$ containing the Markov chain transition probability from block $k-1$ to $k$, we perform EM for the variable $C_k$ in a Bayesian Network where for all $i \neq k$, the edge from $C_{i-1}$ to $C_i$ is removed. Note that this will be performed after ancestor sequences and mutation rates have been learned by the EM algorithms in Sections 3.4.1 and 3.4.2, fixing the conditional distributions for each $A_{s_k}\dots A_{e_k}$ and $H_{s_k}\dots H_{e_k}$. When learning $\theta_k$, we also ensure that all parameters $\theta_1,\dots,\theta_{k-1}$ have already been learned, so that the distribution $\pi_{k-1}$ (see Section 2.1) can be used as a suitable prior for the values of variable $C_{k-1}$. In this way, the only parameter we learn is $\Psi_{c_k|c_{k-1}}$.

The initial conditional distribution over ancestors $c_k = 1\dots q_k$ given prior ancestors $c_{k-1} = 1\dots q_{k-1}$ is set to the uniform distribution, so that $\Psi^0_{c_k|c_{k-1}} \leftarrow \frac{1}{q_k}$. Following this, we repeatedly perform E and M steps, with $i = 1, 2, \dots$ denoting the iteration. In the E step, for ancestors $c_k$ and $c_{k-1}$, we calculate $Pr(c_k, c_{k-1}|h,\Psi^i)$ for each haplotype $h \in \mathcal{H}$ and $Pr(c_k, c_{k-1}|g,\Psi^i)$ for each genotype $g \in \mathcal{G}$. In the M step, we set $\Psi^{i+1}_{c_k|c_{k-1}}$ to its multinomial maximum likelihood value, based on the expected frequency of each variable value obtained during the E step.

The details for the E step are shown below. The joint probability $Pr(c_k, c_{k-1}|h,\Psi^i)$ of ancestors $c_k$ and $c_{k-1}$ for haplotype $h$ is obtained by:

$$Pr(c_k,c_{k-1}|h,\Psi^i) = \frac{Pr(h,c_k|c_{k-1},\Psi^i)Pr(c_{k-1})}{\sum_{c_{k-1}}\sum_{c_k} Pr(h,c_k|c_{k-1},\Psi^i)Pr(c_{k-1})} \tag{3.14}$$

$$\text{where } Pr(h,c_k|c_{k-1},\Psi^i) = Pr(h|c_k)Pr(h|c_{k-1})Pr(c_k|c_{k-1},\Psi^i)$$

$$Pr(h|c_k) = \prod_{j=s_k}^{e_k} Pr(h_j|c_k)$$

$$Pr(h_j|c_k) = \begin{cases} 1 & \text{if } h_j = ? \\ \mu_{j,\hat{a}_{k,c,j}\to h_j} & \text{otherwise} \end{cases}$$

$$Pr(c_{k-1}) = \pi_{k-1,c_{k-1}} \tag{3.15}$$

Parameters for genotype calculations apply for both chromosomes symmetrically, so that $Pr(C'_k = c) = Pr(C_k = c)$, $Pr(C'_k = y|C'_{k-1} = z,\Psi^i) = Pr(C_k = y|C_{k-1} = z,\Psi^i)$ and $Pr(H'_j = h|C'_k = c) =$

$Pr(H_j = h|C_k = c)$. The joint probability $Pr(c_k, c_{k-1}|g, \Psi^i)$ of ancestors $c_k$ and $c_{k-1}$ for genotype $g$ is:

$$
\begin{aligned}
Pr(c_k, c_{k-1}|g, \Psi^i) &= \frac{Pr(g, c_k|c_{k-1}, \Psi^i)Pr(c_{k-1})}{\sum_{c_{k-1}}\sum_{c_k} Pr(g, c_k|c_{k-1}, \Psi^i)Pr(c_{k-1})} \qquad (3.16)\\
\text{where } Pr(g, c_k|c_{k-1}, \Psi^i) &= \sum_{c'_{k-1}}\sum_{c'_k} Pr(c'_{k-1})Pr(g, c_k, c'_k|c_{k-1}, c'_{k-1}, \Psi^i)\\
Pr(g, c_k, c'_k|c_{k-1}, c'_{k-1}, \Psi^i) &= Pr(c_k|c_{k-1}, \Psi^i)Pr(c'_k|c'_{k-1}, \Psi^i)Pr(g|c_k, c'_k)Pr(g|c_{k-1}, c'_{k-1})\\
Pr(g|c_k, c'_k) &= \prod_{j=s_k}^{e_k} Pr(g_j|c_k, c'_k)\\
Pr(g_j|c_k, c'_k) &= \sum_{h_j}\sum_{h'_j} Pr(g_j|h_j, h'_j)Pr(h_j|c_k)Pr(h'_j|c'_k)\\
Pr(g_j|h_j, h'_j) &= \begin{array}{ll} 1 & \text{if } g_j \in \left\{[h_j, h'_j], [h_j, ?], [h'_j, ?], [?, ?]\right\} \\ 0 & \text{otherwise}\end{array}
\end{aligned}
$$

For trio-derived genotypes for which haplotypes $h_j$ and $h'_j$ are known, the summations over $h_j$ and $h'_j$ in Equation 3.16 are replaced with the fixed values.

The details for the M step are shown below, where all quantities on the right hand side of the equations are computed during the E step:

$$
\begin{aligned}
\Psi^{i+1}_{c_k|c_{k-1}} &\leftarrow \frac{Pr(c_k, c_{k-1}|\mathcal{H}, \mathcal{G}, \Psi^i)}{\sum_{c_k} Pr(c_k, c_{k-1}|\mathcal{H}, \mathcal{G}, \Psi^i)} \qquad (3.17)\\
\text{where } Pr(c_k, c_{k-1}|\mathcal{H}, \mathcal{G}, \Psi^i) &= \frac{1}{n}\sum_{h\in\mathcal{H}} Pr(c_k, c_{k-1}|h, \Psi^i) + \frac{2}{n}\sum_{g\in\mathcal{G}} Pr(c_k, c_{k-1}|g, \Psi^i)
\end{aligned}
$$

After $\Psi^{i+1}$ is computed, if the convergence criterion described in Section 3.4 is fulfilled, EM iterations are stopped and transition probabilities are assigned by setting $\theta_{k,c_{k-1}\to c_k} \leftarrow \Psi^{i+1}_{c_k|c_{k-1}}$. Otherwise, we repeat another E step, incrementing $i$.

Clearly, when learning parameter $\theta_1$ for the first block in the Markov chain, the target distribution is not conditional on any prior. The above formulation can be adapted for this case by introducing a pseudo-prior variable $C_0$ which takes a single value $C_0 = 1$, setting $s_0 > e_0$ to remove any terms $\prod_{j=s_0}^{e_0}$ and assigning $\pi_{0,1} = 1$. Once EM is finished, the probabilities for the first block of the Markov chain are assigned by setting $\theta_{1,c} \leftarrow \Psi^{i+1}_{C_1=c|C_0=1}$.

# Chapter 4

# Haplotype Resolution

## Introduction

In this chapter we describe how our model is applied to the problem of haplotype resolution. Haplotype resolution (or phasing) denotes the inference of the hidden haplotype pairs from which a set of observed genotypes are constituted. Haplotypes are essential for many genetic mapping methods, which are based on correlations between phenotypes and combinations of alleles on a single chromosome. A genotype containing $s$ heterozygous sites can be separated into constituent haplotypes in $2^{s-1}$ different ways, so assumptions are required about how the haplotypes in a population are distributed. One common assumption is that similar haplotypes are likely to be present in many individuals, due to shared ancestry.

An early approach to haplotype resolution was Clark's parsimony-based algorithm [10], later improved by Gusfield [41] and Eskin and Halperin [23, 42]. A likelihood-based EM algorithm [25, 73, 128] gives far superior results but is infeasible for large experiments, since for genotypes with $s$ heterozygous loci its complexity is $O(2^s)$. Other methods include MCMC-based algorithms by Stephens *et al.* [125] and Niu *et al.* [92] and an approach based on variable length Markov chains by Eronen *et al.* [22]. None of these methods for haplotype resolution directly consider the haplotype block phenomenon. More recently, Kimmel and Shamir published an algorithm called GERBIL which uses haplotype blocks, and shares many features with our own approach [56].

Section 4.1 describes how our statistical model and learning algorithms are used to perform haplotype resolution. By applying standard methods for inference in Bayesian Networks, the resolution is trivial once a suitable statistical model has been learned. Section 4.2 demonstrates the accuracy of our approach by testing it on some real-world genotype data for which the underlying haplotypes are known. Finally, Section 4.3 discusses how our algorithm can be extended for an ensemble of inferred models, and compares it against the approach of Kimmel and Shamir.

This research was presented at RECOMB 2003 and published in the *Journal of Computational Biology* [34, 36].

## 4.1 Method

We perform haplotype resolution in two stages. First, we search for the best model $M$ with parameters $\Psi_M$ for observed genotype data $\mathcal{G}$, using the MDL criterion and search algorithm described in Chapter 3. We then use this model to define a function $H(g, M, \Psi_M)$ which gives a pair of haplotypes $(h, h')$ which is compatible with each genotype $g \in \mathcal{G}$ and likely under $M$ and $\Psi_M$. Ideally, this function would find the assignment of $h_1, \ldots, h_l, h'_1, \ldots, h'_l$ with maximum likelihood in the model's genotype Bayesian Network, giving $\arg\max_{(h,h')} Pr(g, h, h'|M, \Psi_M)$.

Unfortunately, computing this is infeasible, since it requires a summation over all paths through the two Markov chains to generate joint distributions over $h$ and $h'$ before calculating their maximal assignments, an operation with exponential complexity in terms of $l$. Instead, we find the joint maximum likelihood assignment of the haplotype pair $h_1, \ldots, h_l, h'_1, \ldots, h'_l$ and ancestor indices $c_1, \ldots, c_b, c'_1, \ldots, c'_b$ which is compatible with $g$ by bucket variable elimination [17]. In doing so, we only consider the single most probable path through the Markov chain that could lead to each haplotype, analogous to applying the Viterbi algorithm on a Hidden Markov Model. This approximation is reasonable because one path is likely to give a much higher probability for a particular haplotype than the others, since mutations are rare.

## 4.2 Results

Many studies of the haplotypes in particular genomic regions have been carried out over the past few years [3]. However, in most cases, the haplotypes used for the study were obtained using an existing haplotype resolution algorithm, so they hardly form a suitable basis for a comparison of such methods. Furthermore, not all studies are based on closely-spaced SNP markers, so our

block-based approach would be ineffective on the data sets obtained. Our results are based on two high resolution data sets for which the underlying haplotypes were measured: (a) 22 haplotypes of 52 SNPs over 24 kb in the ACE region [107], (b) 20 haplotypes of 24,047 SNPs over the whole of chromosome 21 [94]. These data sets are described in greater detail in Chapter 1. For the purposes of this comparison, we examined the five contiguous stretches of approximately 100 SNPs in chromosome 21 which extend over less than 35,000 bp.

To compare the quality of haplotype resolution, we used 10 random pairings of the true haplotypes for each region to generate genotypes, which were then passed to each algorithm for haplotype resolution. We applied our approach for three different values of $\mu_{min}$ and $\mu_{max}$ in two ways, first restricting the search to models which place all the SNPs in a single block (i.e. $b = 1$) and then allowing the block divisions to also be learned. The results are compared against those for four other methods: (i) Clark's algorithm, slightly modified to deal with unknowns [10], (ii) Our local variation of the EM algorithm which overcomes its exponential complexity, (iii) The PHASE algorithm developed by Stephens *et al.* [125], (iv) A beta version of the HAPLOTYPER algorithm developed by Niu *et al.* [92]. Table 4.1 compares the quality of haplotype resolution, as measured by the proportion of individuals phased incorrectly. A finer comparison, shown in Table 4.2, is generated by measuring the proportion of pairs of adjacent sites which are phased incorrectly relative to each other.

The local EM method overcomes the complexity of the full EM method as follows. It begins by performing the full EM on the SNPs observed within disjoint subranges of a certain length, say 8. From each range, a fixed number $m$ of the haplotype segments are then chosen, taking those whose probabilities were estimated to be the highest. The best haplotypes from adjacent ranges are then crossed to create a list of $m^2$ possible haplotypes over the combined range. EM is then performed upon this new set, after which the $m$ best values are again chosen for the next level of iteration, and so on. An EM algorithm based on this approach was recently published by Qin *et al.* [103].

The first set of tests, in which the number of blocks $b$ is fixed to 1, demonstrates the effectiveness of our ancestor and mutation model, even when the possible presence of haplotype blocks is ignored. In other words, model-based Bayesian clustering is an effective method for haplotype resolution over closely-linked SNPs. For the high resolution data from chromosome 21, the results are compelling – our approach consistently outperforms previously published algorithms, with the exception of some cases where $\mu_{max} = 10^{-2}$. The contrast is particularly marked in the site pairwise error rates, indicating the suitability of our method for high resolution disease mapping. Our model-based approach also obtained better results than our own Local EM algorithm with the exception of data set C21e, to be discussed further below. For the ACE data set, the results are more mixed, perhaps because the lower SNP density in that study makes it less suitable for our model.

The second set of tests, in which an unrestricted model search is performed (allowing $b \geq 1$), demonstrates the extra accuracy that is achieved by allowing multiple blocks to be included in a model. However, for chromosome 21 data sets (a) through (d), there is no significant difference between the results of the two experiments. This surprising result is explained by the fact that even in the unrestricted model search, many of the models learned from these regions placed all the SNPs in a single block. By contrast, the unrestricted searches for data set (e) showed a clear improvement in mean site pairwise error rate from $(0.0161, 0.0171, 0.0116)$ to $(0.0048, 0.0045, 0.0080)$ for the three values of $\mu_{max}$, reflecting the fact that they all indicated the presence of more than one block. Clearly, for data that extends over longer chromosomal regions, the contrast between the two types of search will increase in prominence.

## 4.3 Discussion

Our method for haplotype resolution can be extended to infer haplotypes using an ensemble of sampled models instead of a single model. First, individual haplotype resolutions are obtained for each model in the ensemble as in Section 4.1. For the final haplotype pair, the alleles at each

heterozygous site are oriented relative to the previous heterozygous site so as to be compatible with the maximum number of individual model-based resolutions. Since homozygous sites are irrelevant in terms of haplotype phasing, they have no role during this operation. Initial studies show that haplotype resolutions calculated from samples in this way are more accurate on average than those based on a single model.

Kimmel and Shamir recently published a block-based algorithm for haplotype resolution called GERBIL [56]. Kimmel and Shamir report that their method improves on our results, so it is interesting to compare their approach with ours. Both methods share a basic common framework – the parameters of a statistical model are learned by EM from the observed genotype evidence, then this model is used to infer the maximum likelihood pair of haplotypes for each individual. The key difference between the methods lies in the statistical model used.

Recall from Chapter 2 that we describe the distribution of haplotypes in a block using a combination of ancestor sequences and mutation rates. The sequence of each ancestor is deterministic, with variations on that ancestor haplotype produced by subsequent rare mutation. By contrast, the statistical model used by Kimmel and Shamir extends the model of Koivisto *et al.* [59] to genotype data. It uses a non-deterministic distribution for each ancestor sequence, where each site in the ancestor has a distribution over the possible alleles, independent of the other sites. Their model is the equivalent of removing variables $A_j$ and $A'_j$ in the Bayesian Network shown in Figure 2.2, and making each variable $H_j$ and $H'_j$ non-deterministic given the prior values of $C_k$ and $C'_k$ respectively.

Our work differs from that of Kimmel and Shamir in a few other ways. We optimize the number of ancestors for a block in terms of the MDL criterion, which penalizes models which over-fit by using an excessive number of ancestors. By contrast, Kimmel and Shamir apply a more straightforward maximum likelihood criterion, and grow the number of ancestor sequences for a block until only a small improvement in likelihood is obtained. Kimmel and Shamir find the globally optimal partition by applying a dynamic programming algorithm similar to that of Zhang *et al.* [149]. By contrast, we use a weaker heuristic approach since our MDL criterion cannot be decomposed into individual functions for each block. Lastly, our statistical model infers the Markov chain between blocks as an integral part of the learning process, and counts the parameters of this chain as part of the MDL criterion. By contrast, Kimmel and Shamir infer the Markov-like relationship between consecutive blocks only after the block partition and ancestor sequences have been learned.

Table 4.1: Mean proportion of subject genotypes phased incorrectly

| Proportion of subjects[a] | C21a[b] | C21b | C21c | C21d | C21e | ACE |
|---|---|---|---|---|---|---|
| Clark | .8222 | .7300 | .5300 | .7900 | .8444 | .5091 |
| Local EM[c] | .5889 | .3900 | .1300 | .5800 | .5667 | .3545 |
| HAPLOTYPER[d] | .6667 | – | .6000 | .6000 | – | .2818 |
| PHASE | .6778 | .5000 | .4800 | .4800 | .6556 | .4727 |
| HaploBlock[e], $b = 1, \mu_{max} = 10^{-4}$ | .4222 | .2200 | .1400 | .2600 | .6889 | .5364 |
| HaploBlock, $b = 1, \mu_{max} = 10^{-3}$ | .4556 | .2300 | .1000 | .3100 | .6778 | .5636 |
| HaploBlock, $b = 1, \mu_{max} = 10^{-2}$ | .4333 | .5500 | .0800 | .4600 | .5667 | .5364 |
| HaploBlock, $\mu_{max} = 10^{-4}$ | .4556 | .3400 | .1200 | .2800 | .5667 | .4818 |
| HaploBlock, $\mu_{max} = 10^{-3}$ | .4778 | .3300 | .1200 | .3800 | .6444 | .6818 |
| HaploBlock, $\mu_{max} = 10^{-2}$ | .7111 | .4700 | .1200 | .4300 | .5667 | .7273 |

[a] Sites with unknowns were excluded from the comparison.

[b] All chromosome 21 regions are from contig NT002836, over the following stretches of base pairs. a: 1262471-1292884, b: 7490174-7517009, c: 10972404-10996329, d: 13622368-13650628, e: 14999072-15030226.

[c] For Local EM and HAPLOTYPER, we took the maximum likelihood result of 20 runs.

[d] The HAPLOTYPER beta version failed on data with many unknowns – averages are for successful runs, if any.

[e] For each HaploBlock run, we set $\mu_{min} = \mu_{max}^2$.

Table 4.2: Mean proportion of adjacent sites phased incorrectly relative to each other

| Proportion of pairs | C21a | C21b | C21c | C21d | C21e | ACE |
|---|---|---|---|---|---|---|
| Clark | .0548 | .0251 | .0280 | .0329 | .0234 | .0381 |
| Local EM | .0095 | .0042 | .0009 | .0047 | .0083 | .0152 |
| HAPLOTYPER | .0224 | – | .0204 | .0077 | – | .0102 |
| PHASE | .0669 | .0403 | .0655 | .0262 | .0183 | .0419 |
| HaploBlock, $b = 1, \mu_{max} = 10^{-4}$ | .0052 | .0011 | .0007 | .0014 | .0161 | .0100 |
| HaploBlock, $b = 1, \mu_{max} = 10^{-3}$ | .0053 | .0016 | .0001 | .0012 | .0171 | .0144 |
| HaploBlock, $b = 1, \mu_{max} = 10^{-2}$ | .0036 | .0074 | .0006 | .0027 | .0116 | .0185 |
| HaploBlock, $\mu_{max} = 10^{-4}$ | .0039 | .0015 | .0001 | .0008 | .0048 | .0109 |
| HaploBlock, $\mu_{max} = 10^{-3}$ | .0030 | .0030 | .0005 | .0015 | .0045 | .0109 |
| HaploBlock, $\mu_{max} = 10^{-2}$ | .0068 | .0058 | .0005 | .0024 | .0080 | .0173 |

# Chapter 5

# Linkage Disequilibrium Mapping

## Introduction

The goal of genetic mapping is to narrow down the location of a hidden genetic factor underlying some observed phenotypic variation. Specifically, linkage disequilibrium (LD) mapping performs this task based on a set of marker measurements from unrelated individuals. This chapter describes how our statistical model and learning algorithms are applied to perform LD mapping in the presence of haplotype blocks.

Until recently, most LD mapping studies were based on correlations between phenotype status and the allele frequencies at individual markers, with little success [108, 8]. In a study with hundreds of markers, the strength of the correlation of a single marker close to the phenotypic site is hard to distinguish from other markers associated by chance [142, 93]. A more powerful approach treats multi-marker haplotypes as the variable for correlation [77, 53, 6, 26, 24]. The descendants of a disease founder are more clearly identified by a haplotype than by a single marker since two haplotypes with different lineages are unlikely to be identical at many sites [132]. Nevertheless, tests based on haplotypes must consider the possibility that recombinations and mutations have taken place, complicating the correlation with disease. Many methods for addressing this challenge have been proposed, based on evolutionary trees [66, 116, 129], haplotype sharing [80, 86], clustering [70, 74], distance metrics [130, 85] and the coalescent [106, 88, 87].

Unlike these previous methods, our approach to LD mapping specifically considers the implications of the block-like structure of haplotype variation. Section 5.1 describes how our model is applied to the LD mapping problem. Section 5.2 shows the results of our approach, in comparison with a method based on individual SNPs and a competing haplotype-based algorithm. Finally, Section 5.3 briefly discusses how our mapping framework could be extended for more complex disease models and haplotype tagging SNPs (htSNPs).

This research was presented at ISMB 2004 and published in *Bioinformatics* [35].

## 5.1 Method

A high density LD mapping study is based on a list $\mathcal{H} = \{h^1, \ldots, h^n\}$ of $n$ phased haplotypes or a list $\mathcal{G} = \{g^1, \ldots, g^n\}$ of $n$ unphased genotypes over a genomic region of interest. We use the symbol $\mathcal{D}$ to refer to input $\mathcal{H}$ or $\mathcal{G}$ as appropriate. The other inputs are a list $\mathcal{P} = \{p^1, \ldots, p^n\}$ of phenotypes associated with each haplotype or genotype and the distances $d_j$ in base pairs between adjacent SNPs $j$ and $j+1$ over $j = 1 \ldots l-1$. For haplotype mapping, each haplotype $h^i$ is a string of $l$ symbols from the set $B$ of SNP alleles, where $l$ is the number of loci examined. For genotype mapping, each genotype $g^i$ is a string of $l$ elements from the set $D$ of unordered SNP allele pairs. Each $p^i$ is in the range $1 \ldots p_{max}$ where $p_{max}$ is the total number of phenotypes observed. In a simple case-control study, $p_{max} = 2$.

We are searching for an unobserved genetic locus within the candidate region that affects the phenotypes observed. Let $L_j$ denote the hypothesis that this locus is situated in the interval between SNPs $j$ and $j+1$, so that we consider the set of hypotheses $\{L_1, \ldots, L_{l-1}\}$. We express the output of a mapping study as a posterior distribution $Pr(L_j|\mathcal{P}, \mathcal{D})$ over these alternatives, normalized so that $\sum_{j=1}^{l-1} Pr(L_j|\mathcal{P}, \mathcal{D}) = 1$. This distribution is calculated in the following four stages.

First, we infer an ensemble $\mathcal{M}$ of statistical models which are locally optimal in terms of the MDL criterion, as explained in Chapter 3. We ignore the phenotypes $\mathcal{P}$ during this process, since they barely affect the data likelihood. Second, for each model $M$ with parameters $\Psi_M$ in the ensemble $\mathcal{M}$, we calculate the posterior probability that each block contains the phenotypic locus. Let $U_k$ denote the hypothesis that the locus is in block $k$ of $M$. The posterior distribution $Pr(U_k|\mathcal{P}, \mathcal{D}, M, \Psi_M)$ is calculated using the method described in Section 5.1.1 or 5.1.2 as appropriate. Note that at this stage the phenotype data is used to assess hypotheses relating to blocks, rather than SNP intervals, since each model inferred assumes that the alleles within each block segregate together.

Third, the posterior distribution $Pr(U_k|\mathcal{P},\mathcal{D},M,\Psi_M)$ over the blocks in model $M$ is converted into a posterior $Pr(L_j|\mathcal{P},\mathcal{D},M,\Psi_M)$ over SNP intervals. For an interval $(j,j+1)$ in block $k$, for which $s_k \leq j < e_k$, we allocate the posterior in proportion to the length $d_j$ of the interval, setting $Pr(L_j|\mathcal{P},\mathcal{D},M,\Psi_M) = \frac{d_j}{V_k} Pr(U_k|\mathcal{P},\mathcal{D},M,\Psi_M)$, where $V_k$ is the total length of block $k$. For an interval $(j,j+1)$ on the boundary between blocks $k$ and $k+1$, for which $j = s_{k+1}-1 = e_k$, we assume that half of the interval lies in each block, setting $Pr(L_j|\mathcal{P},\mathcal{D},M,\Psi_M) = \frac{d_j}{2V_k} Pr(U_k|\mathcal{P},\mathcal{D},M,\Psi_M) + \frac{d_j}{2V_{k+1}} Pr(U_{k+1}|\mathcal{P},\mathcal{D},M,\Psi_M)$. The block length $V_k$ is obtained by summing the interlocus distances $d_j$ within the block and half of those at either end, i.e. $V_k = \sum_{j=s_k}^{e_k-1} d_j + \frac{1}{2}(d_{s_k-1} + d_{e_k})$. Note that $V_1$ and $V_b$ lose elements $d_{s_k-1}$ and $d_{e_k}$ respectively from this sum, where $b$ is the number of blocks in the model.

In the fourth and final stage, the individual posterior distributions $Pr(L_j|\mathcal{P},\mathcal{D},M,\Psi_M)$ obtained from each model $M$ with parameters $\Psi_M$ in the ensemble $\mathcal{M}$ are combined into a single statistic by uniform model averaging, so that $Pr(L_j|\mathcal{P},\mathcal{D}) = \frac{1}{|\mathcal{M}|} \sum_{(M,\Psi_M)\in\mathcal{M}} Pr(L_j|\mathcal{P},\mathcal{D},M,\Psi_M)$. We use a uniform prior for the averaging since the sampling process has already introduced a strong bias towards models with a low MDL score.

### 5.1.1 Haplotypes posterior

Recall that hypothesis $U_k$ states that the phenotypic locus is located in block $k$ of a model. Under Bayes' Rule, the posterior probability of hypothesis $U_k$ is given by $Pr(U_k|\mathcal{P},\mathcal{H},M,\Psi_M) = \frac{Pr(\mathcal{P}|U_k,\mathcal{H},M,\Psi_M)Pr(U_k|\mathcal{H},M,\Psi_M)}{Pr(\mathcal{P}|\mathcal{H},M,\Psi_M)}$. Since $Pr(\mathcal{P}|\mathcal{H},M,\Psi_M)$ is the same for all $k$ and we assume that the prior $Pr(U_k|\mathcal{H},M,\Psi_M)$ does not depend on $\mathcal{H}$, this can be rewritten as:

$$Pr(U_k|\mathcal{P},\mathcal{H},M,\Psi_M) \propto Pr(\mathcal{P}|U_k,\mathcal{H},M,\Psi_M)Pr(U_k|M,\Psi_M) \tag{5.1}$$

In this equation, $Pr(U_k|M,\Psi_M)$ is the prior probability that block $k$ of model $M$ with parameters $\Psi_M$ contains a locus which affects the observed phenotypes, while $Pr(\mathcal{P}|U_k,\mathcal{H},M,\Psi_M)$ is the posterior probability of phenotypes $\mathcal{P}$ given haplotypes $\mathcal{H}$ under that assumption.

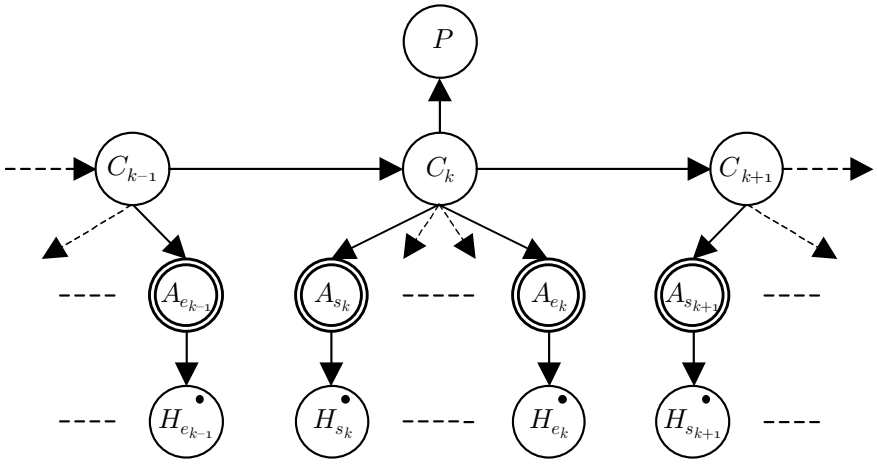



Figure 5.1: Bayesian Network for mapping haplotypes

Phenotype information is expressed as the variable $P$ in our model. Under hypothesis $U_k$, $P$ is directly dependent only on variable $C_k$, as depicted in Figure 5.1. This simple dependency is sufficient because the differences in ancestry reflected by variable $C_k$ capture the ancestral variation at all loci within block $k$, including those which are not observed.

We approximate the term $Pr(\mathcal{P}|U_k,\mathcal{H},M,\Psi_M)$ of Equation 5.1 by assuming sample independence and inferring maximum likelihood parameters for $Pr(P|C_k,M,\Psi_M)$. These parameters are

obtained using the EM algorithm with the haplotypes $\mathcal{H}$ and phenotypes $\mathcal{P}$ as evidence [68]. The subsequence of each haplotype for block $k$ is usually compatible with only one value of $C_k$, so the EM algorithm converges uniquely and quickly.

The prior probability $Pr(U_k|M, \Psi_M)$ of Equation 5.1 is based on two elements. The first element assigns probability in proportion to $V_k$, the length of block $k$. The second element adjusts for the fact that blocks with more ancestors have more parameters for maximizing the likelihood $Pr(\mathcal{P}|U_k, \mathcal{H}, M, \Psi_M)$. We compensate by considering the optimal number of bits $W_k$ required to represent $Pr(P|C_k, M, \Psi_M)$. Using a standard encoding, $W_k = \frac{q_k}{2}(p_{max} - 1)\log_2 n$, where $q_k$ is the number of ancestors for block $k$, $p_{max}$ is the number of phenotypes and $n$ is the number of samples observed [110]. Applying the MDL schema, elements $V_k$ and $W_k$ are combined to obtain $Pr(U_k|M, \Psi_M) \propto V_k \cdot 2^{-W_k}$ [109].

### 5.1.2 Genotypes posterior

For genotype data, the posterior distribution $Pr(U_k|\mathcal{P}, \mathcal{G}, M, \Psi_M)$ is obtained in a similar manner as for haplotypes. Equation 5.1 is trivially rewritten as:

$$Pr(U_k|\mathcal{P}, \mathcal{G}, M, \Psi_M) \propto Pr(\mathcal{P}|U_k, \mathcal{G}, M, \Psi_M) Pr(U_k|M, \Psi_M) \tag{5.2}$$

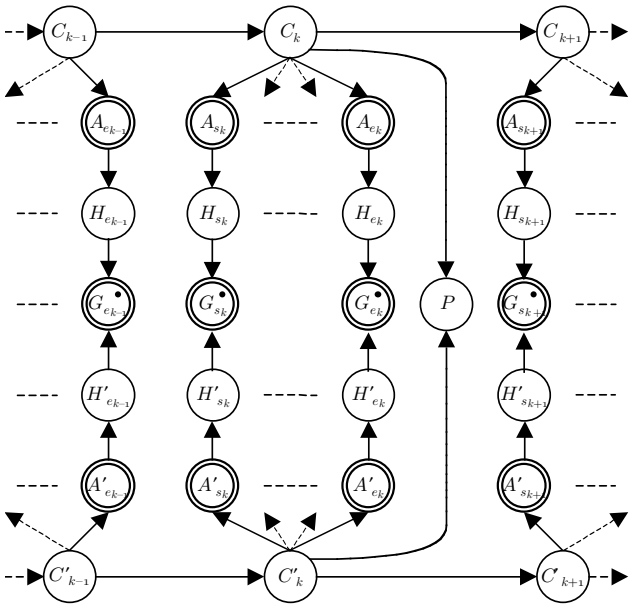



Figure 5.2: Bayesian Network for mapping genotypes

As before, we represent phenotype information as the variable $P$ in our model. For dominant, recessive and codominant disease models, the phenotype is affected by genetic variation in both chromosomes. Therefore, under hypothesis $U_k$, $P$ depends on both variables $C_k$ and $C'_k$, as depicted in Figure 5.2. The differences between haplotype and genotype posterior calculations stem only from this more complex dependency.

Element $Pr(\mathcal{P}|U_k, \mathcal{G}, M, \Psi_M)$ of Equation 5.2 is calculated as before by assuming sample independence and inferring the parameters of $Pr(P|C_k, C'_k, M, \Psi_M)$ by EM. This distribution is symmetrical for the two variables $C_k$ and $C'_k$, reflecting the functional symmetry between the maternal and paternal chromosomes in a cell.

The prior probability $Pr(U_k|M, \Psi_M)$ of Equation 5.2 is also calculated as before, based on the length $V_k$ and the number of bits $W_k$ required to represent $Pr(P|C_k, C'_k, M, \Psi_M)$. Since the

distribution $Pr(P|C_k, C'_k, M, \Psi_M)$ is symmetrical, we set $W_k = \frac{q_k \cdot (q_k+1)}{4}(p_{max} - 1)\log_2 n$. The two elements are combined as before so that $Pr(U_k|M, \Psi_M) \propto V_k \cdot 2^{-W_k}$.

## 5.2 Results

### 5.2.1 Full penetrance haplotype mapping

We assessed our mapping technique using two large sets of empirically determined human haplotypes: (a) 258 transmitted haplotypes of 98 SNPs over 464 kb in the 5q31 region [16], (b) 20 haplotypes of 24,047 SNPs over the whole of chromosome 21 [94]. These data sets are described in greater detail in Chapter 1.

Each test set was generated from a set of haplotypes by randomly selecting a target SNP to be converted into phenotype information. Each haplotype was assigned the phenotype corresponding to the allele it possessed for this SNP, which was then removed from the marker data – the goal of the mapping algorithm was to recover its location. Since all SNPs were biallelic, haplotypes which had the more common allele for the target SNP were labeled as 'healthy' while the others were labeled 'diseased'. This mirrors the LD mapping problem for high penetrance diseases, where a hidden locus which determines phenotypic differences must be found.

For the 5q31 data, we created five separate test sets, selecting SNPs as the target with probability in proportion to the distance between their neighboring SNPs. For chromosome 21, we used 5 randomly selected contiguous subsets of 201 SNPs from the NT002836 contig, then created a single test set from each subset as before. We removed those few haplotypes from test sets for which the target SNP allele was unknown.

For each test set, we obtained the distribution $Pr(L_j|\mathcal{P}, \mathcal{D})$ by inferring an ensemble of 100 models. For comparison, we also obtained posteriors from the BLADE algorithm, allowing it to optimize the number of founders using the MAP criterion [70]. We further calculated a distribution using a version of our model with no inter-locus dependencies, considering each SNP individually as an independent 'block'. We tried to include three other software packages in our comparison, however each proved unobtainable or unsuitable for data sets with a large number of SNPs [66, 80, 106].

