

**A Set of Dynamic Programming Algorithms for Haplotype
Block Partitioning and Tag SNP Selection via Haplotype
Data or Genotype Data**

Kui Zhang, Ting Chen, Michael S. Waterman, Fengzhu Sun*

Molecular and Computational Biology Program

Department of Biological Sciences

University of Southern California

1042 West 36th Place, Los Angeles, CA 90089-1113, USA

*To whom correspondence should be addressed.

Fengzhu Sun, PhD

Department of Biological Sciences

University of Southern California

1042 W. 36th Place DRB-288

Los Angeles, CA, 90089-1113

Tel:(213) 740-2413

Fax:(213) 740-2437

Email: fsun@email.usc.edu

Abstract

Recent studies have revealed a haplotype block structure for human genome such that it can be decomposed into large blocks with high linkage disequilibrium (LD) and relatively limited haplotype diversity, separated by short regions of low LD. One of the practical implications of this observation is that only a small number of tag SNPs can be chosen for mapping genes responsible for human complex diseases, which can significantly reduce genotyping effort without much loss of power. In this paper, we summarize the dynamic programming algorithms developed for haplotype block partitioning and tag SNP selection, with a focus on algorithmic consideration. Extensions of the algorithms for use to genotype data from unrelated individuals as well as genotype data from general pedigrees are considered. Finally, we discuss the implications of haplotype blocks and tag SNPs in association studies to search for complex disease genes.

Introduction

The pattern of linkage disequilibrium (LD) plays a central role in genome-wide association studies of identifying genetic variation responsible for common human diseases (Kruglyak 1999; Nordborg and Tavaré, 2002; Risch and Merikangas, 1996; Weiss and Clark, 2002). Comparing with traditional linkage studies, association studies based on LD have two major advantages. First, only unrelated individuals need to be genotyped, which makes it possible to utilize a large number of individuals. Second, because LD reflects a large number of historical recombination events, rather than just those in a pedigree, it is possible to map genes on a fine scale. Single nucleotide polymorphism (SNP) markers are preferred over microsatellite markers for association studies because of their high abundance along the human genome (SNPs with minor allele frequency greater than 0.1 occur once about every 600 basepairs) (Wang et al., 1998), the low mutation rate, and accessibility to high-throughput genotyping.

However, genotyping a large number of individuals for every SNP is still too expensive to be practical using current technologies.

The number of SNPs that are required for a genome-wide association study depends on the pattern of LD. The more rapid the decay of LD, the more SNPs are needed. Previous studies have observed substantial variation in LD patterns across the human genome (Dunning et al., 2000; Taillon-Miller et al., 2000; Eisenbarth I et al., 2001; Reich et al., 2001). Thus, the number of SNPs that are needed for a genome-wide association study has been greatly debated in recent years. The estimations for the number of SNPs for an association study have large variation using either simulations (Kruglyak, 1999) or empirical studies (e.g., Reich et al., 2001). Recent studies have shown that the human genome can be parsed into discrete blocks of high LD interspersed by shorter regions of low or no LD (Daly et al., 2001; Dawson et al., 2002; Gabriel et al., 2002; Johnson et al., 2001; Patil et al., 2001). Only a small number of characteristic ("tag") SNPs are sufficient to capture most of haplotype structure of the human genome in each block (Johnson et al., 2001; Patil, et al., 2001). Thus the required number of SNPs could be greatly reduced without much loss of power for association studies (Zhang et al., 2002a).

Many methods have been proposed to identify haplotype blocks and corresponding tag SNPs (Gabriel et al., 2002; Patil et al., 2001; Wang et al., 2002; Zhang et al., 2002b). It is not obvious which method should be used for haplotype block partition and tag SNP selection, however. Available methods can be classified into two groups. One is to first identify the boundary of blocks, and then select tag