Table 5.1: Mapping results for full penetrance haplotype tests

| Data set and SNP range | Target SNP | Individual | | BLADE | | HaploBlock | |
|---|---|---|---|---|---|---|---|
| | | Rank | Sequence | Rank | Sequence | Rank | Sequence |
| 5q31 | 3 | 1 | 7 kb | 8 | 71 kb | 3 | 7 kb |
| | 7 | 5 | 43 kb | 1 | 80 kb | 1 | 68 kb |
| | 21 | 5 | 14 kb | 18 | 17 kb | 1 | 5 kb |
| | 80 | 69 | 336 kb | 54 | 277 kb | 7 | 111 kb |
| | 84 | 54 | 255 kb | 9 | 273 kb | 1 | 9 kb |
| | Mean | 13 | 131 kb | 9.6 | 144 kb | 2.6 | 40 kb |
| Chr 21:3877–4077 | 4063 | 2 | 2 kb | 114 | 140 kb | 3 | 12 kb |
| 8538–8738 | 8597 | 28 | 101 kb | 7 | 20 kb | 1 | 17 kb |
| 15510–15710 | 15607 | 1 | 17 kb | 104 | 267 kb | 1 | 24 kb |
| 15855–16055 | 15870 | 2 | 8 kb | 9 | 52 kb | 1 | 10 kb |
| 16807–17007 | 16918 | 36 | 38 kb | 27 | 60 kb | 33 | 57 kb |
| | Mean | 13.8 | 33 kb | 16.4 | 107 kb | 7.8 | 24 kb |

Table 5.1 lists the results for each test set. For each algorithm, the first column shows the position of the interval containing the target SNP, in a ranking of intervals according to their

posterior probability. The ranking compared 96 intervals for the 5q31 data, and 199 intervals for each chromosome 21 test set. Note that larger intervals rank higher under any algorithm, so this statistic is not ideal for comparative study.

To generate a better statistic, we used the posterior density of each interval (i.e. $Pr(L_j|\mathcal{P},\mathcal{D})/d_j$) to determine a resequencing prioritization. We assumed that SNP intervals would be resequenced in descending order of posterior density until the target SNP was found. The second column for each algorithm shows how much of the candidate region would have to be resequenced under this scheme. In the absence of any mapping information, we would expect this to be half of the region's length, i.e. 232 kb for the 5q31 data and 99 kb, 82 kb, 248 kb, 167 kb and 201 kb respectively for each of the chromosome 21 test sets.

In 6 out of the 10 tests, our HaploBlock algorithm ranked the target SNP interval first, whereas the individual SNP and BLADE approaches did so twice and once respectively. In terms of the resequencing prioritization, HaploBlock also comfortably outperformed the other two approaches. This is particularly notable for the 5q31 region, in which it required an average of 40 kb instead of 131 kb and 144 kb, saving around 70% in resequencing costs.

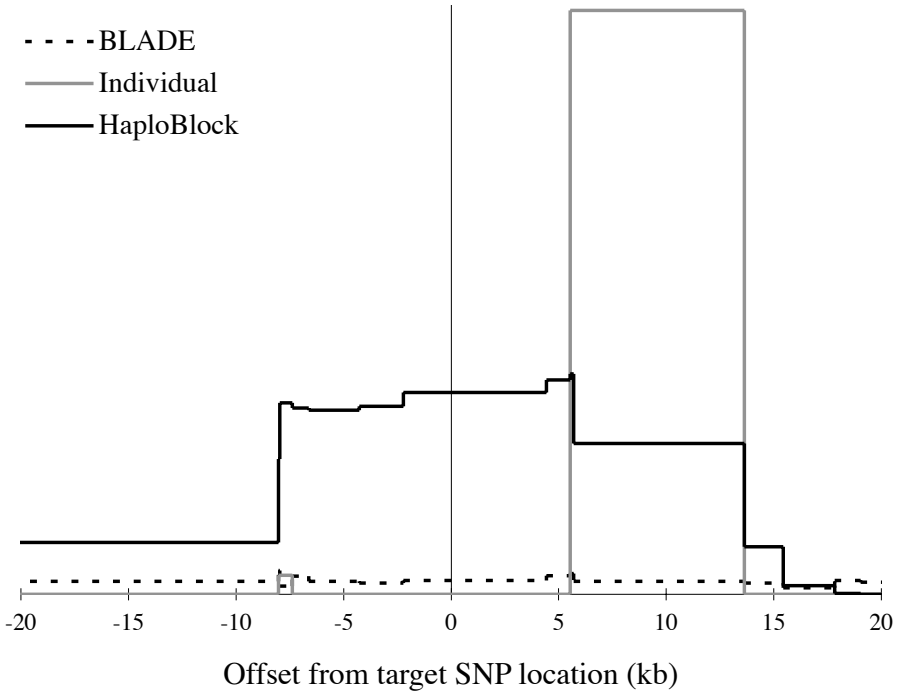



Figure 5.3: Posterior densities for SNP 21 in haplotype data set 5q31

It is instructive to examine the results for the 5q31 data set with target SNP 21, in which all three algorithms performed reasonably well. Figure 5.3 depicts the posterior density curve assigned by each algorithm in the immediate vicinity of the hidden target SNP. The BLADE algorithm failed to find any significant peak in this area, although it did assign a posterior density to a 50 kb window containing the target which was slightly higher than in the rest of the region. The individual SNP method assigned a peak window between SNP 23 (5.5 kb downstream of target) and SNP 25 (13.5 kb downstream), reflecting a strong assocation between the phenotypes and SNP 24. While close by, this window failed to include SNP 21, since both SNPs 22 and 23 were poorly correlated with the phenotypes. By contrast, HaploBlock assigned a wider peak which was well centered around the target, reflecting its location within a block whose haplotypes were strongly associated with the phenotypes. It is interesting to note that the original 5q31 analysis assigned a block from SNP 16 (8 kb upstream of target) to SNP 24 (6 kb downstream) [16]. Similarly, 84 of the 100 models sampled by HaploBlock placed SNPs 16 to 23 in a single block.

HaploBlock performed relatively poorly in terms of resequencing length for SNPs 7 and 80 in the 5q31 region and the last test set for chromosome 21. In all three cases, the target SNP was strongly associated with several haplotype blocks in the surrounding region, reducing the resolution that HaploBlock was able to achieve. Nonetheless, it is encouraging to note that the target was always included in the window of high posterior density output by HaploBlock, while this was not the case for the other two approaches (graphs not shown).

### 5.2.2 Genotypes and partial penetrance

We also assessed the effectiveness of our LD mapping method using unphased genotype marker measurements and/or a partial penetrance model. We based the genotype tests on the 129 offspring in region 5q31, while the haplotype tests used the same 258 haplotypes from 5q31 as before. The chromosome 21 data was not used since it contains too few samples for partial penetrance mapping to be viable.

For a phenotype with penetrance $p$, the disease status was assigned with probability $p$ to haplotypes with the rare allele for the target SNP, while all others were assigned healthy. For genotypes, this model was applied independently to both alleles before combining the results under a codominant model to generate 3 phenotype assignments.

Table 5.2: Mapping results for HaploBlock for genotype and partial penetrance tests

| Data type | | Index of target SNP in 5q31 data set | | | | | |
|---|---|---|---|---|---|---|---|
| (statistic) | Penetrance | 3 | 7 | 21 | 80 | 84 | Mean |
| Haplotypes | 100% | 3 | 1 | 1 | 7 | 1 | 2.6 |
| (rank) | 50% | 3 | 1 | 1 | 10 | 3 | 3.6 |
| | 25% | 3 | 1 | 1 | 4 | 17 | 5.2 |
| | 10% | 3 | 1 | 13 | 16 | 11 | 8.8 |
| Genotypes | 100% | 3 | 1 | 2 | 7 | 3 | 3.2 |
| (rank) | 50% | 5 | 1 | 2 | 17 | 11 | 7.2 |
| | 25% | 5 | 1 | 2 | 18 | 16 | 8.4 |
| | 10% | 5 | 1 | 3 | 21 | 11 | 8.2 |
| Haplotypes | 100% | 7 kb | 68 kb | 5 kb | 111 kb | 9 kb | 40 kb |
| (sequence) | 50% | 66 kb | 68 kb | 5 kb | 117 kb | 42 kb | 60 kb |
| | 25% | 78 kb | 68 kb | 5 kb | 244 kb | 51 kb | 89 kb |
| | 10% | 78 kb | 68 kb | 104 kb | 229 kb | 109 kb | 118 kb |
| Genotypes | 100% | 7 kb | 55 kb | 5 kb | 76 kb | 42 kb | 37 kb |
| (sequence) | 50% | 123 kb | 68 kb | 5 kb | 133 kb | 130 kb | 92 kb |
| | 25% | 39 kb | 55 kb | 13 kb | 217 kb | 179 kb | 100 kb |
| | 10% | 61 kb | 87 kb | 13 kb | 237 kb | 167 kb | 113 kb |

Table 5.2 compares the results of mapping haplotypes and genotypes with varying degrees of penetrance. The results show that our approach remains effective in the absence of phasing information. For genotypes with full penetrance, a mean rank and resequencing length of (3.2, 37 kb) was achieved, compared to (2.6, 40 kb) for haplotypes. Furthermore, our technique exhibits a similar deterioration in performance for haplotypes and genotypes, achieving (8.8, 118 kb) and (8.2, 113 kb) respectively at 10% penetrance.

On a 2 GHz Pentium IV workstation, HaploBlock took about 15 minutes of CPU time to analyze each chromosome 21 test set (200 SNPs, 20 haplotypes) and about 3 and 40 hours respectively for

each set of 5q31 haplotypes and genotypes (97 SNPs, 258 haplotypes or 129 genotypes).

## 5.3 Discussion

Although we demonstrated our method using real-world haplotypes and genotypes, we were forced to simulate phenotypes using a target SNP, since we could locate no publicly available data sets which combine high density SNP data with phenotype information. We wish to apply our approach to such data in future, either as part of a new mapping study or to confirm the effects of a locus whose position is known.

The experiments performed were based on a model in which phenotypes were affected by a single locus in the region of interest. However, it is expected that LD mapping techniques will also prove useful for mapping complex diseases, in which phenotypes are the product of interactions between multiple loci as well as non-genetic factors. To fully address this problem, our model would have to be extended to allow multiple haplotype blocks to influence the phenotypes, via an explicit model of interaction that reduces the number of parameters to be inferred. Nonetheless, the results for the partial penetrance tests indicate that our method is already useful for individually detecting loci with simple additive or multiplicative interactions.

We described a mapping method which uses a full set of SNP measurements taken from a group of subjects. However, it is hoped that haplotype blocks will lead to cost savings in LD studies by reducing the number of SNP measurements required [148, 51]. A pilot study is initially performed on a few subjects, from which the structure of haplotype block variation is inferred. Haplotype tagging SNPs (htSNPs) are then selected to identify the common variants within each block [115, 126]. Measurements taken at these htSNPs from the full set of subjects are extrapolated into full haplotypes based on the pilot study. Our statistical model could be applied to this strategy, using the full SNP measurements taken in the pilot study to infer an ensemble of models. The most informative SNPs in the context of this ensemble would then be chosen as the htSNPs. Measurements taken at these htSNPs would be used with our technique by setting the alleles at all other SNPs to be unknown. Since our Bayesian Network model deals naturally with any number of unobserved variable values, ancestry would be inferred from the htSNPs as intended and the unmeasured loci would be ignored.

# Chapter 6

# Recombination in Viruses

## Introduction

Our statistical model and learning algorithms were designed to infer haplotype blocks from SNP marker data. However with minor modifications they can also be used to analyze raw genetic sequence data from regions with uneven recombination structure. This chapter applies our work to assess the potential role of hotspots of recombination in generating the variability observed in K1, a unique gene in Kaposi's sarcoma-associated herpesvirus (KSHV). KSHV is implicated in all forms of Kaposi's sarcoma (KS), now the most common tumor in HIV positive individuals [123].

K1 lies at the far left hand end of the 140kb double-stranded DNA of KSHV, and is highly variable compared to the rest of the virus, with a preponderance of amino acid altering (non-synonymous) mutations [91, 46, 78, 154]. It has recently been shown that at an individual codon level, specific sites in K1 appear to undergo a considerably greater positive selective pressure than sites in other highly variable mammalian or viral genes, yet there are no known mechanisms to explain this [122]. K1 does not appear to change over time within an individual, nor does it differ between different tumor sites within the same patient, unlike retroviruses such as HIV [124, 138].

Section 6.1 describes the method we used to analyze 269 raw K1 genetic sequences, by converting them to haplotypes. Section 6.2 describes the results of this analysis, in terms of hotspot strength, block diversity and cumulative mutation rates over the length of the K1 gene. This section also describes a similar analysis performed on a different set of viruses with high diversity, yielding strongly contrasting results. Finally, Section 6.3 considers the consequences of our discoveries for K1 as well as some wider implications.

This research was performed in collaboration with Justin Stebbing at the Department of Immunology, Imperial College of London, and published in the *Journal of Molecular Evolution* [38, 39].

## 6.1 Method

Nucleotide sequences encoding the KSHV K1 open reading frame were obtained from NCBI GenBank. These were derived from nested PCR reactions [12, 13, 64, 65, 67, 82, 83, 145, 152]. No two K1 sequences were identical and all 269 K1 sequences were derived from different hosts.

Traditionally, it has been considered that KSHV can be subdivided into strains according to the K1 sequence, which is thought in turn to correspond to geographical origins of the virus. The KHSV A strain is found in Northern Europe and America, the B strain (thought to be the most ancient) is from Africa and the C strain, often associated with classical Kaposi's sarcoma, is found in Mediterranean countries [12]. A KSHV D strain containing nucleotide insertions has also been recently described from Pacific Islands [102, 153]. The evolution and changes in these strains are thought to reflect patterns of migration commencing in Africa [46, 122].

However, sequences in different strains are often closer than two sequences of the same strain. This may reflect recent events where travellers contribute to the spread of viral diversity worldwide, an important contributing factor being world migration of rural populations due to poverty, famine and wars [75, 99, 104]. Approximately 30% of the sequences we analyzed could not be placed in defined strains A-D. Therefore, unlike previous analyses of K1 variability and evolution [122], K1 was not divided into strains and no sequences were excluded.

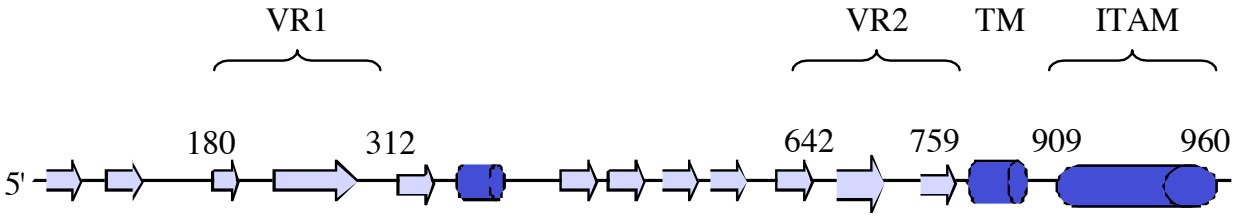



Figure 6.1: Predicted secondary structure for K1

The 269 K1 sequences were converted to amino acids and these were used to create a consensus sequence using MultAlin [14]. This consensus was used to predict secondary structure with PSIPRED, as shown in Figure 6.1 [52, 76, 79]. The variable regions (VR1 and VR2) are repeated strand-helix-strand motifs while the immunoreceptor tyrosine-based activation motif (ITAM) and transmembrane domains are coils. Consensus sequences from K1 derived from each KSHV strain show no significant differences in their structure in spite of amino acid variability.

The K1 genomic sequences were prepared by multiple alignment using the ClustalW algorithm [131]. The results of the alignment were converted to haplotypes by converting each column in the alignment to a marker, and each row to a haplotype sequence over the markers. The nucleotide (or gap symbol) at each position in the alignment was converted to the allele for the respective haplotype and marker. Columns at which no variation was observed were excluded from the haplotypes, since they are of no statistical interest.

We sampled a set of models to produce the minimum length description of the aligned sequences, using the methods described in Chapter 3. The only parameter required by the model search is the maximum cumulative mutation rate, which constrains the distributions for variables $H_j$. In general, as we allow more mutation, less recombination will be inferred. We chose to use three different maximum mutation rates of 0.5, 0.1 and 0.01, to allow different degrees of variation within the offspring of each inferred ancestor. Values greater than 0.5 are meaningless in the context of our technique, since we have no basis on which to infer that a particular allele belongs in an ancestor sequence if a different allele usually appears in its place.

There are four key differences between our model-based approach and traditional phylogenetic tree construction. Firstly, by inferring specific recombination points, our model divides sequences into contiguous stretches, examining the relationships separately within each. This is justified by the observation that a region of high recombination will result in the areas either side having different evolutionary histories. Secondly, we explicitly allow for the presence of mutation hotspots, inferring their presence as part of a model. This is consistent with the observation that mutation occurs in an uneven fashion within the K1 gene and appears clustered in two areas, termed variable regions 1 and 2 (VR1 and VR2). Thirdly, within each inferred stretch, we do not seek to create a complete family tree, accepting instead that distant relationships between sequences are difficult to accurately ascertain and recover. This approach is justified by population genetic considerations which suggest that bottlenecks, genetic drift and selection pressures will narrow a population's genepool, losing the vast majority of ancient strands. The fourth and final difference is that within each such group, we do not attempt to infer relationships between the sequences, opting instead to consider them all as offspring of a single founding ancestor. As before, this is justified by population genetics – if a viral population grows rapidly from a few founders then the most recent common ancestor (MRCA) of any two contemporary sequences is likely to be very close to those founders. In summary, whereas traditional phylogenetic analysis attempts to create a complete tree topology to relate the observed sequences, we infer a set of disconnected stars, each of which centers around a consensus sequence which may itself remain unobserved.

## 6.2 Results

For each of the three maximum mutation rates, we chose to sample 100 models, from which we calculated the mean values of our summary statistics. Each sampled model contains a full description of the variation structure of a set of observed sequences. For our purposes here, the most important parameters of the model are: (a) the location of the recombination hotspots (or block boundaries), (b) the number of inferred ancestors for each stretch between hotspots, and (c) cumulative mutation rates for each site. The cumulative mutation rates represent the probability that the allele observed in a sequence is different from the allele in the ancestor from which it is descended. It should be noted that the full model also describes the linkage dependencies between stretches which are separated by recombination hotspots but we will not be using that information here. By examining

an inferred sample of suitable models, we identify recombination hotspots within conserved and non-conserved areas of K1, thus postulating one mechanism by which it generates its remarkable variability. Each iteration of the sampling algorithm took up to 3 hours of processing time on a 2 Ghz Pentium Xeon workstation, leading to a total running time of several weeks.

### 6.2.1 Hotspot Strength

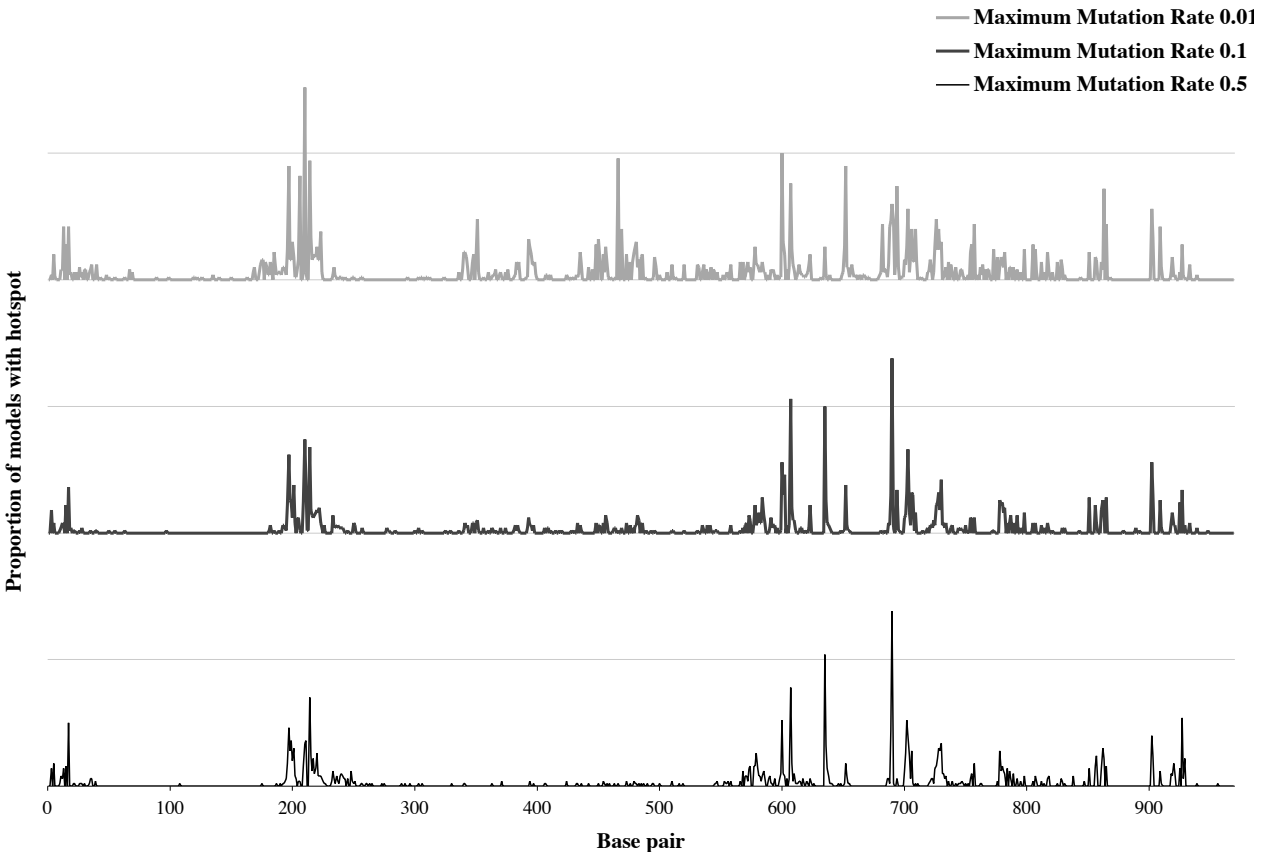



Figure 6.2: Proportion of each set of inferred models with a hotspot at each site

For each of the 3 maximum mutation rates and for each individual base pair, Figure 6.2 demonstrates the proportion of the 100 sampled models which placed a recombination hotspot (i.e. block boundary) at that site. The midlines on each graph represent the point at which 50% of the inferred models have a hotspot, so any peak that reaches or is close to this point is a likely position of a recombination hotspot. High areas which are more spread out suggest a region in which there is a hotspot whose exact location is unclear. Low areas near the zero-line represent regions in which it appears that no recombination hotspots are present. As expected, the less mutation allowed in the model, the more recombination is inferred.

For the 2 higher mutation rates (0.1 and 0.5), codons 212 (base pair 616) and 230 (base pair 690) were identified as recombination hotspots. Base pair 616 is located 27 nucelotides upstream of the second variable region (VR2) of K1. While VR2 is an area characterised by insertions, deletions and non synonymous mutations, base pair 616 is in a relatively conserved area of this gene. Base pair 690 is located mid-way within VR2 itself. At a maximum mutation rate of 0.1, a further site was identified at base pair 606 in the most conserved area of K1 between the variable regions. At these mutation rates, no sites in or around VR1 were identified as recombination hotspots in spite of the known positive selection occurring here ($> 85\%$ of nucleotide substitutions in this region lead to amino acid changes). At the lowest mutation rate (0.01), base pair 606 (codon position

202) was also identified as a likely recombination hotspot. The highest likelihood of recombination was found with a maximum mutation rate of 0.01 at base pair 210 (codon 70), located within the hypervariable area of the first variable region (VR1). VR1 was not flanked by recombination hotspots at any mutation rate.

As groups of small peaks in one area suggest a likely hotspot without an exact site, our data also provides evidence of recombination occurring near the start codon, within VR1, following VR2 and, interestingly, within the cytoplasmic ITAM motif (base pairs 909 to 960). There were no recombination hotspots within the transmembrane region (Figure 6.1).

### 6.2.2 Block Diversity

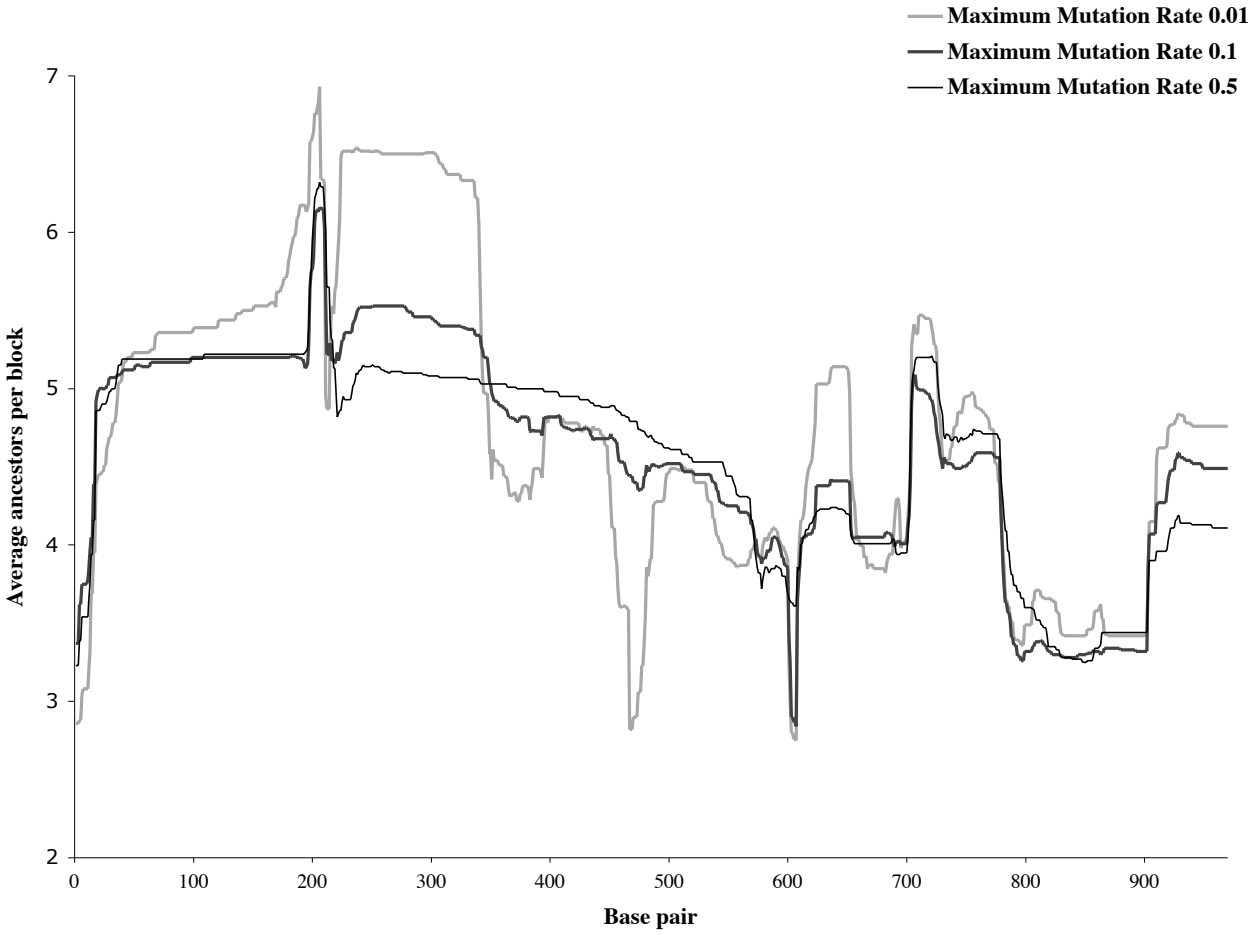



Figure 6.3: Average number of ancestors inferred for block containing each site

Clades of shared ancestry can be visible because of bottlenecks and genetic drift, both of which serve to reduce the variability within each region of low recombination. Although we do not know when KSHV or ORF-K1 first appeared or underwent significant reductions in variation, we can assume that such formative processes have taken place in the past.

Full ancestral sequences among the K1 pool are unknown, and it is unlikely that any are present since they can be expected to have recombined out of recognition. However, for each block between recombination hotspots, our model infers the number of ancestral sequences that appear to be present for that particular block. This is converted to a value for each base pair, by endowing each base pair with the same number of ancestors as the block which contains it. For each of the 3 mutation rates described above, Figure 6.3 shows the number of ancestors inferred for each base

pair, averaged over the 100 samples. As for recombination, the less mutation inferred, the less ancestors are required to explain the observations.

A reduction in clades within a certain region may reflect greater selection in that region. We observe this reduction at the 5' end of K1, around base pairs 469 and 607 (between VR1 and VR2) and in a stretch of nucleotides between base pair 787 and 909. Selective pressures here are negative as they lead to conservation of nucleotides as opposed to variability.

Results were similar at all 3 mutation rates with variation around a mean of 4 ancestors. The highest number of putative ancestors (6 to 7 ancestors) are located in the hypervariable area of VR1. Other peaks are located within and beyond VR2 and the lowest number (3 ancestors) are located in the relatively conserved area between VR1 and VR2 and near the start codon.

### 6.2.3 Cumulative Mutation Rates

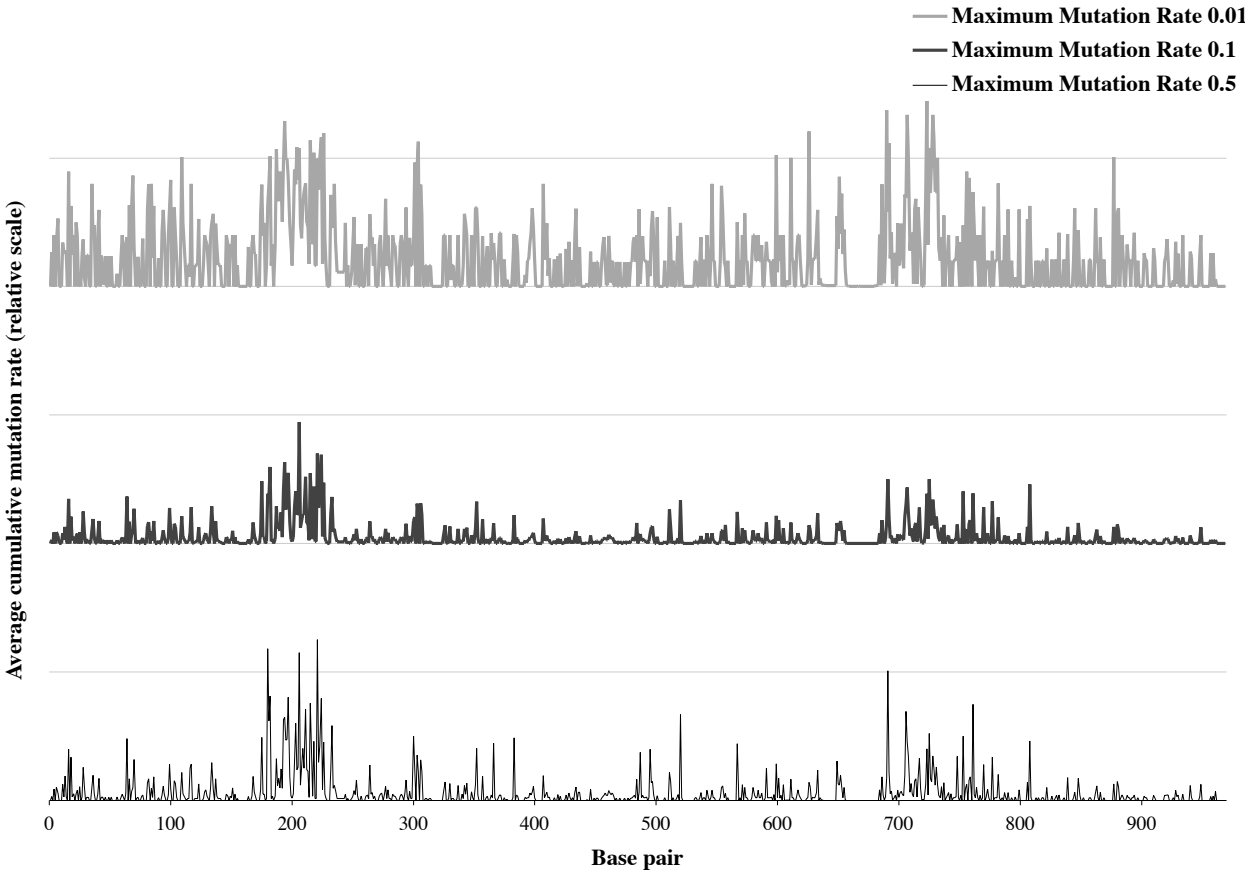



Figure 6.4: Average cumulative mutation rates at each site for each set of models

For each of the 3 mutation rates described above and for each pair, the average cumulative mutation rate over the 100 sampled models is shown in Figure 6.4. Although each model specifies the full allele-to-allele cumulative mutation matrix for each site, we report only on each site's overall cumulative rate of mutation, as calculated from this matrix and the ancestor distribution. The scale is the same for maximum mutation rates 0.1 and 0.5, with the top line for 0.01 magnified by a factor of ten. As expected, the highest probability of mutation is observed within VR1, less so within VR2. There are small peaks suggesting mutation within the conserved areas between VR1 and VR2, but none within the ITAM motif.

### 6.2.4 Comparison with Picornaviruses

Recently, Culley *et al.* described the high diversity of unknown picorna-like viruses in the sea [15]. They analyzed the sequences of multiple picornavirus RNA-dependent RNA polymerases and produced unknown phylogenies that fell outside established picorna-like families. Instead, the polymerase sequences were located in 4 distantly related groups (termed A, B, C and D) with high intra-clade sequence conservation (97.7-100%) and low sequence identity between clades (38.9-54.6%). As their results indicate a diverse but previously unknown community of persistent and widespread viruses, we applied our technique to establish whether recombination may have a role in generating the observed diversity in these single stranded RNA viruses that have a double stranded RNA step in their replication cycle.

We obtained the sequences of the 22 RNA samples used by Culley *et al.* from GenBank and constructed a multiple alignment using ClustalW as before. The allelic distribution at variable sites within this multiple alignment was examined for evidence of recombination using our model sampling technique. To account for several possible population models, we applied three different maximum cumulative mutation rates of 0.5, 0.02 and 0.001, representing a spectrum of probabilities of a mutation having taken place at single nucleotide sites since the ancestral viruses were present in the population.

In each case, the model sampling procedure provided no evidence for the presence of any recombination events. In other words, for each maximum mutation rate assessed, all models sampled placed all of the variable sites within a single block. Four ancestral clades were consistently identified for this block, corresponding to the four picorna-like virus families newly identified. This result appears in stark contrast to that obtained for K1, from which we inferred several points of recombination with a high degree of confidence. The negative result for the picornavirus data provides additional validation for our K1 results.

## 6.3 Discussion

This study enables comparisons to be drawn in genealogical history between different regions of K1. Recombination hotspots are identified in K1 at both conserved and unconserved nucleotide positions. The mechanism of recombination at the different sites may therefore involve separate mechanisms. The processes of generating variability in this DNA viral gene are termed recombination shift or drift based on the time scale in which they are postulated to affect the viral sequence [31, 137]. Recombination shift involves changes that affect variable positively selected sites wherein immediate effects will be evident. Recombination drift is thought to occur when homologous recombination occurs in conserved regions, resulting in longer term sequence changes over generations.

Amongst the challenges of the analysis of the role of recombination in evolution is the detection and estimation of recombination in genomes where the rate of substitution is sufficiently high that some sites have experienced multiple mutational events. Viral genes evolve at a high speed compared with genes of higher organisms and hence viral evolution provides interesting material for the study of molecular evolution by recombination. Although recurrent mutations in viruses can generate patterns of genetic variability that resemble the effects of recombination [81], our model inference technique adopts the more suitable explanation for each region of the observed data. We used different maximum mutation rates, comparing the results of allowing different degrees of mutation. We observed a high degree of consistency between the models, especially between mutation rates 0.1 and 0.5, suggesting that our conclusions are independent of whether KSHV is an ancient [46] or relatively recent pathogen [111]. We do not know however when KSHV was introduced and it is likely that many of its genes have been acquired, presumably by recombination from the host genome over time [89, 91]. Since then, these genes have apparently evolved to facilitate viral survival [91]. However, while most KSHV open reading frames have known homologs or at least suggested homologs, BLAST searches of non-recombinant stretches of K1 (and K15; data not

shown) reveal no sequences that suggest a pirated origin from human genes.

PCR based studies examining the predominant and minor forms of K15 have demonstrated evidence for recombination within KSHV. The first of these used classical linkage analysis and the criterion of lack of co-segregation at multiple genetic loci. This led to the hypothesis that an original recombination event occurred that introduced exogenous sequences from a related primate virus of unknown source and that subsequent mutations led to certain KSHV lineages [102]. A conflicting study showed that the proposed introduction of these exogenous sequences did not occur via a single recombination event [54]. Overall however, analysis of K15 sequences from individuals within the same family provides evidence for recombination in approximately 20-30% of cases.

Positive and negative selective pressures influence nucleotide changes within all genes that change or preserve them respectively. The neutral theory of evolution predicts that the stronger the selective constraint against nucleotide changes, the lower the rate of base substitutions [57]. This prediction is supported by a large number of observations at the DNA sequence level. For example, the rate of synonymous or silent substitutions that produce no alteration in translated proteins is usually much larger than the rate of non-synonymous substitutions [21]. However, that positive selection in K1 favors change is evident by the large number of nucleotide changes in the middle position of a codon triplet, a substitution always resulting in amino acid alterations. The recombination hotspots within this highly variable gene provide a possible mechanism by which positive selective pressures exert their effects over time.

Herpesviruses have evolved through co-speciation with their hosts [78, 122]. Evasion from all host immune control mechanisms will lead to overwhelming viral infection with subsequent death of the host and therefore the virus. For these viruses to persist as a latent infection without causing harm, an equilibrium between pathogen and host must be established. Unlike the error prone reverse transcriptase of retroviruses, herpesviruses do not have a mechanism that will result in rapid sequence variation. Previous data show that the pressure causing this selection is partly to facilitate immune recognition [122]. As K1 is expressed predominantly in the early lytic cycle of viral replication, a certain level of viral replication occurs prior to immune recognition and the subsequent death of the infected cell. Recombination provides a mechanism here to generate diversity in response to selective pressures that could lead to the attraction of an immune response to this variable oncogene. This would ensure that the virus-host equilibrium is established and that latent infection may be achieved. This effect may be most important when the virus is introduced into a new population group containing, for example, new MHC alleles.

Although no homologs of K1 have been identified, it is possible that the K1 sequences represent divergent forms of key genes that evolved very rapidly with all intermediate forms being lost as each subtype of the virus occupied a new biological niche. Alternatively, conserved areas within K1 may represent relics of older forms of the virus or of related viral species that persist as small areas of their original genomes by virtue of rare recombination events with more modern forms. The continuous expansion of viral diversity over time is influenced by social, behavioral and biological forces [99]. Such biological forces are driven by host immune responses to K1, antiviral drugs, the rapid turnover of virus and by mutation events. In retroviruses, the error prone reverse transcriptase makes significant contributions to these mutation events. Our results suggest that recombination contributes to the extreme diversity of a DNA viral gene.

# Chapter 7

# Blocks and Hotspots

# Introduction

There has been considerable debate in the literature over whether recombination hotspots are required to explain the presence of haplotype blocks. Hotspots clearly provide a *possible* explanation of block-like patterns of variation, since they divide a genomic region into stretches of co-segregating alleles. However, some argue that these same patterns could be generated by genetic drift alone [147, 98]. In this chapter we address this question empirically by analyzing a set of high density marker data taken from a 3-generation pedigree in which recombinations can be detected.

Our statistical model is neutral regarding the connection between hotspots and blocks. However, the question is relevant for the MDL criterion we described in Chapter 3. The description length of a model is based on a full transition matrix for the Markov chain, which assigns a probability to every possible combination of ancestor haplotypes across each block boundary. This representation is sensible if multiple historical recombinations took place at each boundary, since this would cause many ancestor combinations to occur. However if block boundaries are due to just one or two historical recombinations, very few ancestor combinations would appear. In the latter case, a sparse representation for the Markov transition matrices would be more appropriate.

Section 7.1 explains the pedigree SNP data used for our analysis, as well as the techniques applied to detect recombination points and haplotype block boundaries. Section 7.2 details the results of the analysis, quantifying the correlation between recombinations and blocks, and estimating the proportion of recombination events that occur on boundaries. Finally, Section 7.3 provides a brief interpretation of our results and describes how they might be improved.

This research was performed with Richard Durbin on site at the Wellcome Trust Sanger Institute, UK. It was presented in a *Keystone Symposium* [40].

## 7.1 Methods

### 7.1.1 Data Model

Our study was based on a set of high density unphased marker data from chromosome 20, produced as part of the International Haplotype Mapping (HapMap) project [44]. The main data set consisted of genotypes of 33,395 SNPs distributed over a 62.7 Mb region, measured for 12 CEPH families of European ancestry. Each family consisted of four grandparents, two parents and two children (see pedigree in Figure 7.1). A second data set was also used, consisting of genotypes for 33,381 SNPs from 42 unrelated Japanese individuals. The set of SNP markers genotyped for this Asian set was nearly identical to that for the CEPH families.

The maternal grandmother (MGM) was missing in one CEPH family, as was the paternal grandfather (PGF) in another. Excluding these two missing grandparents, the CEPH data was

Figure 7.1: Structure of CEPH families

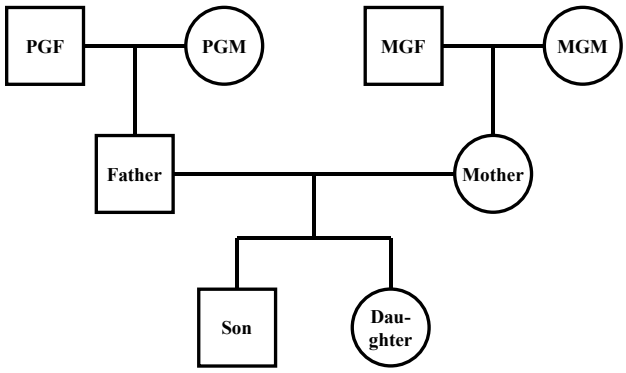
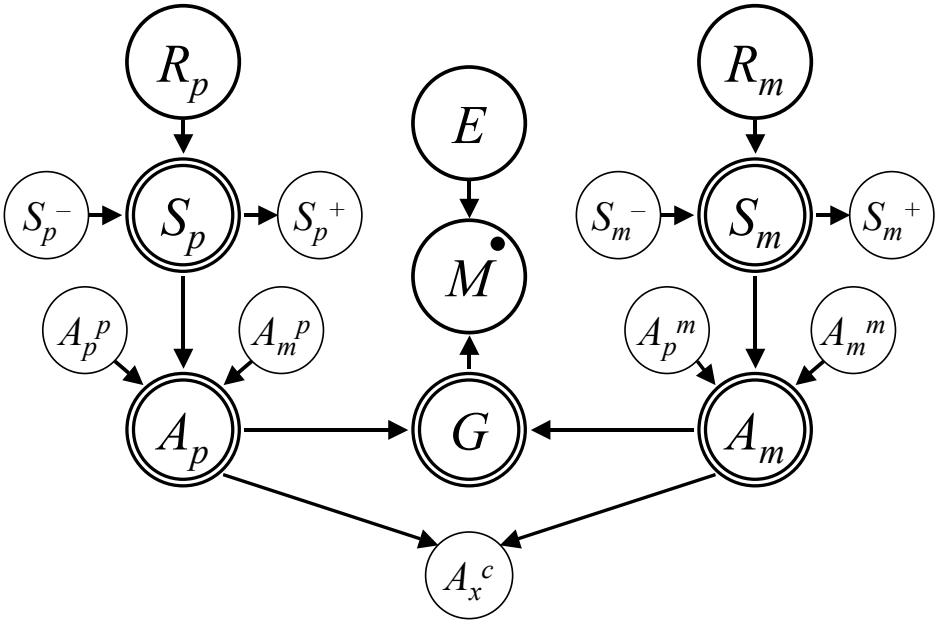


Figure 7.2: Bayesian Network to represent one locus in one individual




characterized by a 1.0% rate of unknowns (for the Asian data, this figure was 0.9%). All genotypes were specified to contain a maximum error rate of 0.3%, where an error is defined as an incorrect genotype measurement for a single marker in an individual. We reestimated this error rate from the observed data, and confirmed the specified bound (results not shown).

The pedigree structure of the CEPH data enabled a near-complete pair of haplotypes to be inferred for the father, mother, son and daughter in each family, by using the genotypes of their respective parents. This process only failed for sites at which the child and both parents in a trio were heterozygous, or a Mendelian error occurred. Recombinations that took place during the creation of each child's chromosomes could also be inferred by noting when the haplotypes passed on by the parent changed origins from one grandparent to the other.

All inferences were performed using a Bayesian Network model to represent the inheritance relationships within the CEPH pedigrees, as well as the possibility of genotyping errors [95, 50]. Bayesian Networks cope naturally with unknowns, allowing failed genotype measurements to be handled naturally as an unassigned variable. The Bayesian Network was queried in a number of ways to infer the desired information from the genotype evidence, with all calculations performed by bucket variable elimination [17].

Figure 7.2 shows the part of the Bayesian Network which represents a single locus for a single individual. Smaller circles indicate variables relating to adjacent loci or related individuals. Table 7.1 describes the variables in this Bayesian Network that relate to the individual and locus represented, alongside their possible values and conditional distributions. Bracketed lists in Table 7.1 denote the respective probabilities for each of the variable's values as listed. Variables that relate to other individuals or loci are explained in Table 7.2.

Note that some variables are not required for some loci or individuals. For the first locus examined, $R_p$, $R_m$, $S_p^-$ and $S_m^-$ are absent because there is no previous locus, so the distributions for variables $S_p$ and $S_m$ become $(0.5, 0.5)$. Since the grandparents constitute the founders of the pedigree, all variables topologically prior to $A_p$ and $A_m$ are absent for grandparent loci, and the prior distributions for variables $A_p$ and $A_m$ are set according to the marginal allele frequencies observed in the data.

In total, each locus (except the first) defines 56 variables. A comprehensive model representing the 33,395 SNPs in the CEPH data would contain almost 2 million variables, with the distributions

Table 7.1: Description of variables in Bayesian Network in Figure 7.2

| Symbol | Description | Values | Distribution |
|---|---|---|---|
| $R_p$ | Recombination in father | $0, 1$ | Query-dependent |
| $R_m$ | Recombination in mother | $0, 1$ | Query-dependent |
| $S_p$ | Source grandparent in father | $0, 1$ | If $R_p = 0$ then $S_p = S_p^-$, else $S_p = 1 - S_p^-$ |
| $S_m$ | Source grandparent in mother | $0, 1$ | If $R_m = 0$ then $S_m = S_m^-$, else $S_m = 1 - S_m^-$ |
| $A_p$ | Allele in father | $1, 2$ | If $S_p = 0$ then $A_p = A_p^p$, else $A_p = A_m^p$ |
| $A_m$ | Allele in mother | $1, 2$ | If $S_m = 0$ then $A_m = A_p^m$, else $A_m = A_m^m$ |
| $G$ | Actual genotype | $1, 2, h$ | If $A_m = A_p$ then $G = A_m$, else $G = h$ |
| $E$ | Error occurred in measuring | $0, 1$ | $(0.997, 0.003)$ |
| $M$ | Measured genotype | $1, 2, h$ | If $E = 0$ then $M = G$, else $(0.25, 0.25, 0.5)$ |

Table 7.2: Description of variables for other individuals and loci in Figure 7.2

| Symbol | Description | Values |
|---|---|---|
| $S_p^-$ | Source grandparent for previous locus in father | $0, 1$ |
| $S_p^+$ | Source grandparent for next locus in father | $0, 1$ |
| $A_p^p$ | Allele at this locus in paternal chromosome of father | $1, 2$ |
| $A_m^p$ | Allele at this locus in maternal chromosome of father | $1, 2$ |
| $S_m^-$ | Source grandparent for previous locus in mother | $0, 1$ |
| $S_m^+$ | Source grandparent for next locus in mother | $0, 1$ |
| $A_p^m$ | Allele at this locus in paternal chromosome of mother | $1, 2$ |
| $A_m^m$ | Allele at this locus in maternal chromosome of mother | $1, 2$ |
| $A_x^c$ | Allele at this locus in child, $x = p$ or $m$ depending on my gender | $1, 2$ |

for variables $R_p$ and $R_m$ determined by recombination distances. The Markov-like structure of the model (from one locus to the next) suggests that computations would be feasible but slow. However, models representing individual loci or pairs of nearby loci sufficed to extract the information required for our purposes, as explained below.

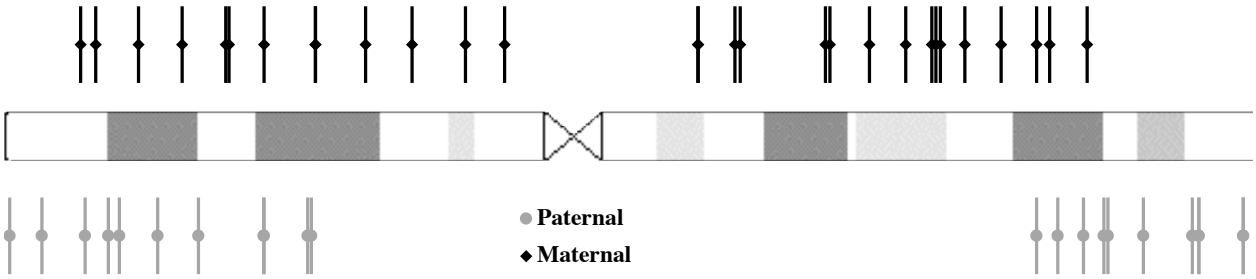
### 7.1.2 Detecting Recombinations

We inferred the location of the recombination events that took place in the chromosomes passed from the CEPH mothers and fathers to their children. First, the posterior distributions $Pr(S_p)$ and $Pr(S_m)$ for each locus in the children's chromosomes were computed, given all of the data observed in the pedigree at that locus. If the grandparental source could be ascertained with 99% certainty, it was considered to be correct, otherwise it was left as unknown. The output of this analysis was examined by eye to locate windows of the children's haplotypes in which the origin switched between the grandfather and grandmother. At this stage, uncertainty over the exact position of the switch arose mostly due to homozygous stretches in the parents from which the chromosome was inherited, obscuring the source grandparent of the inherited alleles. However, some heterozygous sites in the parent were also ambiguous, inviting further effort to resolve their origin.

We targeted each parentally heterozygous locus within these windows, by examining its inheritance pattern together with loci nearby. Each target locus was tested in conjunction with each of the 1,000 closest loci, under the assumption that no recombinations happened in the intermediate region except in the child chromosome of interest. These tests were performed by creating a Bayesian Network model representing the two loci, as described in Section 7.1.1. We assigned a probability of zero to every variable $R_p$ and $R_m$ in the network, except that relating to the child's

chromosome which was given an uninformative recombination prior of 0.5. The posterior distributions $Pr(S_p)$ or $Pr(S_m)$ at the locus of interest were then calculated given the observed data, generating a likelihood assignment for the grandparent of origin. By performing this task for each of the 1,000 loci closest to the target site, a vote was produced regarding its source grandparent. In most cases, the vote was unanimous, so the origin could be clearly assigned. In a few cases where the vote was split due to another recombination having occurred elsewhere in the pedigree, the grandparent was inferred by voting with only the 100 loci nearest the target site.

Figure 7.3: Midpoints of recombination event windows




The grandparental source of almost every parentally heterozygous locus was determined by this method, allowing the recombination event windows to be narrowed further. Nonetheless, many of these windows remained large due to long stretches of parental homozygosity, which hid the origin grandparent of the loci within. We briefly tested the theory that recombination events tend to occur more in long homozygous stretches, but this was not borne out by the data (results not shown). In total, 51 recombination event windows were identified. 29 events occurred in a maternal chromosome, while 22 were paternal, reflecting an expected bias towards maternal recombination. The three largest windows include a 880kb region in which no SNPs were available. The chromosomal locations of the paternal and maternal recombination events were very different, as shown in Figure 7.3. The median window size was 23kb.

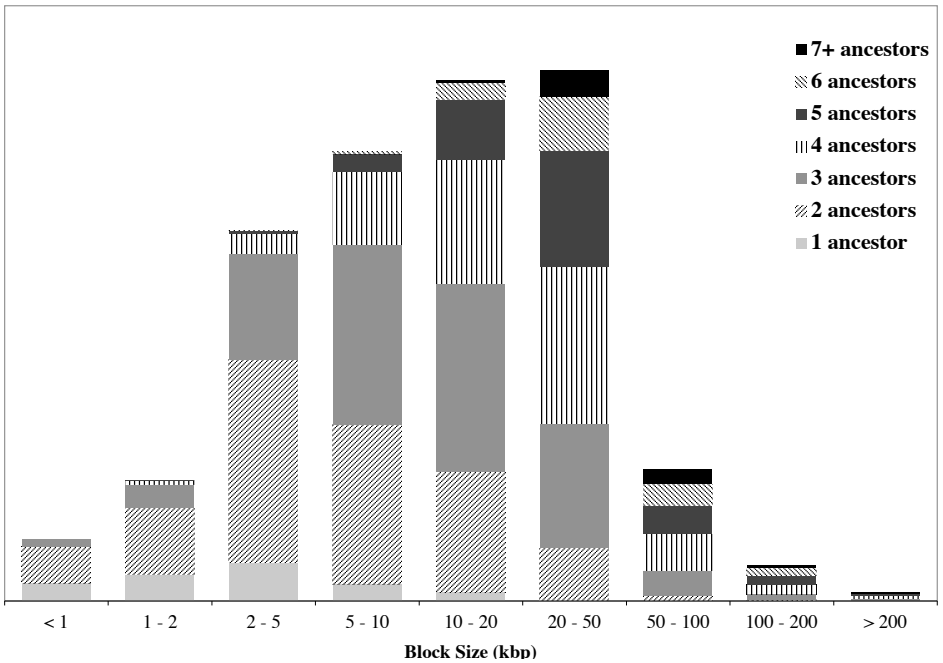
### 7.1.3 Detecting Blocks

We inferred the block structure of the region from the haplotypes of the parents in the CEPH pedigree. First, the parents' haplotypes were obtained by using a separate Bayesian Network for each SNP and computing the posterior distributions $Pr(A_p)$ and $Pr(A_m)$ for each parent given the observed data (see Section 7.1.1). An allele was assigned to a haplotype locus if it had a posterior probability of 0.99 or more, otherwise the locus was left as unknown. Unknowns were due either to heterozygosity in both the grandparents and the parent, or missing genotype measurements.

In total, 48 CEPH parent haplotypes were inferred, with a 7.3% rate of unknown alleles. HaploBlock was used to infer the block structure of these 48 haplotypes, with all input parameters left at their default values. An ensemble of 12 models was obtained, from which the first two were discarded because they had not yet filled out with block boundaries. All statistics were averaged over the remaining 10 models, giving each a uniform prior probability.

The haplotype blocks inferred are summarized in Figure 7.4. As the graph shows, most blocks were between 2 kb and 50 kb in length, and contained 2 to 5 clades co-descended from a single identified ancestor. The density of block boundaries is compared with the density of SNPs in Figure 7.5, averaged for a 2 Mb sliding window over the chromosome. This graph shows that areas with higher SNP density also have more block boundaries, an effect which is likely to be artifactual. However, there are also differences in block density that are not explained by SNP density – for example, the relative block density near the telomeres is higher than the relative SNP density, whereas near the centromere the opposite effect can be seen. This reflects the expected increase in recombination activity towards the edge of the chromosomes, as observed in Figure 7.3.

Figure 7.4: CEPH parent haplotype block summary




Haplotype block boundaries were also ascertained from the CEPH haplotypes using the $|D'|$ linkage disequilibrium method, as implemented in the HaploBlockFinder program [150]. In this case, a block was defined as a stretch in which the $|D'|$ value between every pair of SNPs was above 0.9. Naturally, there are many other block identification algorithms that could also have been used.

Finally, we inferred an ensemble of 12 models from the Asian genotypes using HaploBlock, and the first two models were discarded as before. This enabled us to test the extent to which recombination struture is shared between different human populations. HaploBlockFinder requires haplotypes as input, so it could not be used to apply the $|D'|$ method to the Asian data.

## 7.2 Results

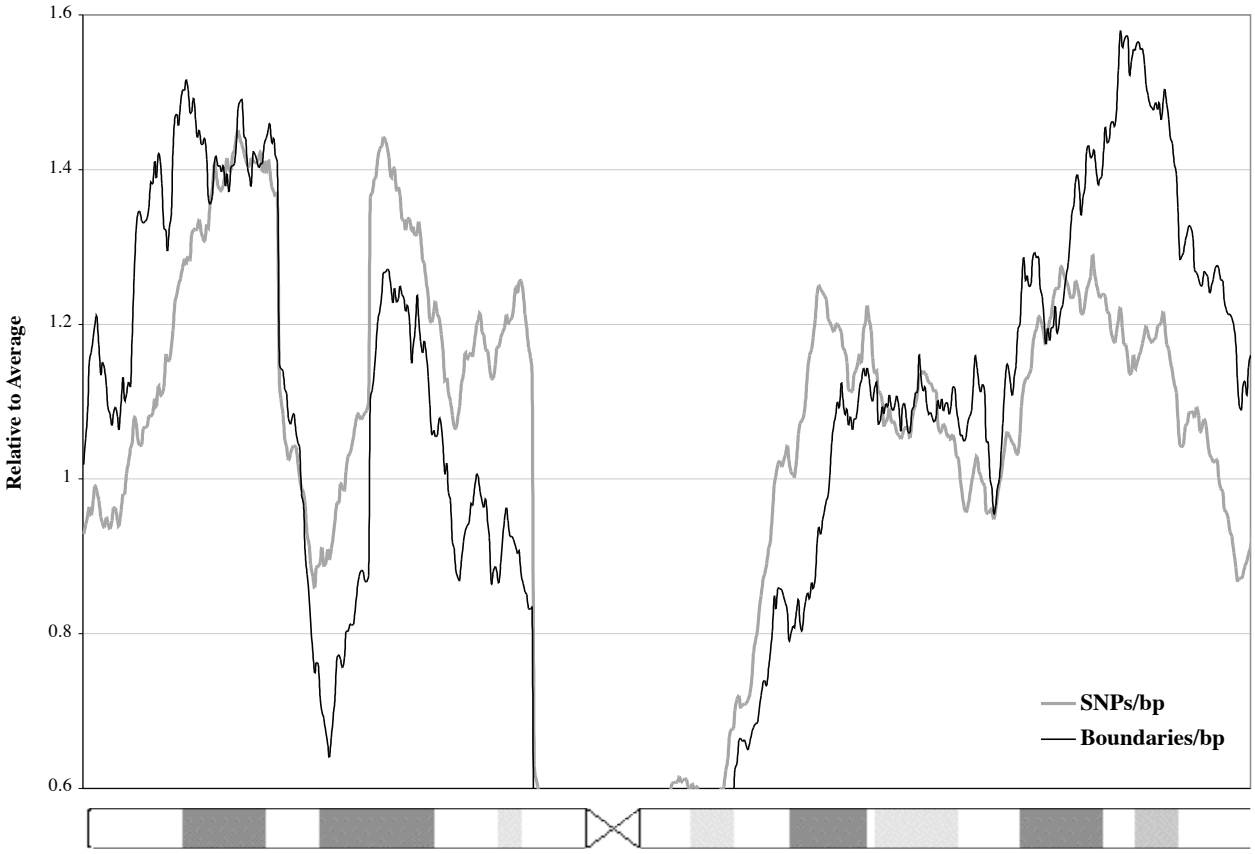
### 7.2.1 Correlation between Boundaries and Recombination

We assessed the extent to which the recombination event windows detected in the CEPH child chromosomes were correlated with the haplotype block boundaries detected in the CEPH parents. Such a correlation would imply that the hotspots responsible for an increase in recombination frequency are also related to the haplotype block phenomenon.

The correlation was measured by comparing the block boundary density for the whole of chromosome 20 against the average density of the 51 recombination windows. The density of a region is defined by the total number of block boundaries within the region, divided by either the number of SNPs in the region or its physical length in base pairs. These calculations yield the respective measurements of boundaries/SNP and boundaries/kb. We also calculated a second pair of density statistics from the conditional entropies of the Markov transition matrices straddling each boundary (see Section 2.1). The entropy density for a region is defined by the sum of the conditional entropy across each block boundary within the window, divided by the number of SNPs or physical length as appropriate.

Table 7.3 summarizes the statistics correlating the recombination event windows with the CEPH block boundaries. On all measures, the recombination windows had a much higher boundary density than the chromosome as a whole. The table also shows P values for the null hypothesis of

Figure 7.5: Boundary and SNP density over chromosome




no systematic correlation between block boundaries and recombination events. The P values were calculated by randomly relocating the event windows a million times, and counting how many of these relocations produced a statistic at least as strong as that observed. During each relocation, the recombination locations were drawn randomly and uniformly over the chromosome, then extended along homozygous stretches in the parent to simulate detectable event windows.

Table 7.3 also shows the degree of correlation between recombination events and block boundaries inferred using the $|D'| > 0.9$ method. As with the HaploBlock-inferred boundaries, the results are highly significant on all measures, suggesting that our results are not overly sensitive to the block partition algorithm used. Figure 7.6 shows an example region which depicts the correlation between block boundaries and two recombination events that were narrowed to within 2 SNP intervals. As the figure shows, both recombination events fall squarely on strong block boundaries, as identified by both HaploBlock and the $|D'|$ method.

Table 7.3: Correlation between CEPH block boundaries and event windows

| Method | Statistic | Chromosome | Event windows | Factor | P value |
|---|---|---|---|---|---|
| HaploBlock | Boundaries/kb $\times 10^{-2}$ | 4.78 | 14.28 | 3.0× | 0.00197 |
| | Boundaries/SNP $\times 10^{-2}$ | 8.97 | 22.85 | 2.3× | 0.00006 |
| | Entropy/kb $\times 10^{-2}$ | 2.40 | 8.51 | 3.5× | 0.00105 |
| | Entropy/SNP $\times 10^{-2}$ | 4.52 | 13.36 | 3.0× | 0.00037 |
| $\|D'\| < 0.9$ | Boundaries/kb $\times 10^{-2}$ | 7.73 | 17.64 | 2.3× | 0.00857 |
| | Boundaries/SNP $\times 10^{-2}$ | 14.53 | 29.51 | 2.0× | 0.00043 |

Figure 7.6: Example of correlation between blocks and recombination events

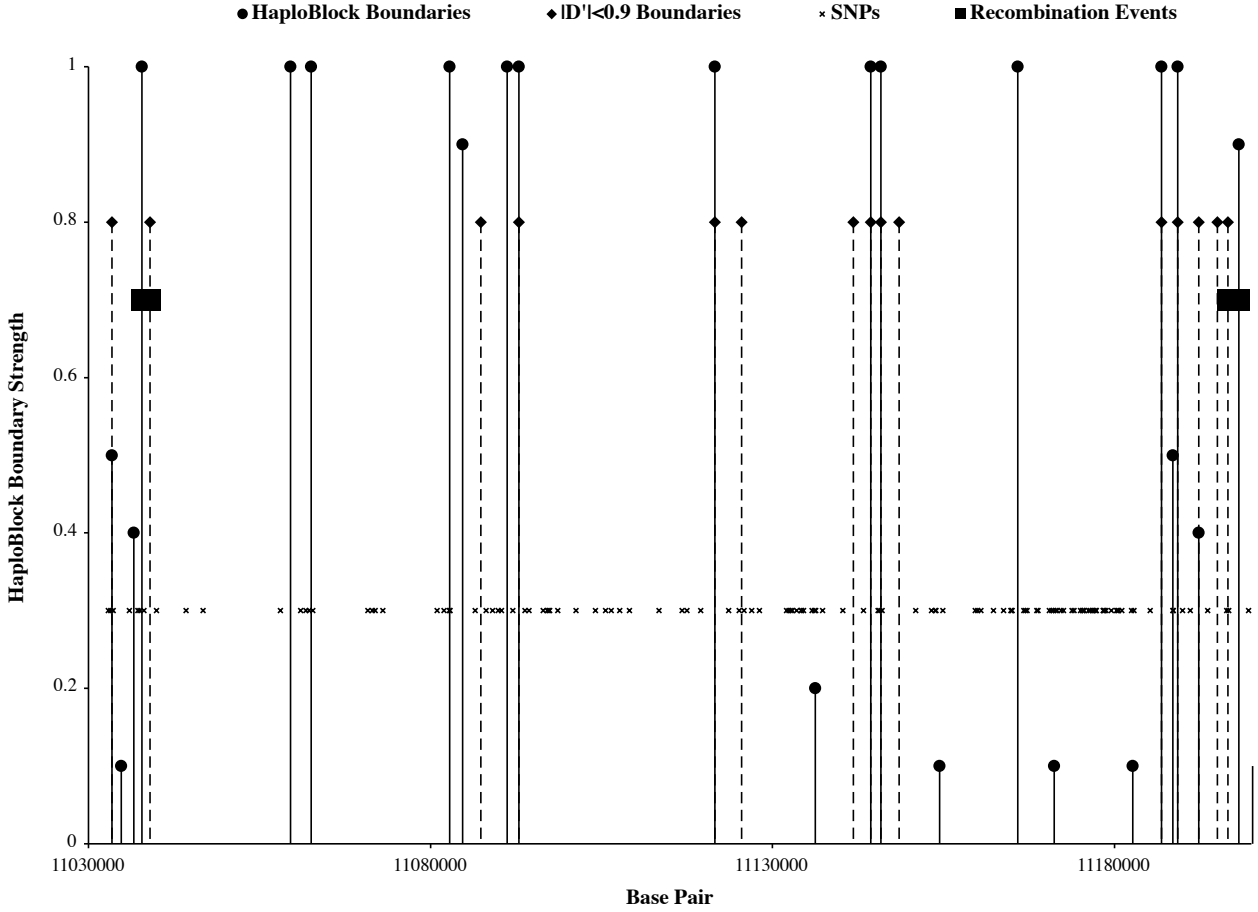


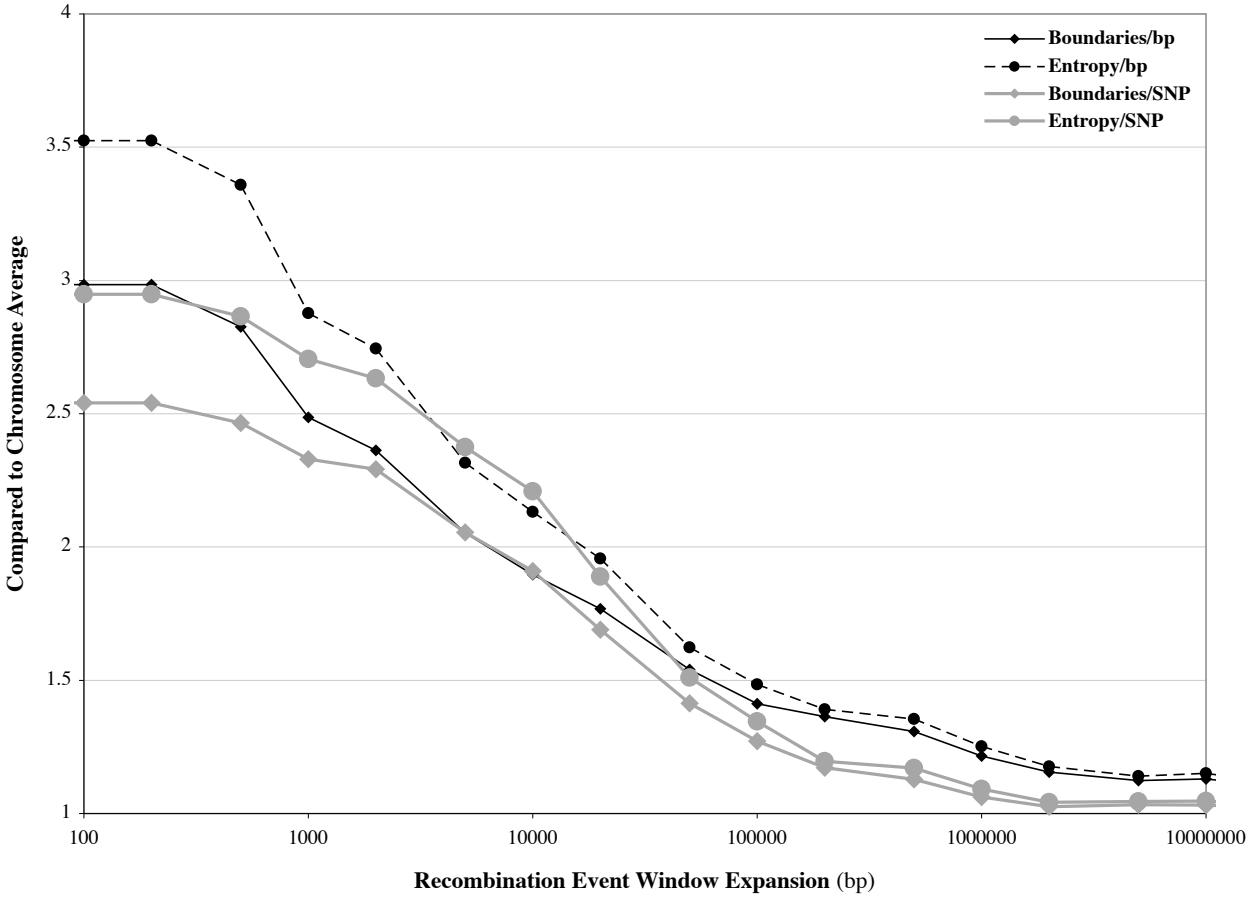

Table 7.4 shows the degree of correlation between the recombination event windows and the block boundaries inferred from the unphased Asian data. Suprisingly, the correlation between the CEPH event windows and the Asian boundaries is even stronger than that with the CEPH boundaries. This supports the claim that recombination hotspots are similar in different populations, in contrast to studies suggesting otherwise [71].

Finally, we assessed the degree to which the observed correlations are localized, i.e. due to short range rather than long range variation in recombination rates. Figure 7.7 shows the effect of artificially growing the recombination event windows symmetrically around their center, and then recalculating the four HaploBlock correlations with the CEPH boundaries accordingly. The graph shows that growing the windows by 20 kb removes over half of the observed increase in boundary density, relative to the chromosome average. This supports the theory that hotspots of increased recombination frequency are focused in a narrow area. Nonetheless, there is also some correlation

Table 7.4: Correlation between Asian block boundaries and CEPH event windows

| Method | Statistic | Chromosome | Event windows | Factor | P value |
|---|---|---|---|---|---|
| HaploBlock | Boundaries/kb $\times 10^{-2}$ | 4.15 | 12.15 | 2.9× | 0.00165 |
| | Boundaries/SNP $\times 10^{-2}$ | 7.80 | 19.79 | 2.5× | 0.00003 |
| | Entropy/kb $\times 10^{-2}$ | 2.21 | 8.13 | 3.7× | 0.00082 |
| | Entropy/SNP $\times 10^{-2}$ | 4.15 | 14.03 | 3.4× | 0.00000 |

Figure 7.7: Effect on statistics of artificially growing recombination event windows




with long-range variation in block boundary density, corresponding to the chromosomal variation in recombination levels shown earlier in Figure 7.5.

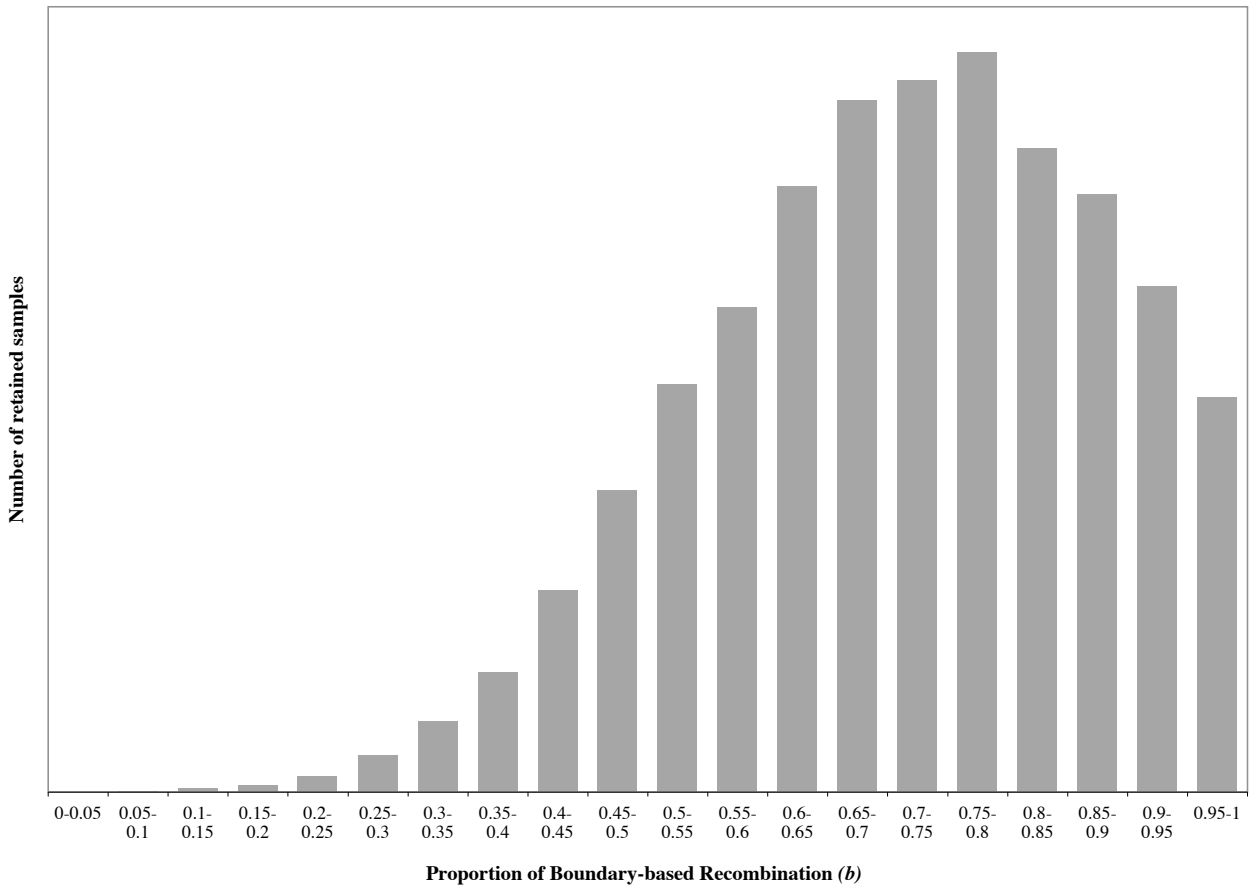
### 7.2.2 Proportion of Recombination on Boundaries

Our previous results show that the rate of recombination is higher near haplotype block boundaries than in the chromosome as a whole. We therefore attempted to quantify the proportion $0 \leq b \leq 1$ of recombination events which occurred on block boundaries, assuming the following generative model: (a) A proportion $b$ of events occur after SNP $j$ with distribution $S(j)$, where $S(j)$ is proportional to the number of models sampled which placed a boundary between SNPs $j$ and $j+1$. (b) A proportion $1-b$ of events occur after SNP $j$ with distribution $D(j)$, where $D(j)$ is proportional to the physical distance between SNPs $j$ and $j+1$, i.e. uniform recombination density.

We estimated the value of $b$ from the observed recombinations and CEPH block boundaries using three methods: an EM algorithm, trend analysis and rejection sampling. The EM algorithm searched for the value of $b$ which maximized the probability of the observed recombination event window locations. We initialized $b \leftarrow 0.5$, then calculated the posterior probability for each recombination event window that the recombination occurred on a haplotype block boundary, rather than at a uniformly random location. Averaging these posterior probabilities over all the event windows generated a new estimate for $b$, and this process was repeated until it converged on $b = 0.77$. We also did an additional run, weighting block boundaries by their conditional entropy. In this case the EM algorithm converged on a value of $b = 0.76$.

The second estimate of $b$ was by trend analysis, in which the relationship between $b$ and the boundary density statistics was examined by simulation. For values of $b$ ranging from 0 to 1, we

Figure 7.8: Histogram of $b$ values retained from rejection sampling




simulated a million random relocations of the recombination events. These locations were then extended based on homozygous stretches in the parent to simulate detectable event windows from which boundary/bp and boundary/SNP statistics were calculated. The observed boundaries/bp statistic matched the mean simulated value for $b = 0.44$, but it also fell in the 80% confidence interval for any value of $b$ between 0.21 and 0.91. The analogous boundaries/SNP point estimate was $b = 0.75$, with a range of 0.49 to 1.0. Clearly, neither of these bounds is narrow enough to estimate $b$ accurately. One problem is that only a small number of the event windows were short enough to give a strong signal as to whether the event fell on a block boundary or not. However, even if all 51 events could be unambiguously identified as one type or the other, we would still be estimating $b$ as the parameter of a binomial distribution with too few samples.

The third method of estimating $b$ was by rejection sampling. We repeated a process in which the value of $b$ was drawn uniformly between 0 and 1, after which recombination event windows were simulated as before. If both the boundaries/bp and boundaries/SNP statistics for these event windows matched to within 1% of the true statistics, the value of $b$ was retained, otherwise it was discarded. 25,000 samples of $b$ were produced in this way and then analyzed, with a mean of $b = 0.71$. A histogram representing the spread of values is shown in Figure 7.8. The median value obtained was $b = 0.72$ with an 80% confidence interval of 0.48 to 0.92.

In summary, there is not enough information in this data set to estimate $b$ with a high degree of confidence, and the true value could reasonably be anywhere between 0.4 and 0.9. Our best estimate is somewhere between 0.7 and 0.8, but this would have to be confirmed by more data.

## 7.3 Discussion

The goal of this analysis was to assess whether haplotype blocks arise due to recombination hotspots, or whether they can be explained purely in terms of genetic drift and random recombination. We addressed this question by examining whether the recombinations observed between the second and third generations of a pedigree were correlated with the haplotype block boundaries observed in the second generation. The results show a highly significant correlation, with P values of 0.002 and less, depending on the measure used. It has already been shown by Jeffreys and others that recombination hotspots exist in the human genome [48, 49]. Our results confirm that recombination hotspots also play a significant role in the genesis of haplotype blocks.

We also showed that the recombination events observed in the CEPH pedigrees are strongly correlated with the haplotype block boundaries inferred from the Asian population sample. This is important for two reasons. First, it removes the suspicion that the result derived from the CEPH data was artifactual, due in some way to the common data set underlying the block and recombination inferences. Second, it suggests that patterns of recombination and blocks are shared between different human populations, meaning that inferences made from one population are likely to be helpful when analyzing another.

We estimated in Section 7.2.2 that anywhere between 40% and 90% of recombinations take place on haplotype block boundaries, with the highest level of confidence in the range 70%–80%. Since we do not expect to observe all recombination hotspots as block boundaries, the proportion of recombination events taking place at hotspots could be even higher. A more accurate estimate would require a pedigree with higher SNP density and more recombinations observed.

# Chapter 8

# Markov Property

# Introduction

Our model for the haplotype block variation in a genomic region uses a Markov chain to approximate the joint distribution over all the blocks in a population. In the Markov model, the probability of a haplotype being descended from each block ancestor is conditional on its ancestor for the previous block. Markov models have been applied in many haplotype block models besides our own [2, 56, 16]. They are also commonly used in LD mapping algorithms for representing background genetic variation [80, 86, 88, 70].

A simpler approximation of the joint distribution over blocks is a model which assumes that all blocks are independent, i.e. that the probability of a haplotype is the product of the frequencies of each block variant within. This type of model is common, and constitutes an implicit assumption in many LD mapping studies which examine correlations between phenotypes and individual markers. The independent model has the advantage of requiring a small number of parameters, namely the frequencies for each block. However, this model breaks down when representing the variation over short distances, since markers which are close together tend to exhibit a high degree of linkage disequilibrium that cannot be captured by an independent approximation.

In this chapter, we analyze data taken from the International Haplotype Mapping (HapMap) project to compare the performance of the Markov and independent models. For any given joint distribution, a Markov approximation will clearly be more accurate than an independent approximation, since it has more parameters available for optimization. However we found an important additional property of the Markov approximation that we consider surprising – when used to model haplotype blocks, the Markov approximation is most accurate in the presence of high levels of linkage disequilibrium. Consequently, the Markov model is more accurate for blocks which are close together than those which are far apart. We also found that when modeling individual SNPs instead of haplotype blocks, this property of the Markov model is not exhibited. In other words, a Markov model over haplotype blocks provides a uniquely accurate way to represent background genomic distributions at high resolution.

Section 8.1 explains how we measure the accuracy of the Markov and independent approximations for haplotype blocks and individual SNPs. Section 8.2 performs these measurements on the HapMap data, showing how the accuracy of the approximations varies with physical distance and local recombination rates. Section 8.3 provides a theoretical explanation of the observed phenomena, with a full mathemetical proof in Section 8.4. Finally, Section 8.5 explores some of the implications of our findings, and discusses ways in which this work could be expanded in future.

This chapter has been submitted for publication [37].

## 8.1 Method

### 8.1.1 Independent and Markov Approximations

Consider a genomic region which contains $l$ markers, placed at physical locations $z_1 \dots z_l$ along the chromosome (measured in base pairs). Each marker $j = 1 \dots l$ has $r_j$ alleles, labelled $1 \dots r_j$. We consider a population in Hardy-Weinberg equilibrium, so the background variation for the region is given in terms of a joint distribution over haplotype frequencies [45]. Let $P(x_1, \dots, x_l)$ be the frequency of haplotype $x_1 \dots x_l$ in the population, where each $x_j$ takes the values $1 \dots r_j$.

Under the independent model, each marker is assumed to be independent. The maximum likelihood independent approximation $T(x_1, \dots, x_l)$ of the joint distribution $P$ is as follows:

$$\begin{aligned} T(x_1, \dots, x_l) &= \prod_{i=1}^{l} P(x_i) \\ \text{where } P(x_i) &= \sum_{x_1 \dots x_{i-1}, x_{i+1} \dots x_l} P(x_1, \dots, x_l) \end{aligned}$$

Under the Markov model, the distribution for each marker is dependent on the allele present at the preceding marker. The maximum likelihood Markov approximation $Q(x_1, \ldots, x_l)$ of the joint distribution is as follows:

$$\begin{aligned} Q(x_1, \ldots, x_l) &= P(x_1) \prod_{i=1}^{l-1} P(x_{i+1}|x_i) \\ \text{where } P(x_{i+1}|x_i) &= \frac{\sum_{x_1 \ldots x_{i-1}, x_{i+2} \ldots x_l} P(x_1, \ldots, x_l)}{\sum_{x_1 \ldots x_{i-1}, x_{i+1} \ldots x_l} P(x_1, \ldots, x_l)} \end{aligned}$$

### 8.1.2 Error Measures

Given a distance $d$ and a number $n \geq 3$ of markers, we generate statistics $Y_{d,n}$ and $Z_{d,n}$ to quantify the average error of the independent and Markov approximations respectively over a genomic region. We set a minimum of $n = 3$ since a Markov model can represent any joint distribution over 1 or 2 loci perfectly, rendering our measure meaningless.

The statistics $Y_{d,n}$ and $Z_{d,n}$ for a genomic region are generated by averaging the respective sets of statistics $Y_{d,n}(j)$ and $Z_{d,n}(j)$ over all valid start markers $j$ within that region. Each statistic $Y_{d,n}(j)$ or $Z_{d,n}(j)$ measures the error of the independent or Markov approximation over $n$ markers, where the first marker $j_1 = j$ and the other markers $j_2 \ldots j_n$ are chosen to be spread approximately evenly over total distance $d$. Each marker $j_i$ is selected to minimize $|z_{j_i} - z_{j_1} - d \cdot (i-1)/(n-1)|$. If any two of the marker indices $j_1 \ldots j_n$ are identical, we conclude that there is insufficient marker density for $n$, $d$ and $j$. In this case, $j$ is not a valid start marker and we omit $Y_{d,n}(j)$ and $Z_{d,n}(j)$ from their respective averages.

We set $Y_{d,n}(j) = ||P(x_{j_1}, \ldots, x_{j_n}) - T(x_{j_1}, \ldots, x_{j_n})||$, the variation distance between the observed joint distribution $P$ and the independent approximation $T$ for markers $j_1 \ldots j_n$. Similarly, we set $Z_{d,n}(j) = ||P(x_{j_1}, \ldots, x_{j_n}) - Q(x_{j_1}, \ldots, x_{j_n})||$, the variation distance between $P$ and the Markov approximation $Q$. The variation distance between two distributions is defined as follows:

$$||A(z_1, \ldots, z_n) - B(z_1, \ldots, z_n)|| = \frac{1}{2} \sum_{z_1 \ldots z_n} |A(z_1, \ldots, z_n) - B(z_1, \ldots, z_n)|$$

This measure is also known as the total variational distance, Kolmogorov distance, or $L_1$ distance. It has an intuitive definition as the total amount of probability mass that must be moved in order to make one distribution equal to the other. For example, $||P - T||$ is the percentage of the population distributed as $P$ which is misrepresented by the independent approximation $T$.

The variation distance between the joint distribution $P$ and its independent approximation $T$ is closely related to the $D$ measure of linkage disequilibrium for two biallelic markers. Consider two markers A and B, each with two alleles $a_1$, $a_2$, $b_1$ and $b_2$ at frequencies $p_1$, $p_2$, $q_1$ and $q_2$ respectively. Let $p_{11}$, $p_{12}$, $p_{21}$ and $p_{22}$ be the respective frequencies of the four gametes $a_1b_1$, $a_1b_2$, $a_2b_1$ and $a_2b_2$. The linkage disequilibrium measure $D$ is defined as $D = p_{11} - p_1q_1 = p_1q_2 - p_{12} = p_2q_1 - p_{21} = p_{22} - p_2q_2$ [18]. For example, if A and B are in perfect linkage equilibrium, then $p_{11} = p_1q_1$, $p_{12} = p_1q_2$, $p_{21} = p_2q_1$ and $p_{22} = p_2q_2$, and so $D = 0$. By comparison, the variation distance between $P$ and $T$ is:

$$\begin{aligned} ||P - T|| &= \frac{1}{2} \left( |p_{11} - p_1q_1| + |p_{12} - p_1q_2| + |p_{21} - p_2q_1| + |p_{22} - p_2q_2| \right) \\ &= \frac{1}{2} \left( |D| + |D| + |D| + |D| \right) = 2|D| \end{aligned}$$

Thus, for two biallelic markers, the variation distance between the joint distribution $P$ and its independent approximation $T$ is twice the absolute value of $D$.

### 8.1.3 HapMap Analysis

We used the October 2004 data release of the International Haplotype Mapping (HapMap) project to profile the error rates of the independent and Markov approximations for the human genome [44]. We inferred the transmitted and untransmitted haplotypes for all 22 autosomes of both parents in the 30 CEPH trios, so that 2640 chromosome haplotypes were examined in total. Haplotype alleles that could not be determined were left as unknown. This occurred at sites for which (a) a genotype was absent, (b) a Mendelian error was detected, or (c) all three members of the trio were heterozygous.

We examined the HapMap data using two approaches: (a) treating each SNP as an individual marker, and (b) grouping the SNPs into haplotype blocks. For the first approach, each SNP marker had $r_j = 2$ alleles, since all SNP data in the HapMap is biallelic. Trivially, $z_j$ was set to the physical location of each SNP.

For the second approach, we used the program HaploBlock to partition the marker data for each chromosome into $l$ blocks and find up to 4 common variants for each block [35]. These variants were considered as the block's alleles, so that $r_j \leq 4$ for all blocks $j$. The physical location $z_j$ of each block $j$ was set to the midpoint of the chromosomal section containing the SNPs within. To prevent a bias in favor of the Markov approximation, we removed the dependencies between adjacent ancestor variables in the HaploBlock statistical model [35].

Recall from Section 8.1.2 that we omit values $Y_{d,n}(j)$ and $Z_{d,n}(j)$ from the averages $Y_{d,n}$ and $Z_{d,n}$ if $n$ markers are not available with roughly equal spacing over distance $d$ starting at marker $j$. Furthermore, to prevent a bias arising from clustered unknowns, we did not calculate statistics $Y_{d,n}(j)$ and $Z_{d,n}(j)$ if less than half of the 120 haplotypes had known alleles for markers $j_1 \ldots j_n$. In the case of haplotype blocks, we also omitted individual haplotypes from a calculation if the required block alleles could not be assigned with at least 50% certainty under the HaploBlock statistical model [36].

## 8.2 Results

### 8.2.1 Distance Profiles

We assessed how the error rates of the independent and Markov approximations varied over different distances $d$. The distance profiles were generated by calculating average values of $Y_{d,n}$ and $Z_{d,n}$ over the entire autosome for values of $3 \leq n \leq 5$.

Figure 8.1 shows the error measures $Z_{d,n}$ for the Markov approximation for haplotype blocks over different distances $d$. Values are shown relative to $Z_{d,n}$ at the longest distance, where linkage disequilibrium is minimal. These baseline error measures $Z_{d,n}$ are 0.133, 0.307 and 0.512 for $n = 3, 4, 5$, respectively. To avoid a bias at short distances towards genomic regions with particularly high levels of variation, the graph in Figure 8.1 only shows the average for distances $d \geq 100$ kb for which at least 75% of the values $Z_{d,n}(j)$ could be generated.

The graph in Figure 8.1 highlights our core observation – that the Markov approximation performs best for haplotype blocks which are close together and between which there are high levels of linkage disequilibrium. For example, for $n = 4$ blocks spread over $d = 250$ kb, the Markov approximation shows a 10% improvement in average accuracy compared to 4 blocks spread over an entire chromosome. For $n = 5$ blocks, the improvement is over 15%. Figure 8.1 also shows that the relationship between distance and accuracy is not monotonic – at intermediate distances, the approximation performs worse than at both shorter and longer distances. This phenomenon can be seen most clearly for $n = 3$ blocks, where the average accuracy of the Markov approximation at $d = 250$ kb is equal to that at long distances, but is less accurate at distances between. These results are explained in Section 8.3 by reference to two contrasting processes of mixing and perturbation.

Figure 8.2 shows the corresponding error measures $Y_{d,n}$ for the independent approximation for haplotype blocks over different distances $d$. In contrast to Figure 8.1, this graph shows a monotonic

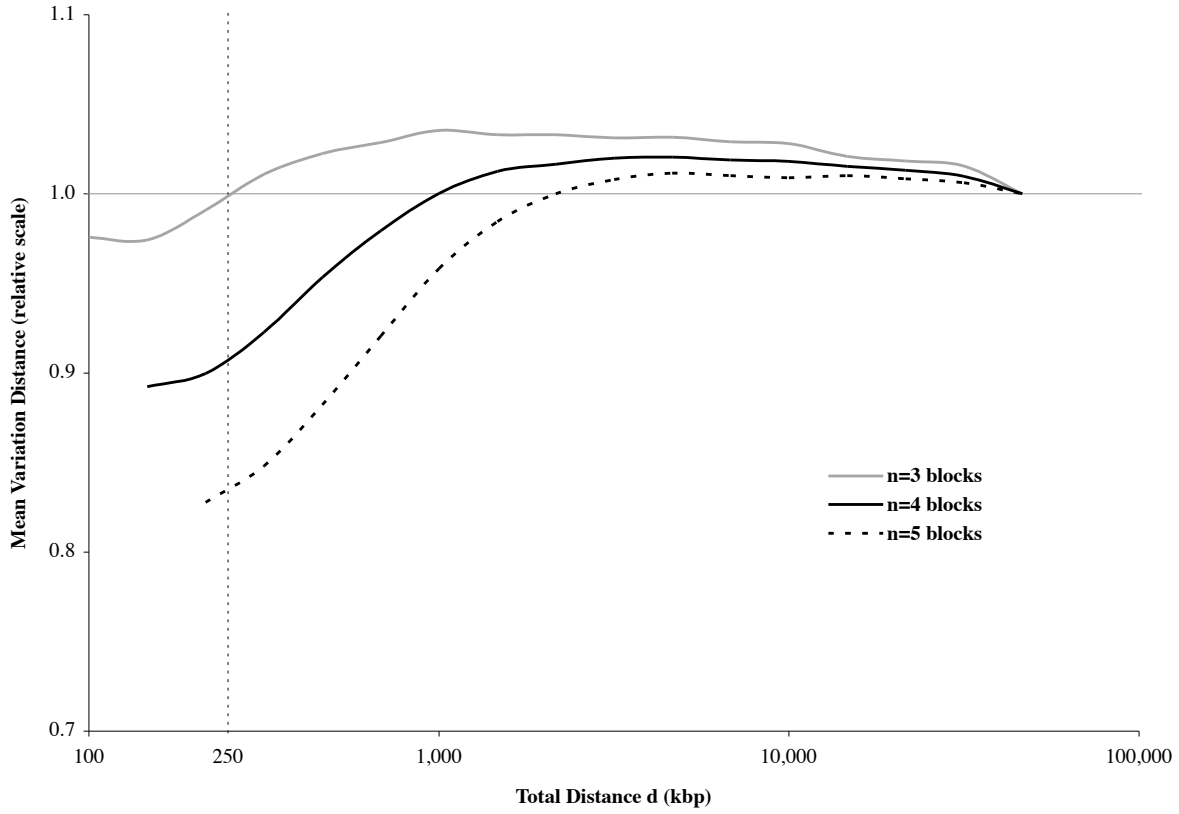



Figure 8.1: Distance profile of Markov approximation for haplotype blocks

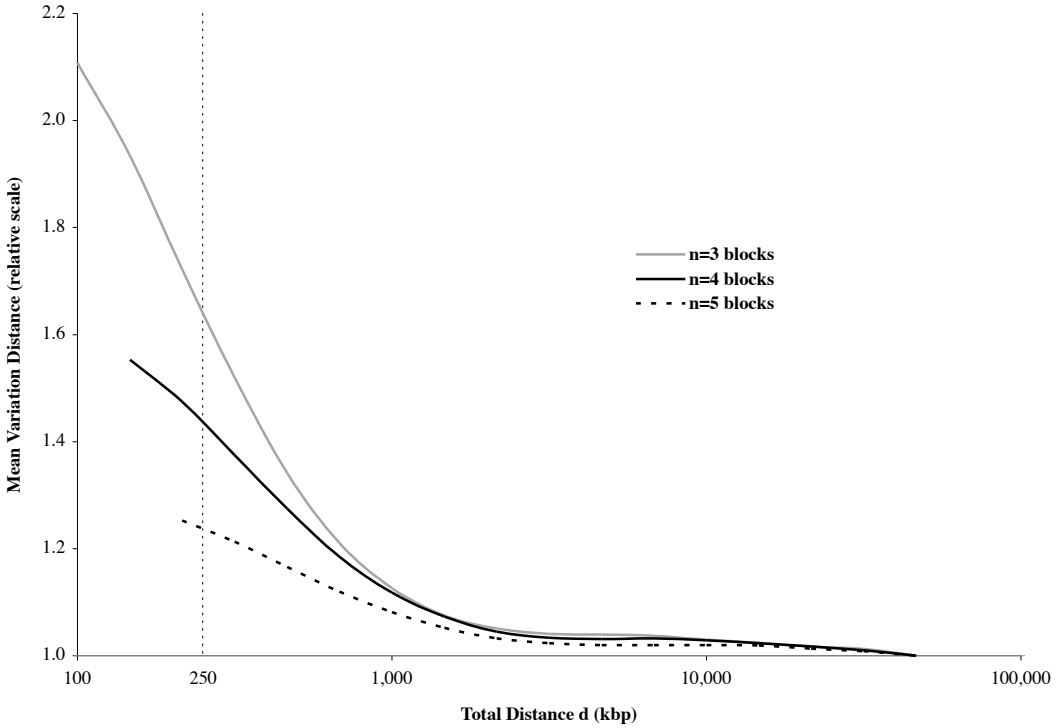



Figure 8.2: Distance profile of independent approximation for haplotype blocks

decrease in the independent approximation's error with physical distance. This reflects the fact that the accuracy of the independent approximation improves as the linkage disequilibrium between blocks decreases. One would naturally expect the Markov approximation to behave similarly, yet the results in Figure 8.1 show otherwise. The values in Figure 8.2 are shown relative to baseline error measures $Y_{d,n}$ of 0.192, 0.362 and 0.560 for $n = 3, 4, 5$, respectively. The baseline increases with the number $n$ of markers due to the increase in the cardinality of distribution $P(x)$, which represents $4^n$ different haplotypes for blocks with 4 alleles.

We generated similar profiles for the case where each SNP is treated as an individual marker with 2 alleles. Figure 8.3 compares the distance profiles obtained for individual SNPs against those for haplotype blocks, using $n = 4$ in all cases. This graph shows that, for modeling individual SNPs, both the independent and the Markov approximations perform best over longer distances, i.e. where there is less linkage disequilibrium between the markers modeled. In other words, the Markov model performs best at short distances only when used with haplotype blocks. As explained later in Section 8.3, this difference in behavior between blocks and SNPs stems from the difference in allele diversity.

The baseline error measures do not converge to zero at large genomic distances, as would be the case in the absence of linkage disequilibrium. The main reason for this is that our sample size is small – even if a pair of markers are in perfect linkage equilibrium in a population, a small sample from that population will contain some LD due to sampling error. A second possibility is that some long-range LD is present in the population, due for example to admixing or preferential mating.

### 8.2.2 Position Profiles

We now assess how the accuracy of the independent and Markov approximations varies along each chromosome in comparison with local recombination rates. Statistics $Y_{d,n}$ and $Z_{d,n}$ and average recombination rates were calculated for a sliding window of 20 Mb across each autosome. We

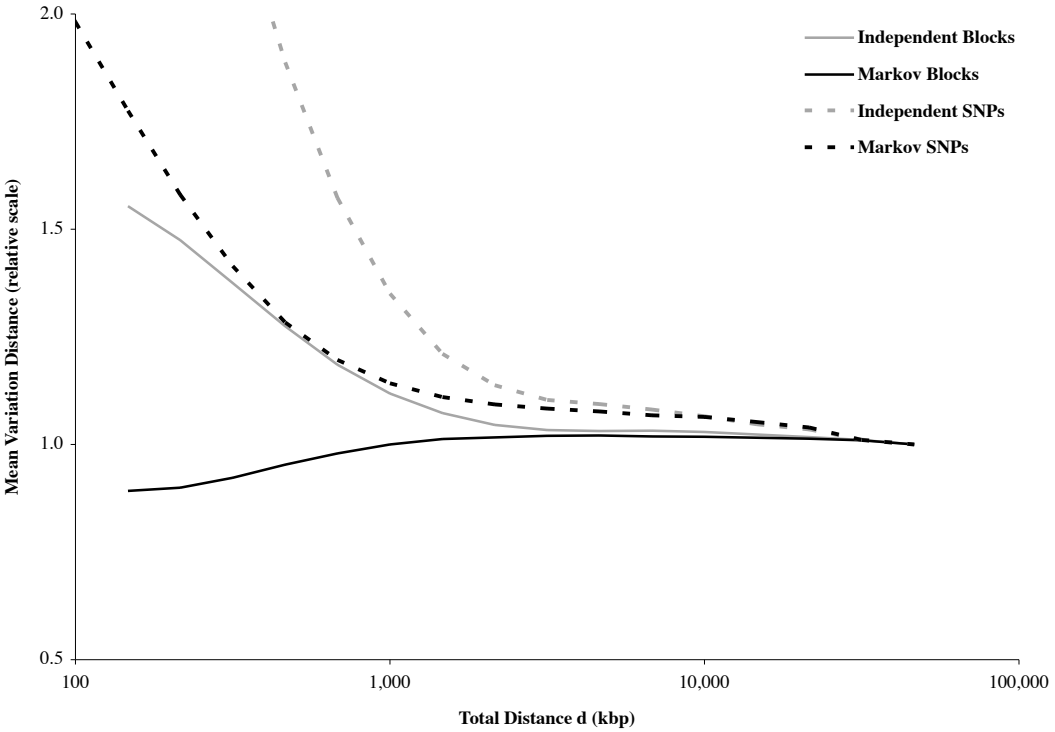



Figure 8.3: Comparison of all distance profiles

used fixed values of $d = 250$ kb and $n = 4$ for both haplotype blocks and individual SNPs. Local recombination rates were taken from the deCODE map and aligned against the genome build for our HapMap data using the UCSC Table Browser [61, 55].

Table 8.1 shows the correlation coefficients between the error measures and the recombination rates over the window positions for each chromosome. Windows with low SNP density due to their proximity to a centromere were excluded from these calculations. As can be seen in Table 8.1, only the Markov approximation for haplotype blocks shows a positive correlation between recombination rates and approximation error, with an average coefficient over the chromosomes of $0.535 \pm 0.271$. This contrasts with the independent approximation for haplotype blocks, with average correlation coefficient $-0.760 \pm 0.263$.

When considering SNPs individually, a different picture emerges. Both the independent and Markov approximations have lower error rates in regions of high recombination, just as the independent approximation for blocks. This confirms our observation from Section 8.2.1 that the Markov model's accuracy in the presence of high LD applies only when it is used with haplotype blocks.

The correlations coefficients for chromosomes 21 and 22 in Table 8.1 differ significantly from the mean values in many cases. This is because the HapMap data covers just 37 Mb of chromosome 21 and 35 Mb of chromosome 22, so that a sliding window of 20 Mb produces a weak signal. If these chromosomes are removed from the sample, the average coefficients for blocks under the Markov and independent models are $0.558 \pm 0.230$ and $-0.822 \pm 0.145$ respectively. The performance of the individual SNP models for chromosome 16 is also a strong outlier, for which the explanation is unclear.

It is instructive to look at one chromosome in more depth, to see how the error measures vary in comparison to local recombination rates. We examine here chromosome 11, since its correlation coefficients as shown in Table 8.1 are close to the averages over all of the chromosomes. A full set of profiles for all 22 autosomes is available online along with the haplotype block models inferred.

Table 8.1: Correlation between recombination rates and error measures

| | Haplotype blocks | | Individual SNPs | |
|---|---|---|---|---|
| Chromosome | Markov | Independent | Markov | Independent |
| 1 | 0.715 | -0.833 | -0.431 | -0.611 |
| 2 | 0.477 | -0.753 | -0.509 | -0.748 |
| 3 | 0.519 | -0.885 | -0.810 | -0.867 |
| 4 | 0.637 | -0.833 | -0.675 | -0.804 |
| 5 | 0.341 | -0.868 | -0.816 | -0.883 |
| 6 | 0.606 | -0.815 | -0.695 | -0.936 |
| 7 | 0.023 | -0.890 | -0.852 | -0.904 |
| 8 | 0.758 | -0.816 | -0.458 | -0.835 |
| 9 | -0.013 | -0.749 | -0.210 | -0.484 |
| 10 | 0.556 | -0.818 | -0.748 | -0.877 |
| 11 | 0.569 | -0.832 | -0.555 | -0.809 |
| 12 | 0.735 | -0.937 | -0.811 | -0.893 |
| 13 | 0.864 | -0.945 | -0.878 | -0.956 |
| 14 | 0.411 | -0.905 | -0.854 | -0.941 |
| 15 | 0.680 | -0.872 | -0.309 | -0.818 |
| 16 | 0.566 | -0.697 | 0.482 | 0.026 |
| 17 | 0.564 | -0.883 | -0.715 | -0.626 |
| 18 | 0.780 | -0.968 | -0.801 | -0.922 |
| 19 | 0.587 | -0.866 | -0.790 | -0.897 |
| 20 | 0.786 | -0.274 | -0.843 | -0.859 |
| 21 | 0.756 | -0.456 | 0.790 | -0.613 |
| 22 | -0.157 | 0.178 | -0.521 | 0.042 |
| Mean | 0.535 | -0.760 | -0.546 | -0.737 |
| S.D. | 0.271 | 0.263 | 0.429 | 0.279 |

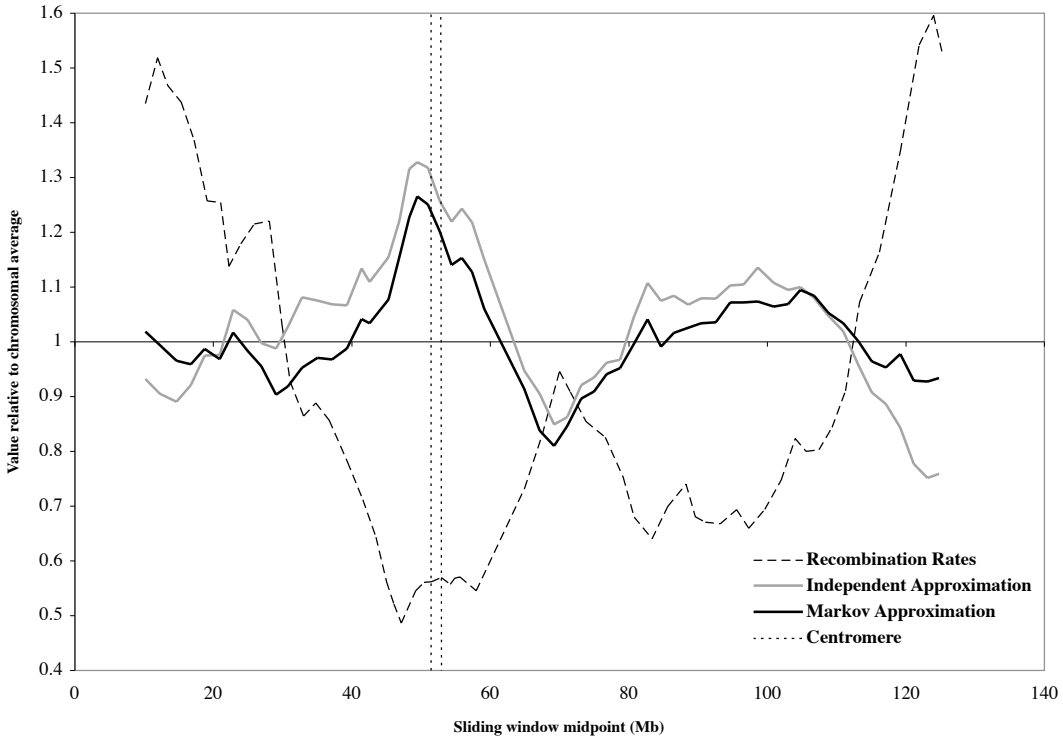



Figure 8.4: Position profiles for individual SNP models over chromosome 11

Figure 8.4 shows how the individual SNP approximation errors vary with recombination rates over the chromosome. As can be seen, the error rates of the two approximations follow each other closely, and are strongly anti-correlated with recombination rates. At the ends of the chromosome where recombination rates are highest, both approximations perform well. At the centromere, where recombination rates are generally lower, the opposite effect is seen. In particular, recombination rates near the centromere are about 50% of the average, while the Markov and independent approximation error is 20%-30% higher than the average.

Figure 8.5 shows how the haplotype block approximation errors vary over chromosome 11 with local recombination rates. The independent approximation performs best at the chromosome ends where recombination rates are highest, and worst near the centromere where they are low. The behavior is very similar to that presented in Figure 8.4 for individual SNPs. By contrast, the Markov approximation for blocks performs worst at the ends of the chromosome, and best near the centromere. Consequently, unlike the SNP approximations shown in Figure 8.4, the independent and Markov approximations for haplotype blocks are significantly out of phase.

## 8.3 Theory

### 8.3.1 Mixing and Perturbation

The process by which markers on a chromosome are mixed into linkage equilibrium by recombination is well understood [30]. The speed of this mixing process depends on two key factors: (a) mixing is faster between more distant markers due to the higher probability of recombination, (b) mixing is faster between markers with fewer alleles (e.g. SNPs) since each recombination is more likely to bring the marker distribution closer to equilibrium [136, 105, 3]. Since the independent approximation error stems from linkage disequilibrium, the speed of mixing also determines the

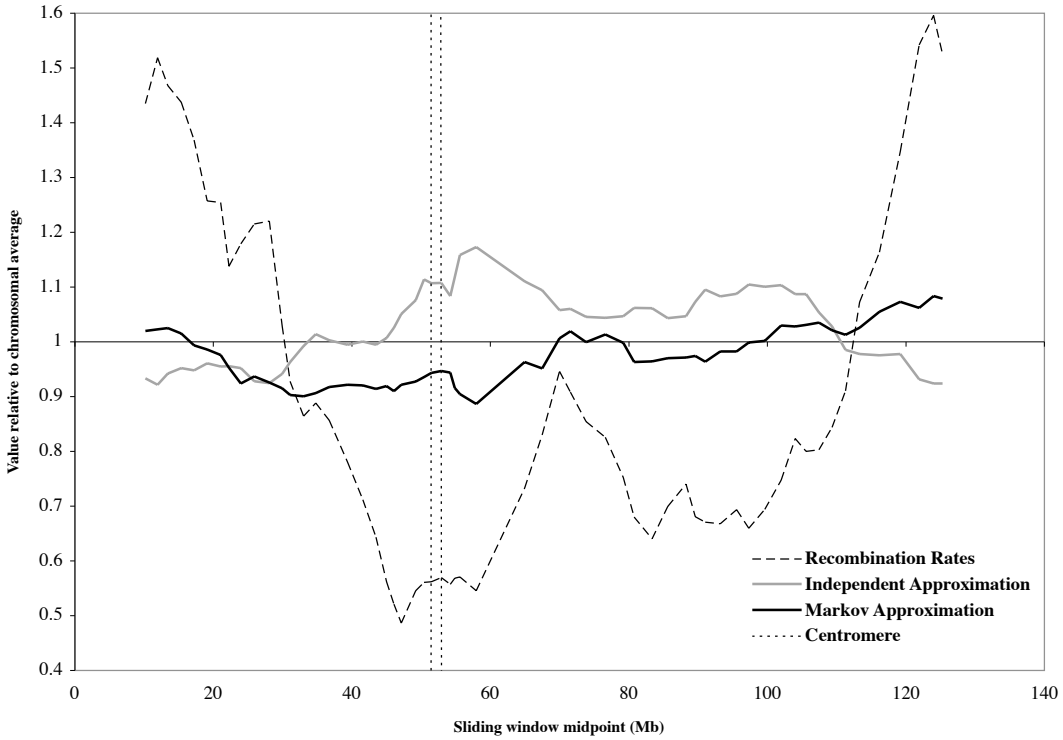



Figure 8.5: Position profiles for haplotype block models over chromosome 11

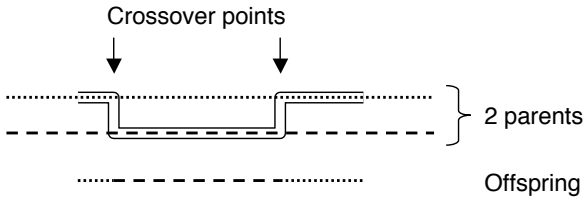



Figure 8.6: Meiotic recombination

accuracy of this approximation at different distances. An independent model is a special case of a Markov model, so the mixing process also contributes to the accuracy of the Markov approximation. To convert an independent model to a Markov model, one simply makes the conditional distribution for each marker in the Markov model identical for each allele of the previous marker. If a set of markers are in linkage equilibrium, they can be modeled with perfect accuracy by either an independent or a Markov approximation.

We introduce here a second process related to recombination which specifically affects the Markov approximation. This perturbation process refers to the long-range correlations generated by double recombinations which contribute to inaccuracy in the Markov model. Let us assume that two parent haplotypes are completely distinct from each other. The joint distribution over any set of markers in the parent haplotypes can be represented perfectly by a Markov model, since the allele at each variable site completely determines that at the next site. However, offspring haplotypes produced by double recombination from these parents receive two disjoint sections from one parent, separated by a section from the other parent, as shown in Figure 8.6. In these cases, the correlation between the disjoint sections cannot be expressed in terms of the intermediate region. Since the Markov model only represents dependencies between immediately adjacent markers, these double

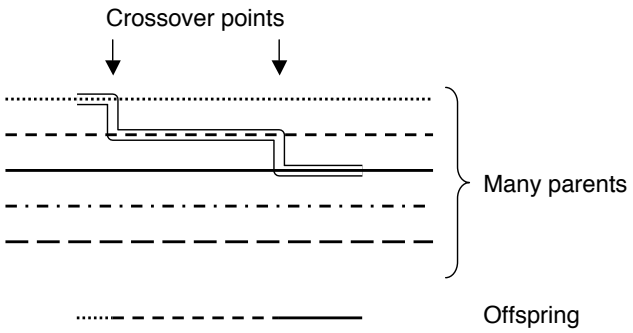



Figure 8.7: Intermixing

recombinations introduce inaccuracy in the Markov approximation for the offspring that was not present in the parents. As with the mixing process, this perturbation effect is strongest where the probability of recombination is higher, since this also means a higher probability of double recombinations.

The perturbation process constitutes a key difference between the dynamics of the independent and Markov models. In an infinite population, the accuracy of the independent approximation for a set of markers increases monotonically from one generation to the next. By contrast, the accuracy of the Markov approximation can increase or decrease, depending on the relative intensity of the mixing and perturbation processes. As we show later, the perturbation process is strongest for markers with a large number of alleles, rendering it is more visible for multi-allelic haplotype blocks than for biallelic SNP markers. This explains why we see a positive correlation between recombination rates and Markov approximation error for blocks, where perturbation is pronounced, but do not see this effect when modeling individual SNPs where perturbation is weaker.

The complex relationship shown in Figure 8.1 between physical distance and the accuracy of the Markov approximation for haplotype blocks is also explained by the balance between mixing and perturbation. At short distances, the Markov approximation over blocks is accurate due to the low probability of double recombination and the consequent lack of perturbation. At long distances, the Markov approximation over blocks is accurate due to the high probability of recombination and the consequent strong mixing. At intermediate distances, some perturbation takes place but mixing is weak, so the performance of the Markov approximation over haplotype blocks is at its worst.

### 8.3.2 Intermixing

For meiotic recombination under random mating, an offspring haplotype is generated from two parent haplotypes by the process depicted in Figure 8.6. Two parent haplotypes are selected independently from the source population. The offspring haplotype is generated from these parents by a reading process which crosses over from one parent to the other with probability $\theta_j$ between markers $j$ and $j+1$, where $\theta_j$ is the recombination fraction between the markers. As a result, the offspring haplotype can contain alternating stretches of genetic material from the two parents.

Our proof makes use of a different process called intermixing. Figure 8.7 depicts the intermixing process with the same crossover points as the meiosis in Figure 8.6. In intermixing, a large number of parent haplotypes are selected independently from the source population. The offspring haplotype is generated from these parents by a reading process which moves to a *new* parent with probability $\theta_j$ between markers $j$ and $j+1$. An offspring haplotype generated by intermixing with $x$ crossovers will contain genetic material from $x+1$ independently selected parents. In contrast to normal meiosis, the theoretical intermixing process cannot introduce new long-range dependencies, since the reading process never returns to a parent previously used.

The key point for our purposes is that if the first two intermixing parents are the same as

those for meiosis, the results of meiosis and intermixing are identical if no more than one crossover took place. With less than two crossovers, intermixing only uses the first two parent haplotypes, producing the same offspring haplotype as meiosis. Differences only arise due to double crossovers, after which meiosis returns to the first parent haplotype whereas intermixing selects a new parent. The proof that follows is based on this similarity between the two processes and the fact that intermixing preserves the Markov properties of a population regardless of how many crossovers take place.

### 8.3.3 Theorem

Consider a population of infinite size in Hardy-Weinberg equilibrium. This population undergoes random mating and meiotic recombination without interference in a series of discrete generations. Consider a set of $n$ markers numbered $1 \ldots n$, with recombination fraction $\theta_j$ between each pair of adjacent markers $j$ and $j+1$.

Define $P_u(x_1, \ldots, x_n)$ as the haplotype distribution over sites $1 \ldots n$ in generation $u$ and distribution $Q_u(x_1, \ldots, x_n) = P_u(x_1) \prod_{i=1}^{n-1} P_u(x_{i+1}|x_i)$ as its Markov approximation. Similarly, $P_{u+1}$ is the haplotype distribution that emerges in generation $u+1$ and $Q_{u+1}$ is its Markov approximation.

We define $Z_u = ||P_u - Q_u||$ as the variation distance between distributions $P_u$ and $Q_u$, and $Z_{u+1} = ||P_{u+1} - Q_{u+1}||$. Let $D_u(j) = 1 - \sum_{x_j} (P_u(x_j))^2$ be the heterozygosity of site $j$ in generation $u$, defined by the probability that two haplotypes chosen randomly from distribution $P_u$ differ at site $j$. Our theorem states that for $n \leq 5$:

$$Z_{u+1} \quad \leq \quad Z_u + \frac{1}{2} \left( \sum_{i=1}^{n-1} \theta_i \right)^2 \cdot \min \left( 1, \sum_{j=3}^{n} D_u(j) \right) \tag{8.1}$$

Thus, the error $Z_{u+1}$ of the Markov approximation in generation $u+1$ is bounded by the error $Z_u$ in generation $u$, plus an additional term which depends on two factors. The first factor is the square of the total of the intermarker recombination fractions. The second factor is the sum of the heterozygosities of sites $3 \ldots n$, bounded to be no more than 1.

A full proof of Equation 8.1 for $n \leq 5$ is provided in the Section 8.4. The outline is as follows. Let $P'_{u+1}$ be the distribution that emerges from performing intermixing on generation $u$ and $Q'_{u+1}$ be its Markov approximation. We use $P'_{u+1}$ and $Q'_{u+1}$ to prove the bound on $Z_{u+1} = ||P_{u+1} - Q_{u+1}||$ by applying the triangular inequality:

$$||P_{u+1} - Q_{u+1}|| \quad \leq \quad \left|\left|P_{u+1} - P'_{u+1}\right|\right| + \left|\left|P'_{u+1} - Q'_{u+1}\right|\right| + \left|\left|Q'_{u+1} - Q_{u+1}\right|\right|$$

The first step is to prove an upper bound on $||P_{u+1} - P'_{u+1}||$, the variation distance between the haplotype distributions generated by meiosis and intermixing. This distance is bounded by $\frac{1}{2} \left( \sum_{i=1}^{n-1} \theta_i \right)^2 \cdot \min(1, \sum_{j=3}^{n} D_u(j))$. The intuition here is that the results of meiosis and intermixing differ only if there was a double recombination, the probability of which is bounded by $\left( \sum_{i=1}^{n-1} \theta_i \right)^2$. If a double recombination did occur, the probability that the offspring haplotype will differ between meiosis and intermixing is bounded by the sum of the heterozygosities $D_u(j)$ for sites $j = 3 \ldots n$, since $j = 3$ is the first site that can be affected by a double recombination. A proof of the bound for all $n$ is provided in Section 8.4.2.

The second step is to bound $||P'_{u+1} - Q'_{u+1}||$, the variation distance between the distribution resulting from intermixing and its Markov approximation. We prove that for $n \leq 5$, this distance is no greater than $||P_u - Q_u|| = Z_u$. This result arises because each crossover event in intermixing selects a new parent haplotype at random, so no new long-range dependencies are introduced. A proof of this bound for $n \leq 5$ is provided in Section 8.4.3. We also conjecture that this bound holds true for all values of $n$, as suggested by extensive simulation.

The final step is to prove that $||Q'_{u+1} - Q_{u+1}|| = 0$, namely that the Markov approximations of the distributions arising from meiosis and intermixing are identical. The intuition here is that the Markov approximation is entirely determined by the joint distribution over each pair of adjacent sites, and this joint distribution is identical for both intermixing and meiosis. A proof of this result for all $n$ is provided in Section 8.4.4.

These results are combined under the triangular inequality to yield Equation 8.1:

$$\begin{aligned} ||P_{u+1} - Q_{u+1}|| &\leq \left|\left|P_{u+1} - P'_{u+1}\right|\right| + \left|\left|P'_{u+1} - Q'_{u+1}\right|\right| + \left|\left|Q'_{u+1} - Q_{u+1}\right|\right| \\ &\leq \frac{1}{2}\left(\sum_{i=1}^{n-1}\theta_i\right)^2 \cdot \min\left(1, \sum_{j=3}^{n} D_u(j)\right) + Z_u \end{aligned}$$

The average heterozygosity for individual SNPs in the HapMap data is $0.267 \pm 0.182$. By contrast, the average heterozygosity of blocks in our inferred models is $0.596 \pm 0.124$, more than double that for SNPs. This explains why the perturbation process is significantly stronger for haplotype blocks than for individual SNPs.

## 8.4 Proof

Here we prove in full the steps outlined in Section 8.3.3.

### 8.4.1 Definitions

Under meiotic recombination, each offspring haplotype over $n$ sites is formed from two parent haplotypes $y^1 = (y^1_1 \ldots y^1_n)$ and $y^2 = (y^2_1 \ldots y^2_n)$. Each meiosis entails a crossover vector $r = (r_1 \ldots r_n) \in \{0,1\}^{n-1}$, in which $r_i = 1$ if a crossover took place between sites $i$ and $i+1$ and $r_i = 0$ otherwise. Let $F(y^1, y^2, r)$ denote the offspring haplotype that is generated by meiosis from $y^1$ and $y^2$ assuming a crossover vector $r$:

$$F(y^1, y^2, r) = y_1^{S(r,1)} \ldots y_n^{S(r,n)} \tag{8.2}$$

In Equation 8.2, $S(r,i)$ is the index of the parent of site $i$ in the offspring, namely $S(r,i) = 1 + \sum_{k=1}^{i-1} r_k$ modulo 2. If there is an even number of recombinations up to site $i$ then $S(r,i)$ is 1, otherwise $S(r,i)$ is 2. Since both parents are selected randomly from the same distribution, we assumed without loss of generality that the first site in the offspring comes from parent $y^1$.

The probability of a crossover occurring between sites $i$ and $i+1$ is denoted by $\theta_i$. We define the probability $G(r)$ of a crossover vector $r$ in terms of these pairwise probabilities:

$$G(r_1, \ldots, r_{n-1}) = \prod_{i=1}^{n-1} \theta_i^{r_i} \cdot (1-\theta_i)^{1-r_i} \tag{8.3}$$

Recall that $P_u(x)$ denotes the frequency of haplotype $x$ in generation $u$. The frequency $P_{u+1}(x)$ of haplotype $x$ in generation $u+1$ due to meiotic recombination is the sum of the probabilities of all joint assignments to $y^1$, $y^2$ and $r$ which yield $x$:

$$P_{u+1}(x) = \sum_{y^1, y^2, r | F(y^1, y^2, r) = x} G(r) P_u(y^1) P_u(y^2) \tag{8.4}$$

For intermixing over $n$ sites, each offspring haplotype can inherit sections from up to $n$ haplotypes in the previous generation, although in most cases less than $n$ will be used. Let $F'(y^1, \ldots, y^n, r)$ denote the haplotype generated from $y^1, \ldots, y^n$ by intermixing under a crossover vector $r$:

$$F'(y^1, y^2, r) \quad = \quad y_1^{S'(r,1)} \ldots y_n^{S'(r,n)} \tag{8.5}$$

In Equation 8.5, $S'(r,i)$ is the index of the parent of site $i$ in the offspring, namely $S'(r,i) = 1 + \sum_{k=1}^{i-1} r_k$. The function $S'(r,i)$ counts the number of crossovers that have taken place up to site $i$. The frequency $P'_{u+1}(x)$ of haplotype $x$ in generation $u+1$ due to intermixing on parent distribution $P_u$ is as follows:

$$P'_{u+1}(x) \quad = \quad \sum_{y^1 \ldots y^n, r | F'(y^1, \ldots, y^n, r) = x} G(r) \prod_{i=1}^{n} P_u(y^i) \tag{8.6}$$

### 8.4.2 Intermixing and Meiosis

In this section, we prove the following bound on the variation distance between the haplotype distribution $P_{u+1}$ arising from meiosis on generation $u$, and the distribution $P'_{u+1}$ arising from intermixing:

$$\left|\left| P_{u+1} - P'_{u+1} \right|\right| \quad \leq \quad \frac{1}{2} \left( \sum_{i=1}^{n-1} \theta_i \right)^2 \cdot \min \left( 1, \sum_{j=3}^{n} D_u(j) \right) \tag{8.7}$$

Recall that $D_u(j)$ is defined as the heterozygosity of site $j$ in generation $u$, where $D_u(j) = 1 - \sum_{x_j} (P_u(x_j))^2$ is the probability that two haplotypes randomly chosen from $P_u$ differ at site $j$.

Let $R = \{0,1\}^{n-1}$ denote the set of all possible crossover vectors $r$. Let $R^-$ be the subset $\{r \in R | \sum_j r_j \leq 1\}$ consisting of crossover vectors representing one or less crossovers, and let $R^+ = \{r \in R | \sum_j r_j \geq 2\}$ denote the subset representing two or more crossovers. Clearly, $R = R^- \cup R^+$ and $R^- \cap R^+ = \emptyset$. The frequency of haplotype $x$ after meiosis, given in Equation 8.4, can therefore be written as:

$$\begin{aligned} P_{u+1}(x) \quad &= \quad \sum_{y^1, y^2, r \in R^- | F(y^1, y^2, r) = x} G(r) P_u(y^1) P_u(y^2) \\ &+ \quad \sum_{y^1, y^2, r \in R^+ | F(y^1, y^2, r) = x} G(r) P_u(y^1) P_u(y^2) \end{aligned} \tag{8.8}$$

Similarly, the frequency of $x$ after intermixing, given in Equation 8.6, can be written as:

$$\begin{aligned} P'_{u+1}(x) \quad &= \quad \sum_{y^1 \ldots y^n, r \in R^- | F'(y^1 \ldots y^n, r) = x} G(r) \prod_{i=1}^{n} P_u(y^i) \\ &+ \quad \sum_{y^1 \ldots y^n, r \in R^+ | F'(y^1 \ldots y^n, r) = x} G(r) \prod_{i=1}^{n} P_u(y^i) \end{aligned} \tag{8.9}$$

Recall that if $r \in R^-$ then $\sum_j r_j \leq 1$. In these cases, $S(r,i) = S'(r,i)$ for all $i$, yielding $F'(y^1, \ldots, y^n, r) = F(y^1, y^2, r)$. In other words, when less than two crossovers occur, the haplotype obtained by meiosis is identical to that obtained by intermixing for the same parents $y^1$ and $y^2$. Consequently, we rewrite Equation 8.9 as follows:

$$P'_{u+1}(x) = \sum_{y^1,y^2,r\in R^-|F(y^1,y^2,r)=x} G(r)P_u(y^1)P_u(y^2) \quad (8.10)$$
$$+ \sum_{y^1\ldots y^n,r\in R^+|F'(y^1\ldots y^n,r)=x} G(r)\prod_{i=1}^{n} P_u(y^i)$$

Since the sums in Equations 8.8 and 8.10 corresponding to no more than one crossover are identical, the variation distance between $P_{u+1}$ and $P'_{u+1}$ is due to two or more crossovers:

$$\left|\left|P_{u+1} - P'_{u+1}\right|\right| = \frac{1}{2}\sum_x \left| \begin{array}{l} \sum_{y^1,y^2,r\in R^+|F(y^1,y^2,r)=x} G(r)P_u(y^1)P_u(y^2) \\ -\sum_{y^1\ldots y^n,r\in R^+|F'(y^1,\ldots,y^n,r)=x} G(r)\prod_{i=1}^{n} P_u(y^i) \end{array} \right| \quad (8.11)$$

By introducing the unity sum $\sum_{y^3\ldots y^n}\prod_{i=3}^{n} P_u(y^i) = 1$ into the first term of Equation 8.11, we obtain:

$$\left|\left|P_{u+1} - P'_{u+1}\right|\right| = \frac{1}{2}\sum_x \left| \begin{array}{l} \sum_{y^1\ldots y^n,r\in R^+|F(y^1,y^2,r)=x} G(r)\prod_{i=1}^{n} P_u(y^i) \\ -\sum_{y^1\ldots y^n,r\in R^+|F'(y^1,\ldots,y^n,r)=x} G(r)\prod_{i=1}^{n} P_u(y^i) \end{array} \right| \quad (8.12)$$

We now derive the bound for $||P_{u+1} - P'_{u+1}||$, as given by Equation 8.7. Let $[a = b]$ denote the function that returns 1 if $a = b$ and 0 otherwise, and define $[a \neq b] = 1 - [a = b]$. Equation 8.12 is reformulated as follows:

$$\begin{aligned} \left|\left|P_{u+1} - P'_{u+1}\right|\right| &= \frac{1}{2}\sum_x \left| \begin{array}{l} \sum_{y^1\ldots y^n,r\in R^+} G(r)\prod_{i=1}^{n} P_u(y^i) \\ \cdot\left\{[F(y^1,y^2,r) = x] - [F'(y^1,\ldots,y^n,r) = x]\right\} \end{array} \right| \\ &\leq \sum_{r\in R^+} G(r) \sum_{y^1\ldots y^n}\prod_{i=1}^{n} P_u(y^i) \cdot \frac{1}{2}\sum_x \left|[F(y^1,y^2,r) = x] - [F'(y^1,\ldots,y^n,r) = x]\right| \\ &= \sum_{r\in R^+} G(r) \sum_{y^1\ldots y^n}\prod_{i=1}^{n} P_u(y^i) \cdot [F(y^1,y^2,r) \neq F'(y^1,\ldots,y^n,r)] \quad (8.13) \end{aligned}$$

The last equality follows because if $F(y^1,y^2,r) = F'(y^1,\ldots,y^n,r)$ then the variation distance $\left|[F(y^1,y^2,r) = x] - [F'(y^1,\ldots,y^n,r) = x]\right|$ is equal to 0 for all $x$, and if $F(y^1,y^2,r) \neq F'(y^1,\ldots,y^n,r)$, then $\left|[F(y^1,y^2,r) = x] - [F'(y^1,\ldots,y^n,r) = x]\right|$ is equal to 1 for exactly two values of $x$, namely $x = F(y^1,y^2,r)$ and $x = F'(y^1,\ldots,y^n,r)$.

The value $[F(y^1,y^2,r) \neq F'(y^1,\ldots,y^n,r)]$ is 1 if the haplotype that arises from meiosis is different from that arising from intermixing. This condition is fulfilled if the haplotypes differ in at least one site. The haplotypes are always identical at sites 1 and 2 since the earliest an observed double recombination can occur is between sites 2 and 3. In other words, $S(r,1) = S'(r,1)$ and $S(r,2) = S'(r,2)$ for any crossover vector $r$. By summing the possibilities for the remaining sites $3\ldots n$, we obtain a simple bound:

$$[F(y^1,y^2,r) \neq F'(y^1,\ldots,y^n,r)] \leq \sum_{j=3}^{n}[y_j^{S(r,j)} \neq y_j^{S'(r,j)}] \quad (8.14)$$

Equations 8.13 and 8.14 yield:

$$\left|\left|P_{u+1} - P'_{u+1}\right|\right| \leq \sum_{r\in R^+} G(r)\sum_{j=3}^{n}\sum_{y^1\ldots y^n}\prod_{i=1}^{n} P_u(y^i)\cdot[y_j^{S(r,j)} \neq y_j^{S'(r,j)}] \quad (8.15)$$

Since, in the worst case, every site from the third onwards has a different source under meiosis and intermixing, $\sum_{y^1 \ldots y^n} \prod_{i=1}^{n} P_u(y^i) \cdot [y_j^{S(r,j)} \neq y_j^{S'(r,j)}]$ is the probability that two independently selected haplotypes from distribution $P_u$ differ at site $j$. This is precisely the definition of heterozygosity $D_u(j)$, so:

$$\left|\left|P_{u+1} - P'_{u+1}\right|\right| \quad \leq \quad \sum_{r \in R^+} G(r) \sum_{j=3}^{n} D_u(j) \tag{8.16}$$

Since $[F(y^1, y^2, r) \neq F'(y^1, \ldots, y^n, r)] \leq 1$ by definition, an additional bound is obtained for $||P_{u+1} - P'_{u+1}||$ from Equation 8.13:

$$\left|\left|P_{u+1} - P'_{u+1}\right|\right| \quad \leq \quad \sum_{r \in R^+} G(r) \sum_{y^1 \ldots y^n} \prod_{i=1}^{n} P_u(y^i) = \sum_{r \in R^+} G(r) \tag{8.17}$$

Finally, using the probability $G(r)$ of a crossover vector $r$ (Equation 8.3), we bound $\sum_{r \in R^+} G(r)$ by summing the probability of every possible pair of crossovers:

$$\sum_{r \in R^+} G(r) \quad \leq \quad \sum_{i=1}^{n-1} \theta_i \sum_{k=i+1}^{n-1} \theta_k \leq \frac{1}{2} \left( \sum_{i=1}^{n-1} \theta_i \right)^2 \tag{8.18}$$

Equations 8.16, 8.17 and 8.18 yield the bound for $||P_{u+1} - P'_{u+1}||$, given by Equation 8.7.

### 8.4.3 Markov Accuracy after Intermixing

Recall that $P'_{u+1}(x)$ is the haplotype distribution that results from intermixing parent haplotype distribution $P_u$ and that $Q'_{u+1}(x)$ is the Markov approximation of $P'_{u+1}(x)$. In this section we prove that for $n \leq 5$:

$$\left|\left|P'_{u+1} - Q'_{u+1}\right|\right| \leq ||P_u - Q_u|| \tag{8.19}$$

For haplotypes with $n > 5$ sites, this problem remains open. However, we conjecture that it is true for all values of $n$, as confirmed by extensive simulation studies up to $n = 16$.

The formula for $P'_{u+1}(x)$ in Equation 8.6 is now rewritten in terms of contiguous sections inherited from a parent, using the probability $G(r)$ of each crossover vector $r$ and the probability of the parent haplotype sections that lead to $x$ under $r$:

$$\begin{aligned} P'_{u+1}(x) &= \sum_{r \in R} G(r) \prod_{k=1}^{S'(r,n)} P_u(x_{(r,k)}) \\ \text{where } x_{(r,k)} &= x_{L(r,k)} \ldots x_{U(r,k)} \\ L(r,k) &= \min\left\{i | S'(r,i) = k\right\} \\ U(r,k) &= \max\left\{i | S'(r,i) = k\right\} \end{aligned} \tag{8.20}$$

In Equation 8.20, the functions $L(r,k)$ and $U(r,k)$ denote respectively the first and last sites in the offspring haplotype which originate from parent $S'(r,i) = k$ under crossover vector $r$. Recall that $S'(r,i)$ is the index of the parent haplotype for site $i$ of the offspring haplotype when intermixing with crossover vector $r$. The term $P_u(x_{(r,k)})$ denotes the marginal distribution $P_u(x_{L(r,k)}, \ldots, x_{U(r,k)}) = \sum_{x_1 \ldots x_{L(r,k)-1}, x_{U(r,k)+1} \ldots x_l} P_u(x_1, \ldots, x_l)$.

The process of intermixing can be viewed as the transformation of a parent haplotype distribution $P_u$ into an offspring distribution $P'_{u+1}$. This transformation can be decomposed into a series of atomic transformations, one over each possible crossover point. Let $P'^{i}_{u+1}$ be the haplotype distribution obtained from intermixing if crossovers are only allowed over sites 1 to $i$. In other words, $P'^{i}_{u+1}$ is the result of intermixing on $P_u$ if all values $\theta_i \dots \theta_{n-1}$ are set to zero. Clearly, the distribution $P'^{1}_{u+1}$ equals the parent haplotype distribution $P_u$, since $P'^{1}_{u+1}$ is the result of intermixing if no crossing over is allowed. Similarly, the distribution $P'^{n}_{u+1}$ equals the distribution $P'_{u+1}$ that emerges from intermixing over all sites, since the full set of crossovers between sites 1 and $n$ are allowed. As a result, the transformation $P_u \rightarrow P'_{u+1}$ can be expressed as a series of transformations $P'^{1}_{u+1} \rightarrow P'^{2}_{u+1} \rightarrow \dots \rightarrow P'^{n}_{u+1}$, where each step $P'^{i}_{u+1} \rightarrow P'^{i+1}_{u+1}$ in the series introduces an additional crossover point between sites $i$ and $i+1$.

Let $R^i$ be the set of crossover vectors in which crossovers only occur between sites 1 to $i$, i.e. $R^i = \{r \in R | r_i = 0 \dots r_{n-1} = 0\}$. Let $G^i(r)$ be the probability of crossover vector $r \in R^i$, defined as follows:

$$G^i(r_1, \dots, r_{n-1}) = \prod_{j=1}^{i-1} \theta_j^{r_j} \cdot (1-\theta_j)^{1-r_j}$$

Using these definitions, the probability $P'^{i}_{u+1}(x)$ of haplotype $x$ after intermixing over sites $1 \dots i$ is analogous to $P'_{u+1}(x)$, given in Equation 8.20:

$$\begin{aligned} P'^{i}_{u+1}(x_1, \dots, x_n) &= \sum_{r \in R^i} G^i(r) \prod_{k=1}^{S'(r,n)} P_u(x_{(r,k)}) \\ &= \sum_{r \in R^i} G^i(r) \left( \prod_{k=1}^{S'(r,n)-1} P_u(x_{(r,k)}) \right) P_u(x_{L(r,S'(r,n))}, \dots, x_n) \end{aligned} \tag{8.21}$$

The recurrence relation between $P'^{i+1}_{u+1}$ and $P'^{i}_{u+1}$ is explicated by splitting $P'^{i+1}_{u+1}(x)$ into two:

$$\begin{aligned} P'^{i+1}_{u+1}(x) &= \sum_{r \in R^{i+1} | r_i = 0} G^{i+1}(r) \prod_{k=1}^{S'(r,n)} P_u(x_{(r,k)}) + \sum_{r \in R^{i+1} | r_i = 1} G^{i+1}(r) \prod_{k=1}^{S'(r,n)} P_u(x_{(r,k)}) \\ &= (1-\theta_i) \sum_{r \in R^{i+1} | r_i = 0} G^i(r) \prod_{k=1}^{S'(r,n)} P_u(x_{(r,k)}) \\ &+ \theta_i \sum_{r \in R^{i+1} | r_i = 1} G^i(r) \left( \prod_{k=1}^{S'(r,n)-1} P_u(x_{(r,k)}) \right) P_u(x_{L(r,S'(r,n))}, \dots, x_n) \end{aligned}$$

If $r_i = 0$ then no recombination took place between sites $i$ and $i+1$, so the sum over $r \in R^{i+1}$ is the same as that over $r \in R^i$. If $r_i = 1$ then the last recombination took place between $i$ and $i+1$, so $U(r, S'(r,n)-1) = i$ and $L(r, S'(r,n)) = i+1$. Consequently,

$$\begin{aligned} P'^{i+1}_{u+1}(x) &= (1-\theta_i) \sum_{r \in R^i} G^i(r) \prod_{k=1}^{S'(r,n)} P_u(x_{(r,k)}) \\ &+ \theta_i \sum_{r \in R^{i+1} | r_i = 1} G^i(r) \left( \prod_{k=1}^{S'(r,n)-1} P_u(x_{(r,k)}) \right) P_u(x_{L(r,S'(r,n))}, \dots, x_n) \end{aligned}$$

$$
\begin{aligned}
&= (1-\theta_i)P'^{i}_{u+1}(x_1,\ldots,x_n) \qquad (8.22)\\
&+ \theta_i \sum_{r\in R^{i+1}|r_i=1} G^i(r)\left(\prod_{k=1}^{S'(r,n)-2} P_u(x_{(r,k)})\right) P_u(x_{L(r,S'(r,n)-1)},\ldots,x_i)P_u(x_{i+1},\ldots,x_n)
\end{aligned}
$$

We now replace the sum over $r \in R^{i+1}|r_i = 1$ by a different sum over $r' \in R^i$, where each vector $r'$ corresponds to a vector $r$ without the crossover between sites $i$ and $i+1$:

$$
\begin{aligned}
P'^{i+1}_{u+1}(x) &= (1-\theta_i)P'^{i}_{u+1}(x_1,\ldots,x_n) \qquad (8.23)\\
&+ \theta_i \sum_{r'\in R^i} G^i(r')\left(\prod_{k=1}^{S'(r',n)-1} P_u(x_{(r',k)})\right) P_u(x_{L(r',S'(r',n))},\ldots,x_i)P_u(x_{i+1},\ldots,x_n)\\
&= (1-\theta_i)\cdot P'^{i}_{u+1}(x_1,\ldots,x_n) + \theta_i \cdot P'^{i}_{u+1}(x_1,\ldots,x_i)P_u(x_{i+1},\ldots,x_n)
\end{aligned}
$$

We have replaced $G^i(r)$ with $G^i(r')$ in the transformation from Equation 8.22 to Equation 8.23 since the function $G^i$ is not affected by crossovers after site $i$. The function $S'(r,n)$ in Equation 8.22 counts the total number of crossovers represented by vector $r$. It is replaced by $S'(r',n)+1$ in Equation 8.23 since $r'$ has one fewer crossover than $r$. The product of marginal distributions $\prod_{k=1}^{S'(r,n)-2} P_u(x_{(r,k)})$ in Equation 8.22 is replaced by the product $\prod_{k=1}^{S'(r',n)-1} P_u(x_{(r',k)})$ in Equation 8.23 since it is related only to chromosomal sections preceding site $i$, whose parent haplotypes are identical under $r$ and $r'$. Similarly, $L(r, S'(r,n)-1)$ in Equation 8.22 is replaced with $L(r', S'(r',n))$ in Equation 8.23 since the left edge of the penultimate contiguous section in $r$ that ends at site $i$ becomes the left edge of the last contiguous section in $r'$.

The distribution $P'^{i}_{u+1}$ is the result of intermixing only up to site $i$, so its marginal $P'^{i}_{u+1}(x_{i+1},\ldots,x_n)$ over sites $i+1\ldots n$ is the same as the parent marginal $P_u(x_{i+1},\ldots,x_n)$. Consequently, Equation 8.23 implies:

$$
P'^{i+1}_{u+1}(x) = (1-\theta_i)\cdot P'^{i}_{u+1}(x_1,\ldots,x_n) + \theta_i \cdot P'^{i}_{u+1}(x_1,\ldots,x_i)P'^{i}_{u+1}(x_{i+1},\ldots,x_n) \qquad (8.24)
$$

Equation 8.24 states that the effect of introducing the additional crossover point between sites $i$ and $i+1$ is to reconstitute a proportion $\theta_i$ of the population from the marginal distributions on either side of the crossover point, leaving the remaining proportion $1-\theta_i$ untouched. Equation 8.24 also holds in the following marginal form by summing over $x_1,\ldots,x_{i-1},x_{i+2},\ldots,x_n$:

$$
P'^{i+1}_{u+1}(x_i,x_{i+1}) = (1-\theta_i)\cdot P'^{i}_{u+1}(x_i,x_{i+1}) + \theta_i \cdot P'^{i}_{u+1}(x_i)P'^{i}_{u+1}(x_{i+1})
$$

We now show a similar result for the Markov approximation $Q'^{i}_{u+1}$, defined as follows:

$$
Q'^{i}_{u+1}(x_1,\ldots,x_n) = P'^{i}_{u+1}(x_1)\prod_{j=1}^{n-1} P'^{i}_{u+1}(x_{j+1}|x_j) \qquad (8.25)
$$

The recurrence relation between $Q'^{i+1}_{u+1}$ and $Q'^{i}_{u+1}$ is explicated as follows:

$$
\begin{aligned}
Q'^{i+1}_{u+1}(x) &= P'^{i+1}_{u+1}(x_1)\prod_{j=1}^{n-1} P'^{i+1}_{u+1}(x_{j+1}|x_j)\\
&= P'^{i}_{u+1}(x_1)\prod_{j=1}^{i-1} P'^{i}_{u+1}(x_{j+1}|x_j)\cdot P'^{i+1}_{u+1}(x_{i+1}|x_i)\cdot \prod_{j=i+1}^{n-1} P'^{i}_{u+1}(x_{j+1}|x_j)
\end{aligned}
$$

$$
\begin{aligned}
&= Q'^{i}_{u+1}(x_1,\ldots,x_i)\cdot\frac{P'^{i+1}_{u+1}(x_i,x_{i+1})}{P'^{i+1}_{u+1}(x_i)}\cdot\prod_{j=i+1}^{n-1}P'^{i}_{u+1}(x_{j+1}|x_j)\\
&= Q'^{i}_{u+1}(x_1,\ldots,x_i)\cdot\frac{(1-\theta_i)P'^{i}_{u+1}(x_i,x_{i+1})+\theta_i P'^{i}_{u+1}(x_i)P'^{i}_{u+1}(x_{i+1})}{P'^{i}_{u+1}(x_i)}\cdot\prod_{j=i+1}^{n-1}P'^{i}_{u+1}(x_{j+1}|x_j)\\
&= (1-\theta_i)\cdot Q'^{i}_{u+1}(x_1,\ldots,x_i)\cdot P'^{i}_{u+1}(x_{i+1}|x_i)\cdot\prod_{j=i+1}^{n-1}P'^{i}_{u+1}(x_{j+1}|x_j)\\
&+ \theta_i\cdot Q'^{i}_{u+1}(x_1,\ldots,x_i)\cdot P'^{i}_{u+1}(x_{i+1})\cdot\prod_{j=i+1}^{n-1}P'^{i}_{u+1}(x_{j+1}|x_j)\\
Q'^{i+1}_{u+1}(x) &= (1-\theta_i)\cdot Q'^{i}_{u+1}(x_1,\ldots,x_n)+\theta_i\cdot Q'^{i}_{u+1}(x_1,\ldots,x_i)\cdot Q'^{i}_{u+1}(x_{i+1},\ldots,x_n) \qquad (8.26)
\end{aligned}
$$

We replaced $P'^{i+1}_{u+1}(x_i)$ with $P'^{i}_{u+1}(x_i)$ at several points above since the intermixing process does not affect the marginal allele frequencies for any individual site. Similarly, we replaced $P'^{i+1}_{u+1}(x_{j+1}|x_j)$ with $P'^{i}_{u+1}(x_{j+1}|x_j)$ for any $j \neq i$ since the additional crossover permitted between sites $i$ and $i+1$ only affects marginal distributions containing both $x_i$ and $x_{i+1}$. Equation 8.26 states the analogous result for the series of Markov approximations $Q'^{1}_{u+1}\ldots Q'^{n}_{u+1}$ as Equation 8.24 states for the series of distributions $P'^{1}_{u+1}\ldots P'^{n}_{u+1}$.

Recall that we aim to prove $||P'_{u+1}-Q'_{u+1}|| \leq ||P_u - Q_u||$ for $n \leq 5$. Since $P'^{1}_{u+1} = P_u$ and $P'^{n}_{u+1} = P'_{u+1}$, this inequality can be expressed as $||P'^{n}_{u+1}-Q'^{n}_{u+1}|| \leq ||P'^{1}_{u+1}-Q'^{1}_{u+1}||$. To establish this inequality, we prove that for $1 \leq i \leq n-1$:

$$
\left|\left|P'^{i+1}_{u+1}-Q'^{i+1}_{u+1}\right|\right| \leq \left|\left|P'^{i}_{u+1}-Q'^{i}_{u+1}\right|\right| \qquad (8.27)
$$

We split the proof of Equation 8.27 into two cases, $i=1$ and $i=2$. By considering the haplotypes from their other end points, these proofs also apply respectively for $i=n-1$ and $i=n-2$, due to symmetry. This covers all values of $1 \leq i \leq n-1$ provided $n \leq 5$.

Two properties of variation distance are needed. Given two multivariate distributions $A(x,y)$ and $B(x,y)$ with marginal distributions $A(x)=\sum_y A(x,y)$ and $B(x)=\sum_y B(x,y)$, the first property states that $||A(x,y)-B(x,y)|| \geq ||A(x)-B(x)||$. Given two mixture distributions $A(x) = \alpha A_1(x)+(1-\alpha)A_2(x)$ and $B(x)=\alpha B_1(x)+(1-\alpha)B_2(x)$, the second property states that $||A(x)-B(x)|| \leq \alpha||A_1(x)-B_1(x)||+(1-\alpha)||A_2(x)-B_2(x)||$. Proofs of these two properties are provided in Section 8.4.5.

For $i=1$, we prove Equation 8.27 by rewriting $P'^{2}_{u+1}$ and $Q'^{2}_{u+1}$ in terms of $P'^{1}_{u+1}$ and $Q'^{1}_{u+1}$, using the recurrence relations in Equations 8.24 and 8.26:

$$
\begin{aligned}
P'^{2}_{u+1}(x) &= (1-\theta_1)\cdot P'^{1}_{u+1}(x_1,\ldots,x_n)+\theta_1\cdot P'^{1}_{u+1}(x_1)\cdot P'^{1}_{u+1}(x_2,\ldots,x_n)\\
Q'^{2}_{u+1}(x) &= (1-\theta_1)\cdot Q'^{1}_{u+1}(x_1,\ldots,x_n)+\theta_1\cdot Q'^{1}_{u+1}(x_1)\cdot Q'^{1}_{u+1}(x_2,\ldots,x_n)\\
&= (1-\theta_1)\cdot Q'^{1}_{u+1}(x_1,\ldots,x_n)+\theta_1\cdot P'^{1}_{u+1}(x_1)\cdot Q'^{1}_{u+1}(x_2,\ldots,x_n)
\end{aligned}
$$

The last equality follows because the marginal distribution for an individual site is identical for both $P'^{1}_{u+1}$ and its Markov approximation $Q'^{1}_{u+1}$. The proof of Equation 8.27 for $i=1$ is completed using the two properties of variation distance:

$$
\begin{aligned}
\left|\left|P'^{2}_{u+1}-Q'^{2}_{u+1}\right|\right| &\leq (1-\theta_1)\cdot\left|\left|P'^{1}_{u+1}-Q'^{1}_{u+1}\right|\right|+\\
&\quad \theta_1\cdot\frac{1}{2}\sum_{x_1\ldots x_n}\left|P'^{1}_{u+1}(x_1)P'^{1}_{u+1}(x_2,\ldots,x_n)-P'^{1}_{u+1}(x_1)Q'^{1}_{u+1}(x_2,\ldots,x_n)\right|\\
&= (1-\theta_1)\cdot\left|\left|P'^{1}_{u+1}-Q'^{1}_{u+1}\right|\right|+
\end{aligned}
$$

$$
\begin{aligned}
& \theta_1 \cdot \frac{1}{2} \sum_{x_1} P'^1_{u+1}(x_1) \sum_{x_2 \ldots x_n} \left| P'^1_{u+1}(x_2, \ldots, x_n) - Q'^1_{u+1}(x_2, \ldots, x_n) \right| \\
\leq \; & (1-\theta_1) \cdot \left|\left| P'^1_{u+1} - Q'^1_{u+1} \right|\right| + \theta_1 \cdot \left|\left| P'^1_{u+1} - Q'^1_{u+1} \right|\right| \\
= \; & \left|\left| P'^1_{u+1} - Q'^1_{u+1} \right|\right|
\end{aligned}
$$

For $i = 2$, the proof of Equation 8.27 proceeds similarly:

$$
\begin{aligned}
P'^3_{u+1}(x) &= (1-\theta_2) \cdot P'^2_{u+1}(x_1, \ldots, x_n) + \theta_2 \cdot P'^2_{u+1}(x_1, x_2) \cdot P'^2_{u+1}(x_3, \ldots, x_n) \\
Q'^3_{u+1}(x) &= (1-\theta_2) \cdot Q'^2_{u+1}(x_1, \ldots, x_n) + \theta_2 \cdot Q'^2_{u+1}(x_1, x_2) \cdot Q'^2_{u+1}(x_3, \ldots, x_n) \\
&= (1-\theta_2) \cdot Q'^2_{u+1}(x_1, \ldots, x_n) + \theta_2 \cdot P'^2_{u+1}(x_1, x_2) \cdot Q'^2_{u+1}(x_3, \ldots, x_n) \qquad (8.28)
\end{aligned}
$$

The last equality follows since the joint distribution over any two adjacent sites is unchanged by the Markov approximation. The proof of Equation 8.27 for $i = 2$ is completed using the two properties of variation distance:

$$
\begin{aligned}
\left|\left| P'^3_{u+1} - Q'^3_{u+1} \right|\right| &\leq (1-\theta_2) \cdot \left|\left| P'^2_{u+1} - Q'^2_{u+1} \right|\right| + \\
& \quad \theta_2 \cdot \frac{1}{2} \sum_{x_1 \ldots x_n} \left| P'^2_{u+1}(x_1, x_2) P'^2_{u+1}(x_3, \ldots, x_n) - P'^2_{u+1}(x_1, x_2) Q'^2_{u+1}(x_3, \ldots, x_n) \right| \\
&= (1-\theta_2) \cdot \left|\left| P'^2_{u+1} - Q'^2_{u+1} \right|\right| + \qquad (8.29) \\
& \quad \theta_2 \cdot \frac{1}{2} \sum_{x_1, x_2} P'^2_{u+1}(x_1, x_2) \sum_{x_3 \ldots x_n} \left| P'^2_{u+1}(x_3, \ldots, x_n) - Q'^2_{u+1}(x_3, \ldots, x_n) \right| \\
&\leq (1-\theta_2) \cdot \left|\left| P'^2_{u+1} - Q'^2_{u+1} \right|\right| + \theta_2 \cdot \left|\left| P'^2_{u+1} - Q'^2_{u+1} \right|\right| \\
&= \left|\left| P'^2_{u+1} - Q'^2_{u+1} \right|\right|
\end{aligned}
$$

The proofs for $i = n-1$ and $i = n-2$ are obtained by reversing the order of the conditional probabilities in the Markov chain. Since this covers all possible values of $1 \leq i \leq n-1$ provided $n \leq 5$, this establishes the inequality $||P'^{i+1}_{u+1} - Q'^{i+1}_{u+1}|| \leq ||P'^{i}_{u+1} - Q'^{i}_{u+1}||$ and therefore that $||P'_{u+1} - Q'_{u+1}|| \leq ||P_u - Q_u||$ for $n \leq 5$, as stated in Equation 8.19.

For $n > 5$, this method breaks down in Equation 8.28 for $i = 3$ since the marginal distribution $Q'^3_{u+1}(x_1, x_2, x_3)$ of the Markov approximation cannot be substituted by the marginal $P'^3_{u+1}(x_1, x_2, x_3)$. This in turn prevents the common factor $P'^3_{u+1}(x_1, x_2, x_3)$ from being extracted in Equation 8.29 and summed over $\sum_{x_1, x_2, x_3}$ to unity. A different form of proof would therefore be required to establish Equation 8.19 for all $n$, as we conjecture.

### 8.4.4 Markov Invariance

In this section we prove that the Markov approximations of the distributions arising from intermixing and meiosis are identical:

$$
\left|\left| Q_{u+1} - Q'_{u+1} \right|\right| = 0 \qquad (8.30)
$$

To prove Equation 8.30, it is sufficient to prove that $P_{u+1}(x_i, x_{i+1}) = P'_{u+1}(x_i, x_{i+1})$ for all $i = 1 \ldots n-1$ since the Markov approximations $Q_{u+1}$ and $Q'_{u+1}$ are defined solely in terms of these joint distributions between adjacent sites.

We compute $P_{u+1}(x_i, x_{i+1})$ by marginalizing $P_{u+1}(x)$, as given in Equation 8.4:

$$
P_{u+1}(x_i, x_{i+1}) = \sum_r G(r) \sum_{y^1, y^2 | y_i^{S(r,i)} = x_i, y_{i+1}^{S(r,i+1)} = x_{i+1}} P_u(y_i^1, y_{i+1}^1) P_u(y_i^2, y_{i+1}^2)
$$

We now split the sum over $r$ into two. If $r_i = 0$ then there is no crossover between sites $i$ and $i+1$. In this case, $S(r,i) = S(r,i+1)$ yielding that both sites in $x$ originate from the same parent. If $r_i = 1$ then there is a crossover between sites $i$ and $i+1$. In this case, $S(r,i) \neq S(r,i+1)$ yielding that each site in $x$ originates from a different parent. Therefore:

$$P_{u+1}(x_i, x_{i+1}) = \sum_{r|r_i=0} G(r) P_u(x_i, x_{i+1}) + \sum_{r|r_i=1} G(r) P_u(x_i) P_u(x_{i+1})$$

Using the definition of $G(r)$ in Equation 8.3, it follows that $\sum_{r|r_i=0} G(r) = 1-\theta_i$ and $\sum_{r|r_i=1} G(r) = \theta_i$. Consequently, $P_{u+1}(x_i, x_{i+1}) = (1-\theta_i) \cdot P_u(x_i, x_{i+1}) + \theta_i \cdot P_u(x_i) P_u(x_{i+1})$. This result corresponds with the intuition that the offspring joint distribution over sites $i$ and $i+1$ is the average of the parent joint distribution and parent marginal distributions, weighted by the probability of a crossover and no crossover respectively. By similar means, it can be shown that $P'_{u+1}(x_i, x_{i+1}) = (1-\theta_i) \cdot P_u(x_i, x_{i+1}) + \theta_i \cdot P_u(x_i) P_u(x_{i+1})$, yielding the desired equality $P_{u+1}(x_i, x_{i+1}) = P'_{u+1}(x_i, x_{i+1})$. This proves Equation 8.30.

### 8.4.5 Properties of Variation Distance

The first property relates the variation distance between two multivariate distributions $A(x,y)$ and $B(x,y)$ to the variation distance between the two marginal distributions $A(x) = \sum_y A(x,y)$ and $B(x) = \sum_y B(x,y)$:

$$\begin{aligned}
||A(x,y) - B(x,y)|| &= \frac{1}{2} \sum_x \sum_y |A(x,y) - B(x,y)| \\
&\geq \frac{1}{2} \sum_x \left| \sum_y \{A(x,y) - B(x,y)\} \right| \\
&= \frac{1}{2} \sum_x |A(x) - B(x)| \\
&= ||A(x) - B(x)||
\end{aligned}$$

The second property relates the variation distance between two mixture distributions $A(x) = \alpha A_1(x) + (1-\alpha) A_2(x)$ and $B(x) = \alpha B_1(x) + (1-\alpha) B_2(x)$ to the variation distances between the respective mixture elements:

$$\begin{aligned}
||A(x) - B(x)|| &= \frac{1}{2} \sum_x |A(x) - B(x)| \\
&= \frac{1}{2} \sum_x |\alpha (A_1(x) - B_1(x)) + (1-\alpha) (A_2(x) - B_2(x))| \\
&\leq \frac{1}{2} \sum_x |\alpha (A_1(x) - B_1(x))| + \frac{1}{2} \sum_x |(1-\alpha) (A_2(x) - B_2(x))| \\
&= \alpha \, ||A_1(x) - B_1(x)|| + (1-\alpha) \, ||A_2(x) - B_2(x)||
\end{aligned}$$

## 8.5 Discussion

Our original choice of model was based on extensive population simulations, which showed that a first-order Markov model was able to accurately capture the effects of many generations of random mating, recombination and drift in a growing but finite population. In this chapter we empirically assessed the accuracy of the independent and Markov approximations for representing background

variation in the human genome. Using data taken from the HapMap project, we showed how the approximation error varies for different physical distances and along each autosome, when modeling both haplotype blocks and individual SNPs. Our core observation is that the Markov model over haplotype blocks is particularly accurate at representing markers in strong linkage disequilibrium. By reference to the perturbation process, we explained why the Markov approximation exhibits this behavior only when modeling haplotype blocks, rather than individual SNPs.

Our motivation was to assess whether it is important to use a Markov chain to represent haplotype block variation, or whether an independent model suffices. Clearly, a Markov approximation can represent the variation for a set of markers more accurately than an independent approximation, due to the larger number of parameters available. However our results show an important additional benefit of the Markov model – that when used with haplotype blocks, it is uniquely suited for modeling genomic variation at high density. Models of background variation combining haplotype blocks and a Markov chain have been used by ourselves and others [36, 2, 56, 16].

The error measure we employed is based on the variation distance between a joint distribution and its maximum likelihood approximation. We used this measure because it permits direct comparison between the independent and Markov approximations, and has an intuitive interpretation in terms of the proportion of a distribution misrepresented by its approximation. However, this measure is not ideal, since it is biased by the allele frequencies at individual markers, just like the $|D|$ measure of linkage disequilibrium to which it is related. It would be fruitful to develop an equivalent of the $D'$ linkage disequilibrium measure for the Markov model, in order to overcome this disadvantage. Nonetheless, since our observations in Section 8.1.3 were based on averages over large numbers of sites, this shortcoming bears little relation to the overall patterns observed.

We showed that the unusual accuracy of the Markov model for representing haplotype blocks over short distances stems from the fact that blocks have higher heterozygosity than individual SNPs. It is interesting to ask whether this phenomenon is preserved if SNPs are grouped more simply into multi-allelic markers, without specifically looking for haplotype blocks. Our initial tests show that if sets of 4 or more adjacent SNPs are grouped in this way, the same properties arise as we saw for haplotype blocks. This confirms our result in Section 8.3 that the behavior of the Markov approximation depends on allele diversity, rather than a more specific feature of haplotype blocks. Nonetheless haplotype blocks offer other advantages over arbitrary groups of SNPs in terms of model simplicity and the selection of haplotype tagging SNPs (htSNPs).

In Section 8.3 we referred to the dependency of the Markov model on the balance between the mixing and perturbation processes. Beyond our initial observations, there is work to be done in understanding how these two processes interact, and developing more precise criteria for determining when each plays a more dominant role. It is also desirable to ascertain whether a population must contain highly distinct haplotypes in order for the perturbation effect to be seen. On this point, recent research has found an abundance of common haplotypes which differ at almost every site in human populations [146]. Finally, it would be valuable to generalize the proof in Section 8.3 to a population of finite size, and to extend it to more than $n = 5$ sites.

# Chapter 9

# Future Work

# Introduction

Our research can be divided broadly into two aspects. The first aspect consists of our statistical model and its associated learning algorithms, as described in Chapters 2 and 3. Chapters 7 and 8 are also relevant for our model, in that they address two key questions relating to its design. The second aspect consists of applications of our model to three biological problems – haplotype resolution, LD mapping and the inference of recombination structure, described in Chapters 4, 5 and 6 respectively. In this chapter we briefly revisit these two aspects, discussing ways in which they could be expanded in future.

## 9.1 Statistical Model and Learning

Chapter 2 describes our Bayesian Network model for the multi-variate distribution underlying a set of haplotype or genotype observations. The model considers the haplotypes within each block as descended from a small number of ancestor sequences, upon which subsequent mutations have taken place with site-specific probabilities. It allows each block to have a different number of ancestors, and represents the linkage disequilibrium between blocks using a Markov chain.

Our statistical model was developed when there was little high density SNP data available. As a result, its design was based on population simulations rather than empirical data. Recently this constraint has been removed, due to the rapid development of high-throughput genotyping techniques [9, 1]. Genotyping arrays that measure 100,000 marker sites simultaneously are now available, and this number is set to rise to 500,000 in the coming year. This technology has enabled the International Haplotype Mapping (HapMap) project to reach a density of one SNP per 3 kb [44]. At this level, most high resolution haplotype structure becomes apparent in our model.

By analyzing data from the HapMap project, we validated two of the assumptions that motivated our work. Chapter 8 showed that the Markov chain over haplotype blocks is well suited to representing the variation over closely linked SNPs. Chapter 7 confirmed that haplotype block boundaries are related to recombination hotspots. This means that multiple recombinations have taken place historically on the boundaries between adjacent blocks, supporting our use of a full transition matrix for the Markov model.

One model of the relationship between the ancestor sequences for each block is worth further consideration. We assume that the ancestor sequences are independent, given the allele frequencies of the SNPs within. This ignores any inter-founder relationships that might have been present, especially if the founders of a population came from another population which was itself recently descended from a bottleneck. Ancestor sequences might be better represented via a coalescent-based model, which assigns a prior probability to a set of haplotypes by integrating over unknown factors such as mutation rates, population size and genealogy [58, 62]. A simpler but related alternative is to find the maximum likelihood phylogenetic tree connecting the ancestor sequences, calculating its probability under suitable assumptions [27]. Recently, Li and Stephens model developed a model using an approximation of the coalescent and a model of recombination hotspots that does not assume the presence of haplotype blocks [69].

In Chapter 3 we detailed our heuristic model learning algorithm, which explores the search space of possible models using the addition, nudging and removal steps. It would be valuable to develop a new algorithm which is guaranteed to find the globally optimum model for a given set of observed data. This might be a variation on Zhang *et al*'s dynamic programming algorithm with an appropriate scoring function [149]. Promisingly, Anderson and Novembre developed such an algorithm for their statistical model with a scoring function based on pairs of adjacent blocks, however further work would be required to adapt their algorithm for our model [2].

The description length (DL) schema described in Chapter 3 could be generalized to assess the suitability of competing models for representing genomic variation. Any statistical model $M$ effectively acts as a compression algorithm for data $D$, reducing it to $-\log_2 Pr(D|M)$ bits with

optimal encoding [118]. Therefore, different statistical models such as those by Koivisto *et al.* [59] and Anderson and Novembre [2] could be assessed in terms of their efficiency at compressing a large set of HapMap data. This comparison would also have to consider the number of bits required to express the model itself.

## 9.2 Application to LD Mapping

The core application of our work is high density linkage disequilibrium (LD) mapping, described in Chapter 5. By testing for association on haplotype blocks instead of individual SNPs, our method has the potential to both increase mapping sensitivity and reduce the number of false positives. Unfortunately, due to the lack of publicly-available high density SNP data with associated phenotype information, the analyses in Chapter 5 required phenotypes to be simulated. It is hoped that a publicly funded project will make real-world data available in future, to enable a more realistic evaluation of the performance of our model for LD mapping.

There are three ways in which our work on LD mapping could be expanded. First, the disease model could be generalized to model the phenotypic effects of complex diseases which are dependent on several different but proximate genetic factors (i.e. in different blocks). Second, an algorithm could be developed for identifying haplotype tagging SNPs (htSNPs) on the basis of an ensemble of statistical models, in order to save on genotyping costs in large studies. Third, a finer-grained technique could be developed to perform mapping at a higher resolution than the individual block. This would be especially relevant in a model which considered the relationship between ancestor sequences (see above).

Much of our work has been focused on genotypes, which contain no information on which of the two alleles observed at each site are co-located on the same chromosome. However three recent developments suggest that the genotype phasing problem might become less relevant in future. First, the rapid reduction in laboratory costs has increased the feasibility of genotyping trios instead of unrelated individuals, allowing haplotypes to be inferred at most sites. Second, researchers are constantly developing new techniques for measuring the haplotypes in individual chromosomes in the laboratory – if one of these methods becomes economically viable, *in silico* haplotyping algorithms will no longer be required.

Finally, there has been recent interest in DNA pooling, in which many haplotypes which share some phenotype are pooled together, to generate a frequency measurement for the alleles at each SNP [117, 100]. Both haplotypes and genotypes are special cases of pooled observations, containing one and two haplotypes respectively. The advantage of large DNA pools is that the alleles of many individuals can be measured simultaneously at a reasonable cost. However, as the size of the pool increases, it becomes increasingly difficult to infer phasing information, though some limited approaches have been published [47, 96, 97]. In theory our statistical model can be extended to deal with DNA pools by generalizing the genotype model for more than 2 haplotypes. However the complexity of performing calculations on this model increases exponentially with the number of haplotypes represented, so this will not be practical for pools of more than two individuals.

# Appendix A

# HaploBlock Manual

# Introduction

HaploBlock (`http://bioinfo.cs.technion.ac.il/haploblock/`) is a software package for inferring statistical models of haplotype block variation and applying them for high density haplotype resolution and linkage disequilibrium mapping. HaploBlock is described in these papers:

- Model-based Inference of Haplotype Block Variation, *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology (RECOMB 2003).* Also to appear in *Journal of Computational Biology*, Volume 11, Number 2-3.
- High Density Linkage Disequilibrium Mapping using Models of Haplotype Block Variation. Accepted for the *Twelfth International Conference on Intelligent Systems for Molecular Biology (ISMB 2004)* and to appear in *Bioinformatics*.

HaploBlock is written in ANSI C and available as a command-line executable for Linux, Mac OS X and Sun OS. The HaploBlock package comes with two executables, compiled to deal with two different encodings for SNP marker alleles. The base pair version (`haploblock_b`) reads and writes files where alleles are represented as their bases (i.e. $\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}, -$) and so can deal with up to five allelic variants at each site. The numerical version (`haploblock_n`) reads and writes files where alleles are represented numerically (i.e. $\mathtt{1}, \mathtt{2}$) and can only deal with biallelic data. If possible, `haploblock_n` should be used, since it runs considerably faster than `haploblock_b`. HaploBlock includes utility functions for converting between base pair and numerical encoding.

## A.1 Quick Start

HaploBlock is most commonly used for high density haplotype resolution or linkage disequilibrium mapping with biallelic marker data. This section explains the minimum required for these operations – HaploBlock has many additional parameters which should be understood before using it as part of a scientific study. Almost all of the processing time will be taken by the model inference stage (`-W`) which can be interrupted since models are output while the algorithm progresses. For reasonable results, 10 models should be sampled at the very least.

### A.1.1 Haplotype resolution

- Arrange the genotypes in file *genofile*, formatted as per Section A.2.1 with numerical encoding.
- Run `./haploblock_n -W -g` *genofile* `-m` *modelfile* to sample 100 models in file *modelfile*.
- Run `./haploblock_n -S -g` *genofile* `-m` *modelfile* `-h` *haplofile* to resolve the genotypes using the sampled models, placing the results in file *haplofile*.
- Interpret the haplotype pairs in file *haplofile* according to Section A.2.1.

### A.1.2 Linkage disequilibrium mapping

- Arrange the haplotype or genotype data with phenotype indicators in file *phenofile*, formatted as per Section A.2.1 with numerical encoding.
- Arrange the physical SNP locations in file *mapfile*, formatted as per Section A.2.4.
- If *phenofile* contains haplotypes, run `./haploblock_n -W -h` *phenofile* `-m` *modelfile* to sample 100 models in file *modelfile*. Otherwise, substitute `-g` for `-h`.
- If *phenofile* contains haplotypes, run `./haploblock_n -X -h` *phenofile* `-m` *modelfile* `-y` *mapfile*. Otherwise, substitute `-g` for `-h`.
- Read the posterior probabilities and densities output for each SNP interval.

## A.2 File Formats

On any platform, HaploBlock can read files with Unix (LF), PC (CR+LF) or Mac OS (CR) line endings. Files will be written with the native line endings of the platform on which the executable is running (currently Unix in all cases). In all file formats, each line can end with an optional comment, preceded by a `#` character. Blank lines and leading white space are ignored.

For data representing $a$ possible SNP marker alleles, we require $a + 1$ symbols to represent each haplotype site, since we allow for unknown values. To encode each genotype allele pair, we require $(a + 1)(a + 2)/2$ symbols, to represent all possible unordered pairs of haplotype alleles, including unknowns. The symbols used for haplotype and genotype data in base pair format (for `haploblock_b`) are shown in Table A.1, following IUPAC conventions where possible. The symbols for data in numerical format (for `haploblock_n`) are shown in Table A.2. In each table, ? represents an unknown allele. Note that alleles in base pair encoding are case sensitive.

### A.2.1 Marker data

SNP marker data is represented as a flat file, with each line encoding a single haplotype or genotype, with an optional phenotype indicator. Alleles are encoded by the symbols in Table A.1 for `haploblock_b` and Table A.2 for `haploblock_n`. Each haplotype or genotype in the file must have the same number of SNP markers. An example genotype file with base pair encoding containing 3 SNPs for 2 individuals without phenotypes is shown below:

```
RYP
PCZ
```

The genotypes defined by this file are $([A, G], [C, T], [G, -])$ and $([G, -], [C, C], [-, ?])$. HaploBlock can also read marker data in FASTA format, where each haplotype or genotype is preceded by a line which begins with the `>` character. In FASTA format, haplotypes or genotypes can be broken over multiple lines since `>` acts as a delimiter.

Each haplotype or genotype can have an optional phenotype attached (excluding FASTA format). Phenotypes are indicated by a leading integer $\geq 0$ separated by white space from the marker data for that haplotype or genotype. Haplotypes or genotypes without a phenotype indicator are treated as having unknown phenotype for the purposes of mapping. An example file with numerical encoding containing 48 SNPs for 5 haplotypes, 4 of which have phenotypes:

```
0 212212121121221211000111211112222211212201221122
1 221212010121121111222111212211222111221112122111
2 221111212222112121212122121211211222121120212121 12
0 112111122211121002112112212112112111122001111121
121212101121222121121211121220212121111211212112
```

When using the option `-o` to indicate that input genotypes are trios, each set of three consecutive genotypes in an input file represents a trio. The first two genotypes are for the parents and the last is for the child. Thus a genotype file for $n$ trios would contain $3n$ lines, with the parent genotypes of trio $i \geq 1$ in lines $3i - 2$ and $3i - 1$ and the child genotype in line $3i$. The probability of a trio genotype under the HaploBlock statistical model is considered to be the probability of the two parent genotypes, with the additional phasing contraints that stem from the child genotype.

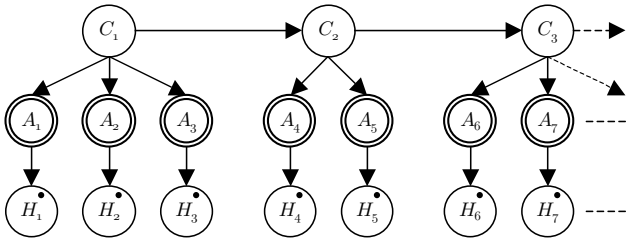
For some of HaploBlock's functions, marker data is interpreted as pairs of haplotypes belonging to individuals. In this case, each sequential pair of haplotypes belongs to the same individual, so that individual $i$'s haplotypes are in positions $2i - 1$ and $2i$ in the file.

In `haploblock_b` only, there is an option `-i` which should be used if marker data was generated by multiple alignment. With this option, HaploBlock will interpret a series of `-` symbols that reaches the start or end of a sequence as unknown alleles, since it is usually a consequence of sequences in the

alignment containing different amounts of genetic data. For example, the sequence `---CT-G-A--` would be read under option `-i` as `???CT-G-A??`.

### A.2.2 Statistical models

A HaploBlock statistical model defines a distribution over haplotypes. The Bayesian Network below depicts an example model, with a random variable $C_k$ for each block $k = 1 \ldots b$ and two random variables $A_j$ and $H_j$ for each SNP $j = 1 \ldots l$. Each variable $C_k$ defines the ancestor from which a haplotype is descended in block $k$. For each block $k$, variables $A_{s_k} \ldots A_{e_k}$ define the sequence of the ancestor indicated by the value of $C_k$, where the first and last SNPs of block $k$ are numbered $s_k$ and $e_k$ respectively. Variables $H_1 \ldots H_l$ define the observed haplotype data over loci $1 \ldots l$.




A particular statistical model is defined by a block partition and parameter vectors $\theta$, $\hat{a}$ and $\mu$, which specify ancestor distributions, ancestor sequences and mutation rates respectively. For the first block, $Pr(C_1 = c) = \theta_{1,c}$ and for subsequent blocks, $Pr(C_k = c | C_{k-1} = c') = \theta_{k,c' \to c}$. For SNP $j$ in block $k$, $Pr(A_j = a | C_k = c) = 1$ if $\hat{a}_{k,c,j} = a$ and 0 otherwise. For SNP $j$, $Pr(H_j = h | A_j = a) = \mu_{j,a \to h}$. For a fuller explanation of this model and its extension to represent genotypes, consult the papers cited in Section A.

A statistical model file represents one or more such models, separated by a line which begins with a hyphen (`-`). For example, the following file contains two trivial models:

```
2
B 1 1.0 AG

- Here is the other

1 2
B 1 1.0 C
T 2 1.0 C T
```

The first line for each model contains a series of integers separated by white space, one for each block $k = 1 \ldots b$ in order. Each integer on this line contains the index $e_k$ of the last SNP in block $k$, so that the last integer gives the total number of SNPs $l$ represented by the model. Multiple models contained within a file must have the same value for $l$.

Every subsequent line of a model description begins with a symbol `B`, `T` or `M`. Lines beginning with `B` specify an ancestor sequence $\hat{a}$, those with `T` specify a Markov transition probability $\theta$ and those with `M` specify a mutation rate $\mu$.

An ancestor sequence line is formatted as either '`B` $k$ $\pi_{k,c}$ $\hat{a}_{k,c}$' or '`B` $k$ $\pi_{k,c}$ $v_{k,c}$ $\hat{a}_{k,c}$', specifying an ancestor with sequence $\hat{a}_{k,c}$ and marginal probability $\pi_{k,c}$ for block $k$. In the second form, $v_{k,c}$ sets a label which may subsequently be used as a shortcut for $\hat{a}_{k,c}$ in the model. The index $c$ is not specified and is assigned automatically. Sequences $\hat{a}_{k,c}$ must contain $e_k - s_k + 1$ characters and are specified using base pair or numerical encoding as appropriate. Labels $v_{k,c}$ must be unique within each block and must not begin with a character that encodes an allele (lower case letters are safe). To aid readability, model files generated by HaploBlock automatically list ancestors for each block

$k$ in descending order of their prior probability $\pi_{k,c}$ and use labels for the ancestors of blocks which contain more than a few SNPs.

A Markov transition line is formatted as '`T` $k$ $\theta_{k,c'\to c}$ $\hat{a}_{k-1,c'}$ $\hat{a}_{k,c}$', specifying the conditional probability $\theta_{k,c'\to c}$ of the ancestor with sequence $\hat{a}_{k,c}$ for block $k$ given that ancestor $\hat{a}_{k-1,c'}$ was present for block $k-1$. As with `B` lines, the indices $c'$ and $c$ are assigned automatically. Sequences $\hat{a}_{k,c}$ and $\hat{a}_{k-1,c'}$ must contain $e_k - s_k + 1$ and $e_{k-1} - s_{k-1} + 1$ characters respectively. If previously set, a label can be used instead of sequence $\hat{a}_{k,c}$ and/or $\hat{a}_{k-1,c'}$. Omitted transitions are assumed to have zero probability. However, if no transitions are specified for block $k$, it is assumed to be independent of the previous block, with the marginal distribution specified by the block's `B` lines.

A mutation rate line is formatted as '`M` $j$ $a$ $h$ $\mu_{j,a\to h}$', specifying the mutation rate at site $j$ from allele $a$ to allele $h \neq a$. The alleles $a$ and $h$ are specified using base pair or numerical encoding as appropriate. Omitted mutations are assumed to have zero probability and the probability of each non-mutation is automatically set to $\mu_{j,a\to a} = 1 - \sum_{h\neq a} \mu_{j,a\to h}$.

Note that model files contain redundancy in order to make them more readable. Ancestor sequences for each block are picked up from both `B` lines and `T` lines. If there are any `T` lines for block $k$, all of the marginal probabilities specified by its `B` lines are ignored and calculated directly from the Markov chain. In practice, this means that `B` lines are only required for the first block.

An example model file with numerical encoding describing a distribution with 2 blocks over 8 SNPs, each with 2 ancestors, is shown below:

```
3 8

M 3 1 2 0.005
M 7 2 1 0.025

B 1 0.6 111
B 1 0.4 221

B 2 0.6 a 21121
B 2 0.4 b 12212

T 1 0.6 111
T 1 0.4 221

T 2 0.8 111 a
T 2 0.2 111 b

T 2 0.3 221 a
T 2 0.7 221 b
```

The parameters defined by this model file are: $b = 2$, $l = 8$, $s_1 = 1$, $e_1 = 3$, $s_2 = 4$, $e_2 = 8$, $q_1 = 2$, $q_2 = 2$, $\mu_{1,1\to1} = 1.0$, $\mu_{1,1\to2} = 0.0$, $\mu_{1,2\to1} = 0.0$, $\mu_{1,2\to2} = 1.0$, ..., $\mu_{3,1\to1} = 0.995$, $\mu_{3,1\to2} = 0.005$, $\mu_{3,2\to1} = 0.0$, $\mu_{3,2\to2} = 1.0$, ..., $\mu_{8,1\to1} = 1.0$, $\mu_{8,1\to2} = 0.0$, $\mu_{8,2\to1} = 0.025$, $\mu_{8,2\to2} = 0.975$, $\hat{a}_{1,1} = 111$, $\hat{a}_{1,2} = 221$, $\hat{a}_{2,1} = 21121$, $\hat{a}_{2,2} = 12212$, $v_{2,1} =$ `a`, $v_{2,2} =$ `b`, $\theta_{1,1} = 0.6$, $\theta_{1,2} = 0.4$, $\theta_{2,1\to1} = 0.8$, $\theta_{2,1\to2} = 0.2$, $\theta_{2,2\to1} = 0.3$, $\theta_{2,2\to2} = 0.7$.

### A.2.3 Allele mapping

An allele mapping file represents a conversion between base pair and numerical allele encoding. Each line $j$ in the file represents SNP $j$. The first symbol in each line contains the base pair for the major allele, encoded numerically as `1`. The second symbol contains the base pair for the minor allele, encoded numerically as `2`. An example allele mapping file for 3 SNPs is shown below:

```
AG
GC
TC
```

The file implies the haplotype mappings $AGT \leftrightarrow 111$, $GCC \leftrightarrow 222$, $GGC \leftrightarrow 212$.

### A.2.4 Physical map

A physical map file describes the relative location of some SNP markers. Each line $j$ in the file specifies the location of SNP $j$. Since a map file is only used to calculate the relative sizes of the intervals between adjacent SNPs, any starting point and unit of measurement can be used. An example physical map file for 4 SNPs is shown below:

```
148.191060
148.192022
148.193108
148.194345
```

## A.3 Function Reference

The first parameter to `haploblock_b` or `haploblock_n` specifies which function to perform. Each function takes a subset of the additional parameters listed in Table A.3, as explained in the sections that follow. Each additional parameter is preceded by an identifying prefix, so these parameters can be specified in any order. Parameters which are omitted receive their default value from Table A.3. The `-r` (reporting level) parameter is accepted by all functions, determining the level of detail with which progress is reported. Within the function descriptions below, the value specified for each parameter is indicated by that parameter's alphabetic symbol in `this typeface`.

If a function is not specified correctly, HaploBlock outputs a list of available function codes and exits. Similarly, if a function's parameters are not specified correctly, HaploBlock outputs a list of possible parameters for the function and exits. Otherwise, HaploBlock displays the function to be performed and the values taken for each parameter before proceeding. Once a function has successfully completed, HaploBlock displays its running time.

### A.3.1 Generate data

#### Generate marker data or models by simulation (`-P`)

This function takes the following parameters from Table A.3: `-c` (population capacity), `-d` (physical length), `-e` (simulation time), `-f` (number of founders), `-h` (haplotype file), `-j` (no Markov chain), `-m` (model file), `-n` (sample size), `-p` (hotspot density), `-s` (SNP density), `-u` (minimum mutation rate), `-v` (maximum mutation rate), `-w` (growth rate), `-x` (crossover density), `-y` (map file).

The population simulation begins with a bottleneck event and proceeds to form new generations based on exponential growth, random mating, no migration, neutral selection and recombination at hotspots only. Note that the simulation is based on many of the assumptions underlying the HaploBlock model, so it is clearly not a basis on which to assess its validity!

The simulation distributes SNPs and recombination hotspots within a chromosomal region of length `d` by a random Poisson process. SNPs are distributed with average density per nucleotide `s` and recombination hotspots are placed independently with density `p`. The crossover probability per generation remains fixed throughout the simulation and is selected randomly and independently for each recombination hotspot from the uniform distribution over $0 \ldots 2 \cdot \mathtt{x} \cdot \mathtt{d}/(t+1)$, where $t$ is the number of hotspots placed.

The population is initiated with `f` founders whose gender is assigned randomly. The SNP alleles on these founders' `2f` haplotypes are assigned randomly, where each SNP is biallelic with equal

probability for each allele. Over `e` generations, each individual in a new generation is descended from a random independent union between a male and female from the previous generation, and has gender assigned randomly. The new individual's haplotypes are obtained from those of its parents by simulated meiosis, in which crossover occurs randomly at each hotspot with the probability calculated above. If the population of generation $i$ is $p_i$, the population of generation $i+1$ is given by $p_{i+1} = (1+\mathtt{w})^{(1-p_i/\mathtt{c})}$, representing an initial growth rate of `w` which tends to zero as the population reaches capacity.

The simulation's output is based on the final generation. SNPs with no variation in this generation are removed, to reflect what would be visible in real-world data. For each remaining SNP, the cumulative mutation rate (over all generations) from each allele to the other is set independently to $\mathtt{u} \cdot \exp(r \cdot \log(\mathtt{v}/\mathtt{u}))$ where $r$ is a random variable distributed uniformly over $0 \ldots 1$. Note that these mutation distributions are only used to generate data once the simulation is complete.

Three types of data can be output from the simulation. If parameter `y` is specified, the SNP locations are output to physical map file `y`. If parameter `h` is specified, `n` haplotypes are sampled (with repeats) from the final generation and output with mutations to file `h`. If parameter `m` is specified, a statistical model is inferred from the final generation and output to file `m`.

When inferring a statistical model from the final generation, the number of ancestors $q_k$ for each block $k$ and their sequences $\hat{a}$ are determined after uniting any block ancestors with identical sequences. The parameters $\theta$ of the Markov chain are determined simply by counting the frequency with which ancestors in adjacent blocks appear together. However, if option `j` is specified, a model is inferred with independent blocks and no Markov chain. The mutation rates $\mu$ in the model are set according to those randomized at the end of the simulation process.

### Generate haplotypes from models (`-H`)

This function takes the following parameters from Table A.3: `-h` (haplotype file), `-k` (unknown rate), `-m` (model file), `-n` (sample size). It generates `n` haplotype samples independently using the model/s in file `m`. If file `m` contains more than one model then each haplotype is based on a model drawn uniformly and independently. To simulate failed measurements in a laboratory, each marker allele is converted to an unknown with probability `k`. The generated haplotype data is output to file `h`.

### Generate genotypes from models (`-G`)

This function takes the following parameters from Table A.3: `-g` (genotype file), `-k` (unknown rate), `-m` (model file), `-n` (sample size). It generates `n` genotype samples independently using the model/s in file `m`. If file `m` contains more than one model then each genotype is based on a model drawn uniformly and independently. Each marker allele is converted to an unknown with probability `k` before pairing. The generated genotype data is output to file `g`.

### Generate quasi-phenotypes from marker data (`-Q`)

This function takes the following parameters from Table A.3: `-d` (disease dominance), `-g` (genotype file), `-h` (haplotype file), `-o` (new haplotype file), `-p` (disease penetrance), `-s` (selected SNP), `-u` (new genotype file), `-y` (map file), `-z` (new map file). Multiple haplotype and/or genotype input files can be specified using multiple `-h` or `-g` parameters, but all must have the same number of SNPs. The function reads in the haplotypes in files `h` and/or the genotypes in files `g`, converts the alleles at a target SNP into phenotypes, and outputs the resulting haplotype and genotype marker files with phenotypes (and minus the target SNP) to `o` and `u` respectively.

If parameter `s` is included, it specifies the index ($\geq 1$) of the target SNP in the marker data, otherwise a target will be selected randomly. If a map file `y` is specified, this random selection takes account of physical distances between SNPs, giving each SNP probability in proportion to the distance between its neighboring SNPs. If no map file is specified, the random selection will be

uniform over all but the first and last SNPs. If parameter `z` is specified, a new physical map file is written to file `z` with the target SNP removed, suitable for use with `o` and `u`.

For haplotype data, the phenotype for each haplotype is assigned based on its allele for the target SNP. The most common allele for the target is mapped to phenotype `0`. Each additional allele observed is mapped to a different phenotype, numbered from `1` upwards in descending order of allele frequency. To simulate penetrance, the phenotype is then set to `0` with probability $1 - \mathtt{p}$ independently of the allele at the target SNP. If a haplotype's allele for the target is unknown, that haplotype is always assigned the unknown phenotype.

For genotype data, the two alleles at the target SNP are converted separately to phenotypes, which are then combined under the dominance model specified by `d`. If parameter `d` is `0` (recessive), the numerically lower number is set as the overall phenotype. If `d` is `1` (codominant), every different (unordered) pair of phenotypes is mapped to a different overall phenotype. If `d` is `2` (dominant), the numerically higher number is set as the overall phenotype. If one of the two phenotypes is unknown, the overall phenotype is set to unknown unless the missing phenotype makes no difference under the dominance model specified. These rules generalize the standard dominance model for more than 2 alleles, and will produce the expected results for biallelic data.

### A.3.2 Infer models from data

#### Infer a single model from marker data (`-F`)

This function takes the following parameters from Table A.3: `-b` (maximum blocks), `-c` (convergence criterion), `-g` (genotype file), `-h` (haplotype file), `-i` (detect alignments), `-j` (no Markov chain), `-l` (initial block length), `-m` (model file), `-o` (genotypes are trios), `-q` (maximum ancestors), `-u` (minimum mutation rate), `-v` (maximum mutation rate), `-z` (resume model search). Multiple haplotype and/or genotype input files can be specified using multiple `-h` or `-g` parameters, but all must have the same number of SNPs. HaploBlock will search for a single model (see RECOMB 2003 paper) for the marker data in `h` and/or `g`, constraining mutation rates by $\mathtt{u} \leq \mu_{j,a \to h} \leq \mathtt{v}$, the number of blocks by $b \leq \mathtt{b}$ and the number of ancestors by $q_k \leq \mathtt{q}$. If option `o` is specified, all genotype input data will be treated as trios (see Section A.2.1). If option `j` is specified, a model will be inferred with independent blocks and no Markov chain. If option `z` is specified, the search begins from the last model in file `m`, otherwise it begins from an initial model which contains evenly-spaced hotspots to ensure that the length of each block is no more than `l`. At the end of each full search round, the best model seen replaces that in file `m`, so this function can be interrupted and later resumed using parameter `z`. If option `c` is specified, the search is stopped when the DL score improved by less than `c` in the last round – otherwise, the search continues indefinitely until no more improvements can be found.

#### Infer a model ensemble from marker data (`-W`)

This function takes the same parameters as function `-F` above, with the addition of `-n` (sample size) and the removal of `-c` (convergence criterion). It infers an ensemble of models (see journal version of RECOMB 2003 paper or ISMB 2004 paper) for the marker data in `h` and/or `g`. After each round of the sampling algorithm, a model is appended to file `m`, so this function can be interrupted and later resumed using parameter `z`.

### A.3.3 Haplotype resolution

#### Join haplotype pairs to form genotypes (`-J`)

This function takes the following parameters from Table A.3: `-h` (haplotype file), `-g` (genotype file), `-i` (detect alignments). It combines the alleles at each SNP for each haplotype pair in file `h`, writing the resulting genotypes to file `g`.

### Haplotype resolution by models (-S)

This function takes the following parameters from Table A.3: `-h` (haplotype file), `-g` (genotype file), `-i` (detect alignments), `-m` (model file). It resolves the genotypes in file `g` by applying the models in file `m` (see RECOMB 2003 paper). If file `m` contains more than one model then the resolution is performed separately for each model and each site in the final haplotype pair is assigned to the allele pair which was inferred most often. For heterozygous sites, the pair is oriented relative to the previous heterozygous site so as to be compatible with the maximum number of the individual model-based resolutions. The inferred haplotype pairs are written to file `h`.

### Haplotype resolution by Clark's algorithm (-L)

This function takes the following parameters from Table A.3: `-h` (haplotype file), `-g` (genotype file), `-i` (detect alignments). It splits the genotypes in file `g` using the Clark algorithm (modified slightly to work with unknowns) as described in *Inference of haplotypes from PCR-amplified samples of diploid populations* (Clark A.G., 1990, Mol Biol Evol. 7:111). Any genotypes which remain unresolved are divided arbitrarily and the inferred haplotype pairs are written to file `h`.

### Haplotype resolution by Local EM (-I)

This function takes the following parameters from Table A.3: `-e` (number of repeats), `-h` (haplotype file), `-g` (genotype file), `-i` (detect alignments). It splits the genotypes in file `g` using a modification of the standard EM haplotype resolution algorithm which overcomes its exponential complexity using a divide-and-conquer approach. This approach is similar to that described in *Bayesian Haplotype Inference for Multiple Linked Single-Nucleotide Polymorphisms* (Niu et al., 2002, Am J. Hum. Genet. 70:157). The inference is performed independently `e` times, after which the set of inferred haplotype pairs with highest likelihood is written to file `h`.

### Haplotype resolution of trios (-T)

This function takes the following parameter from Table A.3: `-h` (haplotype file), `-g` (genotype file), `-i` (detect alignments). It extracts the four parent haplotypes from the trio genotypes in file `g`. For loci at which this is not possible, the output haplotypes are assigned as unknown. This could happen if (a) the trio indicated a Mendelian error, (b) if all three individuals were heterozygous at the site, or (c) if some genotype data was missing.

### Evaluate haplotype resolution (-E)

This function takes the following parameters from Table A.3: `-h` (haplotype file), `-i` (detect alignments), `-t` (test file). Files `h` and `t` should both contain $2n$ haplotypes, where $n$ is the number of individuals, determined automatically from the files. Let $h_{i,1,j}$ and $h_{i,2,j}$ be the respective alleles of the first and second haplotypes of the true pair (from file `h`) for individual $i = 1 \ldots n$ at site $j = 1 \ldots l$. Similarly, let $h'_{i,1,j}$ and $h'_{i,2,j}$ be the respective alleles of the first and second haplotypes of the inferred pair (from file `t`) for individual $i$ at site $j$.

Let the function $\delta(x, y)$ return 1 if alleles $x \neq y$ and 0 otherwise. The value $\lambda_{i,j}$ indicates whether the haplotypes in the true and inferred pairs for individual $i$, oriented as they appear in the files, are incompatible at site $j$, where $\lambda_{i,j} = \delta(h_{i,1,j}, h'_{i,1,j}) \vee \delta(h_{i,2,j}, h'_{i,2,j})$. Similarly, the value $\lambda'_{i,j}$ indicates whether the haplotypes in the true and inferred pairs for individual $i$, oriented in reverse, are incompatible at site $j$, where $\lambda'_{i,j} = \delta(h_{i,1,j}, h'_{i,2,j}) \vee \delta(h_{i,2,j}, h'_{i,1,j})$. Let $\alpha_{i,j}$ denote whether site $j$ of individual $i$ is heterozygous, where $\alpha_{i,j} = \delta(h_{i,1,j}, h_{i,2,j})$. Clearly, for all individuals $i$ and sites $j$, if $\alpha_{i,j} = 1$, exactly one of $\lambda_{i,j}$ or $\lambda'_{i,j}$ is 1, otherwise both $\lambda_{i,j}$ and $\lambda'_{i,j}$ are 0. Let $\beta_i$ be the ordered list of heterozygous sites in individual $i$, so that for all $j = 1 \ldots l$, $\alpha_{i,j} = 1 \Leftrightarrow j \in \beta_i$ and for all $t = 1 \ldots |\beta_i| - 1$, $\beta_{i,t} < \beta_{i,t+1}$.

HaploBlock provides four metrics for measuring phasing errors, calculated from these values. The first metric $\Delta_I$ is the number of individuals who were not phased perfectly, where $\Delta_I = \sum_i \left(\bigvee_j \lambda_{i,j}\right) \wedge \left(\bigvee_j \lambda'_{i,j}\right)$. This measure is common in the literature but provides little information on the degree of correctness of the inferred haplotype pairs. The second metric $\Delta_S$ is the number of sites which were phased incorrectly, taking the better orientation for each individual, where $\Delta_S = \sum_i \min\left(\sum_j \lambda_{i,j}, \sum_j \lambda'_{i,j}\right)$. This provides a good overall measure of the degree of correctness of the phased haplotype pairs if marker order is unimportant.

The third metric $\Delta_A$ is the number of adjacent pairs of sites which were phased incorrectly relative to each other, given by $\Delta_A = \sum_i \sum_{j=1}^{l-1} (\lambda_{i,j} \vee \lambda_{i,j+1}) \wedge \left(\lambda'_{i,j} \vee \lambda'_{i,j+1}\right)$. This measures the local correctness of the phased haplotype pairs and is particularly relevant if the inferred haplotypes are to be used for disease mapping. The fourth metric $\Delta_H$ (sometimes called 'switch rate') is the number of pairs of heterozygous sites (in order but not necessarily adjacent) which were phased incorrectly relative to each other, given by $\Delta_H = \sum_i \sum_{t=1}^{|\beta_i|-1} \left(\lambda_{i,\beta_{i,t}} \vee \lambda_{i,\beta_{i,t+1}}\right) \wedge \left(\lambda'_{i,\beta_{i,t}} \vee \lambda'_{i,\beta_{i,t+1}}\right)$. Note that $\Delta_I = 0 \Leftrightarrow \Delta_S = 0 \Leftrightarrow \Delta_H = 0 \Rightarrow \Delta_A = 0$, for example if true haplotypes $(ACA, TCT)$ were inferred as $(ACT, TCA)$, we would obtain the statistics $\Delta_I = 1, \Delta_S = 1, \Delta_A = 0, \Delta_H = 1$.

When calculating error rates, unknowns in the true haplotypes (file `h`) and inferred haplotypes (file `t`) are dealt with differently. If site $j$ on either true haplotype for individual $i$ is unknown, we automatically exclude that site from consideration, setting $\lambda_{i,j} = \lambda'_{i,j} = \alpha_{i,j} = 0$. However, if the inferred haplotype pair contains an unknown which 'hedges its bets' against a heterozygous site in the true haplotype pair, we set $\lambda_{i,j} = \lambda'_{i,j} = \frac{1}{2}$, rounding each metric as appropriate.

### A.3.4 Linkage disequilibrium mapping

#### LD mapping by models (`-X`)

This function takes the following parameters from Table A.3: `-g` (genotype file), `-h` (haplotype file), `-m` (model file), `-y` (map file). Multiple haplotype and/or genotype input files can be specified using multiple `-h` or `-g` parameters, but all must have the same number of SNPs. The function performs linkage disequilibrium mapping on the phenotyped haplotypes in file `h` and/or the phenotyped genotypes in file `g` by applying the models in file `m` (see ISMB 2004 paper). The file `y` specifies the physical location of the SNPs in the haplotype and genotype files – if no map file is specified, the SNPs are assumed to be uniformly spaced. For each interval between adjacent SNPs, the function outputs the posterior probability that the interval contains the phenotype locus, as well as its posterior density, with standard deviations over the models in `m`.

#### LD mapping by individual SNPs (`-Y`)

This function takes the same parameters as function `-X` above, with the exception of `-m` (model file). The function performs linkage disequilibrium mapping on the phenotyped haplotypes in file `h` and/or the phenotyped genotypes in file `g` by assuming that the alleles at each SNP are independent (see ISMB 2004 paper), producing a similar output to function `-X` above.

### A.3.5 Analyze models

#### Summarize models (`-M`)

This function takes the following parameters from Table A.3: `-m` (model file), `-n` (sample size). Given an input set $\{M^1, \ldots, M^z\}$ of models in file `m`, the function outputs the description length $DL(M^i)$ of each model $M^i$, with parameter accuracy in the model descriptions based on a sample of `n` haplotypes (see RECOMB 2003 paper). It also outputs the proportion of models with a hotspot between SNPs $j-1$ and $j$ for each $j = 2 \ldots l$, given by $\frac{1}{z} \sum_{i=1}^{z} |\{k | s_k^i = j\}|$ where $\square^i$ refers to parameter $\square$ for model $M^i$. Similarly, it outputs the average transition conditional entropy

between SNPs $j-1$ and $j$, given by $\frac{1}{z}\sum_{i=1}^{z}\sum_{k|s_k^i=j}\xi^i_{(k-1)\to k}$. Let $k^i(j)$ be the block in which site $j$ falls in model $M^i$, so that $s^i_{k^i(j)} \le j \le e^i_{k^i(j)}$. HaploBlock outputs the average number of ancestors for each SNP $j = 1 \ldots l$, given by $\frac{1}{z}\sum_{i=1}^{z} q^i_{k^i(j)}$. Lastly, it outputs the average overall site mutation rate for each SNP $j$, given by $\frac{1}{z}\sum_{i=1}^{z}\rho^i_j$, where $\rho^i_j = \sum_{a\in B}(\sum_{h\neq a}\mu^i_{j,a\to h}\cdot\sum_{c|a=\hat{a}^i_{k^i(j),c,j}}\pi^i_{k^i(j),c})$. Standard deviations over the models for these last two statistics are also displayed.

#### Evaluate data under models (`-D`)

This function takes the following parameters from Table A.3: `-g` (genotype file), `-h` (haplotype file), `-i` (detect alignments), `-m` (model file). Multiple haplotype and/or genotype input files can be specified using multiple `-h` or `-g` parameters, but all must have the same number of SNPs as the model file. The function outputs the data probability $Pr(H, G|M^i)$ of the haplotypes and/or genotypes in files `h` and/or `g` under each model $M^i$ in file `m`, as well as the total description length $DL(H, G, M^i)$ (see RECOMB 2003 paper).

#### Compare models (`-V`)

This function takes the following parameters from Table A.3: `-m` (model file), `-n` (sample size), `-t` (test file). The K-L divergence between the haplotype distributions of the true model $M$ (in file `m`) and test model $M'$ (in file `t`) is defined as $\sum_h Pr(h|M)\log\left(Pr(h|M)/Pr(h|M')\right)$. Since an exact calculation of this is infeasible (except in the extreme case where all mutation rates are zero), we generate an unbiased estimate. A list $H$ of `n` haplotypes is drawn randomly and independently from the distribution defined by the true model. The K-L divergence is estimated as $\frac{1}{|H|}\sum_{h\in H}\log\left(Pr(h|M)/Pr(h|M')\right)$. Note that HaploBlock's default value of `n` is inappropriate for this function – for a good estimate, it should be set to at least $10^5$.

### A.3.6 Miscellaneous haplotype operations

#### Randomly reorder haplotypes (`-R`)

This function takes the following parameters from Table A.3: `-h` (haplotype file), `-i` (detect alignments), `-o` (new haplotype file). The function randomly reorders the haplotypes in file `h`, writing the resulting haplotype list to file `o`.

#### Randomly resample haplotypes (`-C`)

This function takes the following parameters from Table A.3: `-h` (haplotype file), `-i` (detect alignments), `-n` (sample size), `-o` (new haplotype file). The function randomly and independently samples `n` haplotypes (with repeats) from the uniform distribution over the haplotypes in file `h`, writing the resulting haplotype list to file `o`.

#### Complete haplotypes by models (`-A`)

This function takes the following parameters from Table A.3: `-h` (haplotype file), `-m` (model file), `-o` (new haplotype file). The function infers the value of any unknown sites in the haplotypes in file `h` using the models in file `m`. This is performed similarly to model-based haplotype resolution by finding the most likely a posteriori assignment of the unknown variables in the model, given the value of the known variables (see RECOMB 2003 paper). If file `m` contains more than one model then the completion is performed separately for each model and each site in the final haplotypes is assigned to the allele which was inferred most often. The inferred haplotypes are written to file `o`.

### A.3.7 Format conversions

#### Convert marker data to numerical encoding (-N)

This function (available in `haploblock_b only`) takes the following parameters from Table A.3: `-a` (allele file), `-h` (haplotype file), `-g` (genotype file), `-i` (detect alignments), `-o` (new haplotype file), `-u` (new genotype file). Multiple haplotype and/or genotype files can be specified using multiple `-h` or `-g` parameters, but all must have the same number of SNPs. The function converts the haplotypes in files `h` and/or the genotypes in files `-g` into numerical encoding by assigning the major observed allele to numerical allele `1` and the minor allele to `2`. If more than two alleles are observed for any site, the conversion will fail. The allele mapping obtained is output to file `a` and the numerically encoded haplotypes and genotypes are output to files `o` and/or `u` respectively.

#### Convert models to base pair encoding (-B)

This function (available in `haploblock_b only`) takes the following parameters from Table A.3: `-a` (allele file), `-m` (model file), `-t` (new model file). It converts the statistical models in file `m` to base pair encoding using the allele mapping in file `a`, and outputs the resulting models in file `t`.

#### Reformat models (-O)

This function takes the following parameters from Table A.3: `-m` (model file), `-t` (new model file). It reads in the statistical models in file `m` and outputs them in file `t`, using default ordering and formatting. This function is useful for making a model file more human readable.

## A.4 Version History

The following versions of HaploBlock have been released publicly:

- **Version 1.3, March 2005**

  Added ability to analyze trio genotype data with `-o` parameter.
  Added `-T` function to resolve trio genotypes into parent haplotypes.
  Added `-c` convergence parameter for model finding function (`-F`).
  Fixed issue with reading some files containing `#` comments.

- **Version 1.2, April 2004**

  Added `-X` function for LD mapping based on block models.
  Added `-Y` function for LD mapping based on individual SNPs.
  Added `-Q` function to generate quasi-phenotypes from marker data.
  Added `-I` function to perform haplotype resolution by Local EM.
  Added `-B` function to convert models to base pair encoding.
  Added `-O` function to reformat models for readability.
  Added physical map file format for use with mapping and other functions.
  Extended model file format to allow a partial Markov chain.
  Extended marker data file format for optional phenotypes.
  Added `-j` option to `-F`, `-W` and `-P` functions to generate models with no Markov chain.
  Added output of haplotypes and physical map from `-P` simulation function.
  Extended `-N` function to convert genotype data.
  Optimized model inference from unphased genotype data.
  Improved convergence testing to reduce EM iterations.

- **Version 1.1, March 2003**

  Added `-W` function to infer an ensemble of models.
  Added `-A` function to complete haplotypes using models.
  Extended model file format to express multiple models in an ensemble.
  Extended all functions to work with multiple models.
  Extended model file format to allow ancestor labeling.
  Added `-z` option to resume model search from point of interruption.
  Added `-b` and `-q` options to specify maximum blocks and ancestors for search.

- **Version 1.0, December 2002**

  Added caching of many model calculations to drastically reduce search time.
  Fixed underflow issue by extracting common log factors where appropriate.
  Added `-l` option to specify initial block length for search.
  Removed prior factor when calculating $T_k$ parameter description length.
  Added `-i` option to detect partial sequences from alignments.
  Added new pairwise measure of haplotype resolution accuracy.

- **Version 1.0 Beta, October 2002**

  Initial release

Table A.1: Base pair allele encoding

| Symbol | Haplotype | Genotype |
|---|---|---|
| A | $A$ | $[A, A]$ |
| M | | $[A, C]$ |
| R | | $[A, G]$ |
| W | | $[A, T]$ |
| L | | $[A, -]$ |
| E | | $[A, ?]$ |
| C | $C$ | $[C, C]$ |
| S | | $[C, G]$ |
| Y | | $[C, T]$ |
| O | | $[C, -]$ |
| F | | $[C, ?]$ |
| G | $G$ | $[G, G]$ |
| K | | $[G, T]$ |
| P | | $[G, -]$ |
| I | | $[G, ?]$ |
| T | $T$ | $[T, T]$ |
| Q | | $[T, -]$ |
| J | | $[T, ?]$ |
| - | $-$ | $[-, -]$ |
| Z | | $[-, ?]$ |
| N | $?$ | $[?, ?]$ |

Table A.2: Numerical allele encoding

| Symbol | Haplotype | Genotype |
|---|---|---|
| 1 | $A$ | $[A, A]$ |
| 0 | $?$ | $[A, a]$ |
| 4 | | $[A, ?]$ |
| 2 | $a$ | $[a, a]$ |
| 5 | | $[a, ?]$ |
| 3 | | $[?, ?]$ |

Table A.3: HaploBlock parameters

| Code | Parameter | Default value | Range | Units |
|---|---|---|---|---|
| `-a` | Allele mapping file | – | – | – |
| `-b` | Maximum blocks | none | $\geq 1$ | blocks (0=no maximum) |
| `-c` | Population capacity | 10,000 | $\geq 1$ | individuals |
| `-c` | Convergence criterion | 0 | $\geq 0$ | DL bits |
| `-d` | Dominance model | 0 | 0\|1\|2 | recessive\|codominant\|dominant |
| `-d` | Physical length | 10,000 | $\geq 1$ | nucleotides |
| `-e` | Number of repeats | 20 | $\geq 1$ | iterations |
| `-e` | Simulation time | 500 | $\geq 0$ | generations |
| `-f` | Bottleneck founders | 20 | $\geq 1$ | individuals |
| `-g` | Genotype file | – | – | – |
| `-h` | Haplotype file | – | – | – |
| `-i` | Detect alignments | off | – | – |
| `-j` | No Markov chain | off | – | – |
| `-k` | Unknown rate | 0.0 | $0.0 \ldots 1.0$ | unknowns/SNP |
| `-l` | Initial block length | 100 | $\geq 0$ | SNPs (0=no blocks) |
| `-m` | Statistical models file | – | – | – |
| `-n` | Sample size | 100 | $\geq 1$ | samples |
| `-o` | New haplotype file | – | – | – |
| `-o` | Genotypes are trios | off | – | – |
| `-p` | Disease penetrance | 1.0 | $0.0 \ldots 1.0$ | probability |
| `-p` | Hotspot density | $10^{-4}$ | $\geq 0.0$ | hotspots/nucleotide |
| `-q` | Maximum ancestors | none | $\geq 1$ | ancestors (0=no maximum) |
| `-r` | Reporting level | 3 | $0 \ldots 4$ | – |
| `-s` | Selected SNP | none | – | – |
| `-s` | SNP density | $10^{-3}$ | $\geq 0.0$ | SNPs/nucleotide |
| `-t` | New model file | – | – | – |
| `-t` | Test file | – | – | – |
| `-u` | Minimum mutation rate | $10^{-6}$ | $0.0 \ldots 1.0$ | mutations/SNP |
| `-u` | New genotype file | – | – | – |
| `-v` | Maximum mutation rate | $10^{-3}$ | $0.0 \ldots 1.0$ | mutations/SNP |
| `-w` | Growth rate | 0.05 | $\geq 0.0$ | rate/generation |
| `-x` | Crossover density | $10^{-8}$ | $0.0 \ldots 1.0$ | crossovers/nucleotide/generation |
| `-y` | Physical map file | – | – | – |
| `-z` | New physical map file | – | – | – |
| `-z` | Resume model search | off | – | – |

# Bibliography


[1] Affymetrix, Inc. http://www.affymetrix.com/.

[2] E. Anderson and J. Novembre. Finding haplotype block boundaries by using the minimum-description-length principle. *Am J Hum Genet*, 73(2):336–54, 2003.

[3] K.G. Ardlie, L. Kruglyak, and M. Seielstad. Patterns of linkage disequilibrium in the human genome. *Nat Rev Genet*, 3(4):299–309, 2002.

[4] N. Arnheim, P. Calabrese, and M. Nordborg. Hot and cold spots of recombination in the human genome: The reason we should find them and how this can be achieved. *Am J Hum Genet*, 73(1):5–16, 2003.

[5] V. Bafna, B. V. Halldorsson, R. Schwartz, A. G. Clark, and S. Istrail. Haplotypes and informative SNP selection algorithms: Don't block out information. In *Proc Seventh Annual Inter Conf on Computational Molecular Biology (RECOMB 2003)*, pages 19–27, April 10–13 2003.

[6] D. Botstein and N. Risch. Discovering genotypes underlying human phenotypes: Past successes for Mendelian disease, future approaches for complex disease. *Nat Genet*, 33:Suppl:228–37, 2003.

[7] L. Cardon and G. Abecasis. Using haplotype blocks to map human complex trait loci. *Trends Genet*, 19(3):135–40, 2003.

[8] L. R. Cardon and J. I. Bell. Association study designs for complex diseases. *Nat Rev Genet*, 2(2):91–9, 2001.

[9] M. Chee, R. Yang, E. Hubbell, A. Berno, X. C. Huang, D. Stern, J. Winkler, D. J. Lockhart, M. S. Morris, and S. P. Fodor. Accessing genetic information with high-density DNA arrays. *Science*, 274(5287):610–14, 1996.

[10] A. G. Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol Biol Evol*, 7(2):111–22, 1990.

[11] A. G. Clark. The role of haplotypes in candidate gene studies. *Genet Epidemiol*, 27(4):321–33, 2004.

[12] P. M. Cook, D. Whitby, M. L. Calabro, M. Luppi, D. N. Kakoola, H. Hjalgrim, K. Ariyoshi, B. Ensoli, A. J. Davison, and T. F. Schulz. Variability and evolution of Kaposi's sarcoma-associated herpesvirus in Europe and Africa. International Collaborative Group. *AIDS*, 13(10):1165–1176, 1999.

[13] R. D. Cook, T. A. Hodgson, A. C. W. Waugh, E. M. Molyneux, E. Borgstein, A. Sherry, C. Gee Teo, and S. R. Porter. Mixed patterns of transmission of human herpesvirus-8 (Kaposi's sarcoma-associated herpesvirus) in Malawian families. *J Gen Virol*, 83(Pt 7):1613–1619, 2002.

[14] F. Corpet. Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Res*, 16(22):10881–10890, 1988.

[15] A. I. Culley, A. S. Lang, and C. A. Suttle. High diversity of unknown picorna-like viruses in the sea. *Nature*, 424(6952):1054–7, 2003.

[16] M. J. Daly, J. D. Rioux, S. F. Schaffner, T.J. Hudson, and E. S. Lander. High-resolution haplotype structure in the human genome. *Nat Genet*, 29(2):229–32, 2001.

[17] R. Dechter. Bucket elimination: A unifying framework for probabilistic inference. In *Proc Twelfth Conf on Uncertainty in Artificial Intelligence (UAI-96)*, pages 211–219, August 1–4 1996.

[18] B. Devlin and N. Risch. A comparison of linkage disequilibrium measures for fine-scale mapping. *Genomics*, 29(2):311–22, 1995.

[19] C. Ding and C. R. Cantor. Direct molecular haplotyping of long-range genomic DNA with M1-PCR. *PNAS USA*, 100(13):7449–53, 2003.

[20] J. A. Douglas, M. Boehnke, E. Gillanders, J. M. Trent, and S. B. Gruber. Experimentally-derived haplotypes substantially increase the efficiency of linkage disequilibrium studies. *Nat Genet*, 28(4):361–4, 2001.

[21] T. Endo, K. Ikeo, and T. Gojobori. Large-scale search for genes on which positive selection may operate. *Mol Biol Evol*, 13(5):685–690, 1996.

[22] L. Eronen, F. Geerts, and H. Toivonen. A Markov chain approach to reconstruction of long haplotypes. In *Proc 9th Pacific Symp on Biocomputing (PSB '04)*, pages 104–15, January 6–10 2004.

[23] E. Eskin, E. Halperin, and R. M. Karp. Efficient reconstruction of haplotype structure via perfect phylogeny. *J Bioinform Comput Biol*, 1(1):1–29, 2003.

[24] D. M. Evans, L. R. Cardon, and A. P. Morris. Genotype prediction using a dense map of SNPs. *Genet Epidemiol*, 27(4):375–84, 2004.

[25] L. Excoffier and M. Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol Biol Evol*, 12(5):921–7, 1995.

[26] R. Fan and M. Knapp. Genome association studies of complex diseases by case-control designs. *Am J Hum Genet*, 72(4):850–68, 2003.

[27] J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J Mol Evol*, 17(6):368–76, 1981.

[28] S. M. Fullerton, A. G. Clark, K. M. Weiss, D. A. Nickerson, S. L. Taylor, J. H. Stengard, V. Salomaa, E. Vartiainen, M. Perola, E. Boerwinkle, and S. A. Sing. Apolipoprotein E variation at the sequence haplotype level: Implications for the origin and maintenance of a major human polymorphism. *Am J Hum Genet*, 67(4):881–900, 2000.

[29] S. B. Gabriel, S. F. Schaffner, H. Nguyen, J. M. Moore, J. Roy, B. Blumenstiel, J. Higgins, M. DeFelice, A. Lochner, M. Faggart, S. N. Liu-Cordero, C. Rotimi, A. Adeyemo, R. Cooper, R. Ward, E. S. Lander, M. J. Daly, and D. Altshuler. The structure of haplotype blocks in the human genome. *Science*, 296(5576):2225–9, 2002.

[30] H. Geiringer. On the probability theory of linkage in Mendelian heredity. *Ann of Math Stats*, 15:25–37, 1944.

[31] M. J. Gething, J. Bye, J. Skehel, and M. Waterfield. Cloning and DNA sequence of double-stranded copies of haemagglutinin genes from H2 and H3 strains elucidates antigenic shift and drift in human influenza virus. *Nature*, 287(5780):301–306, 1980.

[32] Zoubin Ghahramani. Learning dynamic Bayesian Networks. *Lect Notes in Comp Sci*, 1387:168–197, 1998.

[33] D. Goldstein. Islands of linkage disequilibrium. *Nat Genet*, 29(2):109–11, 2001.

[34] G. Greenspan and D. Geiger. Model-based inference of haplotype block variation. In *Proc Seventh Annual Inter Conf on Computational Molecular Biology (RECOMB 2003)*, pages 131–7, April 10–13 2003.

[35] G. Greenspan and D. Geiger. High density linkage disequilibrium mapping using models of haplotype block variation. *Bioinformatics*, 20(Supplement 1):I137–144, 2004.

[36] G. Greenspan and D. Geiger. Model-based inference of haplotype block variation. *J Comp Biol*, 11(2–3):493–504, 2004.

[37] G. Greenspan and D. Geiger. Modeling haplotype block variation using Markov chains. 2005. Submitted.

[38] G. Greenspan, D. Geiger, F. Gotch, M. Bower, S. Patterson, M. Nelson, B. Gazzard, and J. Stebbing. Model-based inference of recombination hotspots in a highly variable oncogene. *J. Mol. Evol.*, 58(3):239–51, 2004.

[39] G. Greenspan, D. Geiger, F. Gotch, M. Bower, S. Patterson, M. Nelson, B. Gazzard, and J. Stebbing. Recombination does not occur in newly identified diverged oceanic picornaviruses. *J. Mol. Evol.*, 58(3):359–60, 2004.

[40] G. Greenspan, S. Hunt, P. Deloukas, B. Bentley, and R. Durbin. Significant correspondence of recombination events in pedigrees with haplotype block boundaries. In *Keystone Symposia on Human Genome Sequence Variation and the Inherited Basis of Common Disease*, January 8–13 2004.

[41] D. Gusfield. Inference of haplotypes from samples of diploid populations: Complexity and algorithms. *J Comp Biol*, 8(3):305–23, 2001.

[42] E. Halperin and E. Eskin. Haplotype reconstruction from genotype data using imperfect phylogeny. *Bioinformatics*, 20(12):1842–9, 2004.

[43] M. H. Hansen and B. Yu. Model selection and the principle of minimum description length. *J Am Stat Assoc*, 96(454):746–74, 2001.

[44] The International HapMap Consortium. The International HapMap Project. *Nature*, 426(6968):789–796, 2003.

[45] G. H. Hardy. Mendelian proportions in a mixed population. *Science*, 18:49–50, 1908.

[46] G. S. Hayward. KSHV strains: The origins and global spread of the virus. *Semin Cancer Biol*, 9(3):187–99, 1999.

[47] E. Inbar, B. Yakir, and Darvasi. A. An efficient haplotyping method with DNA pools. *Nucleic Acids Res*, 30(15), 2002.

[48] A. Jeffreys, L. Kauppi, and R. Neumann. Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex. *Nat Genet*, 29(2):217–222, 2001.

[49] A. Jeffreys, A. Ritchie, and R. Neumann. High resolution analysis of haplotype diversity and meiotic crossover in the human TAP2 recombination hotspot. *Hum Mol Genet*, 9(5):725–33, 2000.

[50] F. Jensen. *An Introduction to Bayesian Networks*. Springer Verlag, New York, 1996.

[51] G. C. Johnson, L. Esposito, B. J. Barratt, A. N. Smith, J. Heward, G. Di Genova, H. Ueda, H. J. Cordell, I. A. Eaves, F. Dudbridge, R. C. Twells, F. Payne, W. Hughes, S. Nutland, H. Stevens, P. Carr, E. Tuomilehto-Wolf, J. Tuomilehto, S. C. Gough, D. G. Clayton, and J. A. Todd. Haplotype tagging for the identification of common disease genes. *Nat Genet*, 29(2):233–7, 2001.

[52] D. T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J Mol Biol*, 292(2):195–202, 1999.

[53] R. Judson, J. Stephens, and A. Windemuth. The predictive power of haplotypes in clinical response. *Pharmacogenomics*, 1(1):15–26, 2000.

[54] D. N. Kakoola, J. Sheldon, N. Byabazaire, R. J. Bowden, E. Katongole-Mbidde, T. F. Schulz, and A. J. Davison. Recombination in human herpesvirus-8 strains from Uganda and evolution of the K15 gene. *J Gen Virol*, 82(Pt 10):2393–2404, 2001.

[55] D. Karolchik, A. S. Hinrichs, T. S. Furey, K. M. Roskin, C. W. Sugnet, D. Haussler, and W. J. Kent. The UCSC Table Browser data retrieval tool. *Nucleic Acids Res*, 32:D493–6, 2004.

[56] G. Kimmel and Shamir. R. Maximum likelihood resolution of multi-block genotypes. In *Proc Eighth Annual Inter Conf on Computational Molecular Biology (RECOMB 2004)*, pages 2–9, March 27–31 2004.

[57] M. Kimura. Evolutionary rate at the molecular level. *Nature*, 217(129):624–626, 1968.

[58] J. F. C. Kingman. On the genealogy of large populations. *J Appl Probab*, 19A:27–43, 1982.

[59] M. Koivisto, M. Perola, T. Varilo, W. Hennah, J. Ekelund, M. Lukk, L. Peltonen, E. Ukkonen, and H. Mannila. An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries. In *Proc 8th Pacific Symp on Biocomputing (PSB '03)*, pages 502–13, January 3–7 2003.

[60] A. S. Kondrashov. Direct estimates of human per nucleotide mutation rates at 20 loci causing Mendelian diseases. *Human Mutation*, 21(1), 2002.

[61] A. Kong, D. F. Gudbjartsson, J. Sainz, G. M. Jonsdottir, S. A. Gudjonsson, B. Richardsson, S. Sigurdardottir, J. Barnard, B. Hallbeck, G. Masson, A. Shlien, S. T. Palsson, M. L. Frigge, T. E. Thorgeirsson, J. R. Gulcher, and K. Stefansson. A high-resolution recombination map of the human genome. *Nat Genet*, 31(3):241–7, 2002.

[62] M. K. Kuhner, J. Yamato, and J. Felsenstein. Estimating effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. *Genetics*, 140(4):1421–30, 1995.

[63] S. Kumar and S. Subramanian. Mutation rates in mammalian genomes. *PNAS USA*, 99(2):803–8, 2002.

[64] V. Lacoste, J. P. Judde, J. Briere, M. Tulliez, B. Garin, E. Kassa-Kelembho, J. Morvan, P. Couppie, E. Clyti, J. Forteza Vila, B. Rio, A. Delmer, P. Mauclere, and A. Gessain. Molecular epidemiology of human herpesvirus 8 in Africa: Both B and A5 K1 genotypes, as

well as the M and P genotypes of K14.1/K15 loci, are frequent and widespread. *Virology*, 278(1):60–74, 2000.

[65] V. Lacoste, E. Kadyrova, I. Chistiakova, V. Gurtsevitch, J. G. Judde, and A. Gessain. Molecular characterization of Kaposi's sarcoma-associated herpesvirus/human herpesvirus-8 strains from Russia. *J Gen Virol*, 81(Pt 5):1217–1222, 2000.

[66] J. C. Lam, K. Roeder, and B. Devlin. Haplotype fine mapping by evolutionary trees. *Am J Hum Genet*, 66(2):659–73, 2000.

[67] T. M. Lampinen, S. Kulasingam, J. Min, M. Borok, L. Gwanzura, J. Lamb, K. Mahomed, G. B. Woelk, K. B. Strand, M. L. Bosch, D. C. Edelman, N. T. Constantine, D. Katzenstein, and M. A. Williams. Detection of Kaposi's sarcoma-associated herpesvirus in oral and genital secretions of Zimbabwean women. *J Infect Dis*, 181(5):1785–1790, 2000.

[68] S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Comp Stat Data Analysis*, 19:191–201, 1995.

[69] N. Li and M. Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165(4):2213–33, 2003.

[70] J. S. Liu, C. Sabatti, J. Teng, B. J. Keats, and N. Risch. Bayesian analysis of haplotypes for linkage disequilibrium mapping. *Genome Res*, 11(10):1716–24, 2001.

[71] N. Liu, S. L. Sawyer, N. Mukherjee, A. J. Pakstis, J. R. Kidd, K. K. Kidd, A. J. Brookes, and H. Zhao. Haplotype block structures show significant variation among populations. *Genet Epidemiol*, 27(4):385–400, 2004.

[72] P. M. Lizardi, X. Huang, Z. Zhu, P. Bray-Ward, D. C. Thomas, and D. C. Ward. Mutation detection and single-molecule counting using isothermal rolling-circle amplification. *Nat Genet*, 19(3):225–32, 1999.

[73] J. C. Long, R. C. Williams, and M. Urbanek. An E-M algorithm and testing strategy for multiple-locus haplotypes. *Am J Hum Genet*, 56(3):799–810, 1995.

[74] X. Lu, T. Niu, and J. S. Liu. Haplotype information and linkage disequilibrium mapping for single nucleotide polymorphisms. *Genome Res*, 13(9):2112–7, 2003.

[75] M. H. Malim and M. Emerman. HIV-1 sequence variation: Drift, shift, and attenuation. *Cell*, 104(4):469–472, 2001.

[76] R. L. Marsden, L. J. McGuffin, and D. T. Jones. Rapid protein domain assignment from amino acid sequence using predicted secondary structure. *Protein Sci*, 11(12):2814–2824, 2002.

[77] E. R. Martin, E. H. Lai, J. R. Gilbert, A. R. Rogala, A. J. Afshari, J. Riley, K. L. Finch, J. F. Stevens, K. J. Livak, B. D. Slotterbeck, S. H. Slifer, L. L. Warren, P. M. Conneally, D. E. Schmechel, I. Purvis, M. A. Pericak-Vance, A. D. Roses, and J. M. Vance. SNPing away at complex diseases: Analysis of single-nucleotide polymorphisms around APOE in Alzheimer disease. *Am J Hum Genet*, 67(2):383–94, 2000.

[78] D. J. McGeoch. Molecular evolution of the gamma-Herpesvirinae. *Philos Trans R Soc Lond B Biol Sci*, 356(1408):421–35, 2001.

[79] L. J. McGuffin, K. Bryson, and D. T. Jones. The PSIPRED protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000.

[80] M. McPeek and A. Strahs. Assessment of linkage disequilibrium by the decay of haplotype sharing, with application to fine-scale genetic mapping. *Am J Hum Genet*, 65(3):858–75, 1999.

[81] G. McVean, P. Awadalla, and P. Fearnhead. A coalescent-based method for detecting and estimating recombination from gene sequences. *Genetics*, 160(3):1231–1241, 2002.

[82] Y. X. Meng, T. Sata, F. R. Stamey, A. Voevodin, H. Katano, H. Koizumi, M. Deleon, M. A. De Cristofano, R. Galimberti, and P. E. Pellett. Molecular characterization of strains of human herpesvirus 8 from Japan, Argentina and Kuwait. *J Gen Virol*, 82(Pt 3):499–506, 2001.

[83] Y. X. Meng, T. J. Spira, G. J. Bhat, C. J. Birch, J. D. Druce, B. R. Edlin, R. Edwards, C. Gunthel, R. Newton, F. R. Stamey, C. Wood, and P. E. Pellett. Individuals from North America, Australasia, and Africa are infected with four different genotypes of human herpesvirus 8. *Virology*, 261(1):106–119, 1999.

[84] S. Michalatos-Beloin, S. A. Tishkoff, K. L. Bentley, K. K. Kidd, and G. Ruano. Molecular haplotyping of genetic markers 10 kb apart by allele-specific long-range PCR. *Nucleic Acids Res*, 24(23):4841–3, 1996.

[85] J. Molitor, P. Marjoram, and D. Thomas. Application of bayesian spatial statistical methods to analysis of haplotypes effects and gene mapping. *Genet Epidemiol*, 25(2):95–105, 2003.

[86] A. Morris, J. Whittaker, and D. Balding. Bayesian fine-scale mapping of disease loci, by hidden Markov models. *Am J Hum Genet*, 67(1):155–69, 2000.

[87] A. P. Morris, J. C. Whittaker, C. F. Xu, L. K. Hosking, and D. J. Balding. Multipoint linkage-disequilibrium mapping narrows location interval and identifies mutation heterogeneity. *PNAS USA*, 100(23), 2003.

[88] A.P. Morris, J.C. Whittaker, and D.J. Balding. Fine-scale mapping of disease loci via shattered coalescent modeling of genealogies. *Am J Hum Genet*, 70(3):686–707, 2002.

[89] J. M. Murphy. Pirated genes in Kaposi's sarcoma. *Nature*, 385(6614):296–297, 1997.

[90] M. W. Nachman and S. L. Crowell. Estimate of the mutation rate per nucleotide in humans. *Genetics*, 156(1):297–304, 2000.

[91] J. Nicholas, J. C. Zong, D. J. Alcendor, D. M. Ciufo, L. J. Poole, R. T. Sarisky, C. J. Chiou, X. Zhang, X. Wan, H. G. Guo, M. S. Reitz, and G. S. Hayward. Novel organizational features, captured cellular genes, and strain variability within the genome of KSHV/HHV8. *J Natl Cancer Inst Monogr*, (23):79–88, 1998.

[92] T. Niu, Z. S. Qin, X. Xu, and J. S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Am J Hum Genet*, 70(1):157–69, 2002.

[93] J. Ohashi and K. Tokunaga. The power of genome-wide association studies of complex disease genes: Statistical limitations of indirect approaches using SNP markers. *J Hum Genet*, 46(8):478–82, 2001.

[94] N. Patil, A. J. Berno, D. A. Hinds, W. A. Barrett, J. M. Doshi, C. R. Hacker, C. R. Kautzer, D. H. Lee, C. Marjoribanks, D. P. McDonough, B. T. Nguyen, M. C. Norris, J. B. Sheehan, N. Shen, D. Stern, R. P. Stokowski, D. J. Thomas, M. O. Trulson, K. R. Vyas, K. A. Frazer, S. P. Fodor, and D. R. Cox. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294(5547):1719–23, 2001.

[95] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 2nd edition, 1988.

[96] I. Pe'er and J. Beckmann. Resolution of haplotypes and haplotype frequencies from SNP genotypes of pooled samples. In *Proc Seventh Annual Inter Conf on Computational Molecular Biology (RECOMB 2003)*, pages 237–46, April 10–13 2003.

[97] I. Pe'er and J. Beckmann. Recovering frequencies of known haplotype blocks from single-nucleotide polymorphism allele frequencies. *Genetics*, 166(4):2001–6, 2004.

[98] I. Pe'er and J. S. Beckmann. On the applicability of a haplotype map to un-assayed populations. *Hum Genet*, 114(2):214–7, 2004.

[99] L. Perrin, L. Kaiser, and S. Yerly. Travel and the spread of HIV-1 genetic variants. *Lancet Infect Dis*, 3(1):22–27, 2003.

[100] R. M. Pfeiffer, J. L. Rutter, M. H. Gail, J. Struewing, and J. L. Gastwirth. Efficiency of DNA pooling to estimate joint allele frequencies and measure linkage disequilibrium. *Genet Epidemiol*, 22(1):94–102, 2002.

[101] M. S. Phillips, R. Lawrence, R. Sachidanandam, A. P. Morris, D. J. Balding, M. A. Donaldson, J. F. Studebaker, W. M. Ankener, S. V. Alfisi, F. S. Kuo, A. L. Camisa, V. Pazorov, K. E. Scott, B. J. Carey, J. Faith, G. Katari, H. A. Bhatti, J. M. Cyr, V. Derohannessian, C. Elosua, A. M. Forman, N. M. Grecco, C. R. Hock, J. M. Kuebler, J. A. Lathrop, M. A. Mockler, E. P. Nachtman, S. L. Restine, S. A. Varde, M. J. Hozza, C. A. Gelfand, J. Broxholme, G. R. Abecasis, M. T. Boyce-Jacino, and L. R. Cardon. Chromosome-wide distribution of haplotype blocks and the role of recombination hot spots. *Nat Genet*, 33(3):382–7, 2003.

[102] L. J. Poole, J. C. Zong, D. M. Ciufo, D. J. Alcendor, J. S. Cannon, R. Ambinder, J. M. Orenstein, M. S. Reitz, and G. S. Hayward. Comparison of genetic variability at multiple loci across the genomes of the major subtypes of Kaposi's sarcoma-associated herpesvirus reveals evidence for recombination and for two distinct types of open reading frame K15 alleles at the right-hand end. *J Virol*, 73(8):6646–6660, 1999.

[103] Z. S. Qin, T. Niu, and J. S. Liu. Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. *Am J Hum Genet*, 71(5):1242–7, 2002.

[104] T. C. Quinn. Population migration and the spread of types 1 and 2 human immunodeficiency viruses. *PNAS USA*, 91(7):2407–2414, 1994.

[105] Y. Rabani, Y. Rabinovich, and A. Sinclair. A computational view of population genetics. *Rand Struct and Alg*, 12(4):313–334, 1998.

[106] B. Rannala and J. P. Reeve. High-resolution multipoint linkage-disequilibrium mapping in the context of a human genome sequence. *Am J Hum Genet*, 69(1):159–78, 2001.

[107] M. J. Rieder, S. L. Taylor, A. G. Clark, and D. A. Nickerson. Sequence variation in the human angiotensin converting enzyme. *Nat Genet*, 22(1):59–62, 1999.

[108] N. Risch. Searching for genetic determinants in the new millennium. *Nature*, 405(6788):847–56, 2000.

[109] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[110] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Ann Stat*, 11:416–431, 1983.

[111] J. J. Russo, R. A. Bohenzky, M. C. Chien, J. Chen, M. Yan, D. Maddalena, J. P. Parry, D. Peruzzi, I. S. Edelman, Y. Chang, and P. S. Moore. Nucleotide sequence of the Kaposi sarcoma-associated herpesvirus (HHV8). *PNAS USA*, 93(25):14862–14867, 1996.

[112] J. A. Schneider, T. E. Peto, R. A Boone, A. J. Boyce, and J. B. Clegg. Direct measurement of the male recombination fraction in the human beta-globin hot spot. *Hum Mol Genetic*, 11(3):207–15, 2002.

[113] R. Schwartz, B. V. Halldorsson, V. Bafna, A. G. Clark, and S. Istrail. Robustness of inference of haplotype block structure. *J Comp Biol*, 10(1):13–9, 2003.

[114] Schwarz, G. Estimating the dimension of a model. *Ann Stat*, 6(2):461–4, 1978.

[115] P. Sebastiani, R. Lazarus, S.T. Weiss, L.M. Kunkel, I.S. Kohane, and M.F. Ramoni. Minimal haplotype tagging. *PNAS USA*, 100(17):9900–5, 2003.

[116] H. Seltman, K. Roeder, and B. Devlin. Evolutionary-based association analysis using haplotype data. *Genet Epidemiol*, 25(1):48–58, 2003.

[117] P. Sham, J. S. Bader, I. Craig, M. O'Donovan, and M. Owen. DNA pooling: A tool for large-scale association studies. *Nat Rev Genet*, 3(11):862–71, 2002.

[118] C. E. Shannon. A mathematical theory of communication. *Bell Sys Tech J*, 27:379–423, 623–656, 1948.

[119] S. T. Sherry, M. Ward, and K. Sirotkin. dbSNP-database for single nucleotide polymorphisms and other classes of minor genetic variation. *Genome Res*, 9(8):677–9, 1999.

[120] N. G. Smith, M. T. Webster, and H. Ellegren. Deterministic mutation rate variation in the human genome. *Genome Res*, 12(9):1350–6, 2002.

[121] The International SNP Map Working Group. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 409(6822):928–33, 2001.

[122] J. Stebbing, D. Bourboulia, M. Johnson, S. Henderson, I. Williams, N. Wilder, M. Tyrer, M. Youle, N. Imami, T. Kobu, W. Kuon, J. Sieper, F. Gotch, and C. Boshoff. Kaposi's sarcoma-associated herpesvirus cytotoxic T lymphocytes recognize and target Darwinian positively selected autologous K1 epitopes. *J Virol*, 77(7):4306–14, 2003.

[123] J. Stebbing, S. Portsmouth, and M. Bower. Insights into the molecular biology and seroepidemiology of Kaposi's sarcoma. *Curr Opin Infect Dis*, 16(1):25–31, 2003.

[124] J. Stebbing, N. Wilder, S. Ariad, and M. Abu-Shakra. Lack of intra-patient strain variability during infection with Kaposi's sarcoma-associated herpesvirus. *Am J Hematol*, 68(2):133–4, 2001.

[125] M. Stephens, N. J. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *Am J Hum Genet*, 68(4):978–89, 2001.

[126] D. O. Stram. Tag SNP selection for association studies. *Genet Epidemiol*, 27(4):365–74, 2004.

[127] A. Templeton, A. Clark, K. Weiss, D. Nickerson, E. Boerwinkle, and C. Sing. Recombinational and mutational hotspots within the human lipoprotein lipase gene. *Am J Hum Genet*, 66(1):69–83, 2000.

[128] A. R. Templeton. A cladistic analysis of phenotypic associations with haplotypes inferred from restriction endonuclease mapping. II. The analysis of natural populations. *Genetics*, 120:1145–1154, 1988.

[129] A. R. Templeton, T. Maxwell, D. Posada, J. H. Stengard, E. Boerwinkle, and C. F. Sing. Tree scanning: A method for using haplotype trees in phenotype/genotype association studies. *Genetics*, 169(1):441–53, 2005.

[130] D. C. Thomas, D. O. Stram, D. Conti, J. Molitor, and P. Marjoram. Bayesian spatial modeling of haplotype associations. *Hum Hered*, 56(1–3):32–40, 2003.

[131] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–80, 1994.

[132] L. Tiret, O. Poirier, V. Nicaud, S. Barbaux, S. M. Herrmann, C. Perret, S. Raoux, C. Francomme, G. Lebard, D. Tregouet, and F. Cambien. Heterogeneity of linkage disequilibrium in human genes has implications for association studies of common diseases. *Hum Mol Genet*, 11(4):419–29, 2002.

[133] S.A. Tishkoff and B.C. Verrelli. Role of evolutionary history on haplotype block structure in the human genome: Implications for disease mapping. *Curr Opin Genet Dev*, 13(6):569–75, 2003.

[134] J. Tost, O. Brandt, F. Boussicault, D. Derbala, C. Caloustian, D. Lechner, and I. G. Gut. Molecular haplotyping at high throughput. *Nucleic Acids Res*, 30(19):e96, 2002.

[135] R. C. Twells, C. A. Mein, M. S. Phillips, J. F. Hess, R. Veijola, M. Gilbey, M. Bright, M. Metzker, B. A. Lie, A. Kingsnorth, E. Gregory, Y. Nakagawa, H. Snook, W. Y. Wang, J. Masters, G. Johnson, I. Eaves, J. M. Howson, D. Clayton, H. J. Cordell, S. Nutland, H. Rance, P. Carr, and J. A. Todd. Haplotype structure, LD blocks, and uneven recombination within the LRP5 gene. *Genome Res*, 13(5):845–55, 2003.

[136] T. Varilo, T. Paunio, A. Parker, M. Perola, J. Meyer, J. D. Terwilliger, and L. Peltonen. The interval of linkage disequilibrium (LD) detected with microsatellite and SNP markers in chromosomes of Finnish populations with different histories. *Hum Mol Genet*, 12(1):51–59, 2003.

[137] M. Verhoeyen, R. Fang, W. M. Jou, R. Devos, D. Huylebroeck, E. Saman, and W. Fiers. Antigenic drift between the haemagglutinin of the Hong Kong influenza strains A/Aichi/2/68 and A/Victoria/3/75. *Nature*, 286(5775):771–776, 1980.

[138] B. D. Walker and B. T. Korber. Immune control of HIV: The obstacles of HLA and viral diversity. *Nat Immunol*, 2(6):473–5, 2001.

[139] J. D. Wall and J. K. Pritchard. Haplotype blocks and linkage disequilibrium in the human genome. *Nat Rev Genet*, 4(8):587–97, 2003.

[140] J.D. Wall and J.K. Pritchard. Assessing the performance of the haplotype block model of linkage disequilibrium. *Am J Hum Genet*, 73(3):502–15, 2003.

[141] N. Wang, J. Akey, K. Zhang, R. Chakraborty, and L. Jin. Distribution of recombination crossovers and the origin of haplotype blocks: The interplay of population history, recombination, and mutation. *Am J Hum Genet*, 71(5):1227–34, 2002.

[142] M. W. Weiss and J. D. Terwilliger. How many diseases does it take to map a gene with SNPs? *Nat Genet*, 26(2):151–7, 2000.

[143] C. Wiuf and D. Posada. A coalescent model of recombination hotspots. *Genetics*, 164(1):407–17, 2003.

[144] A. T. Woolley, C. Guillemette, C. Li Cheung, D. E. Housman, and C. M. Lieber. Direct haplotyping of kilobase-size DNA using carbon nanotube probes. *Nat Biotechnol*, 18(7):760–3, 2000.

[145] Zhang Y., Davis T., Wang X., Deng J., Baillargeon J., Yeh T., Jenson H., and Gao S. Distinct distribution of rare US KSHV genotypes in South Texas. Implications for KSHV epidemiology and evolution. *Ann Epidemiol*, 10(7):470, 2000.

[146] J. Zhang, A.G. Rowe, W.L.and Clark, and K. H. Buetow. Genomewide distribution of high-frequency, completely mismatching SNP haplotype pairs observed to be common across human populations. *Am J Hum Genet*, 73(5):1073–81, 2003.

[147] K. Zhang, J. M. Akey, N. Wang, M. Xiong, R. Chakraborty, and L. Jin. Randomly distributed crossovers may generate block-like patterns of linkage disequilibrium: An act of genetic drift. *Hum Genet*, 113(1):51–9, 2003.

[148] K. Zhang, P. Calabrese, M. Nordborg, and F. Sun. Haplotype block structure and its applications to association studies: Power and study designs. *Am J Hum Genet*, 71(6):1386–94, 2002.

[149] K. Zhang, M. Deng, T. Chen, M. Waterman, and F. Sun. A dynamic programming algorithm for haplotype block partitioning. *PNAS USA*, 99(11):7335–9, 2002.

[150] K. Zhang and L. Jin. HaploBlockFinder: Haplotype block analyses. *Bioinformatics*, 19(10):1300–1, 2003.

[151] K. Zhang, F. Sun, M. S. Waterman, and T. Chen. Haplotype block partition with limited resources and applications to human chromosome 21 haplotype data. *Am J Hum Genet*, 73(1):63–73, 2003.

[152] Y. J. Zhang, T. L. Davis, X. P. Wang, J. H. Deng, J. Baillargeon, I. T. Yeh, H. B. Jenson, and S. J. Gao. Distinct distribution of rare US genotypes of Kaposi's sarcoma-associated herpesvirus (KSHV) in South Texas: Implications for KSHV epidemiology. *J Infect Dis*, 183(1):125–129, 2001.

[153] J. Zong, D. M. Ciufo, R. Viscidi, L. Alagiozoglou, S. Tyring, P. Rady, J. Orenstein, W. Boto, H. Kalumbuja, N. Romano, M. Melbye, G. H. Kang, C. Boshoff, and G. S. Hayward. Genotypic analysis at multiple loci across Kaposi's sarcoma herpesvirus (KSHV) DNA molecules: Clustering patterns, novel variants and chimerism. *J Clin Virol*, 23(3):119–148, 2002.

[154] J. C. Zong, D. M. Ciufo, D. J. Alcendor, X. Wan, J. Nicholas, P. J. Browning, P. L. Rady, S. K. Tyring, J. M. Orenstein, C. S. Rabkin, I. J. Su, K. F. Powell, M. Croxson, K. E. Foreman, B. J. Nickoloff, S. Alkan, and G. S Hayward. High-level variability in the ORF-K1 membrane protein gene at the left end of the Kaposi's sarcoma-associated herpesvirus genome defines four major virus subtypes and multiple variants or clades in different human populations. *J Virol*, 73(5):4156–70, 1999.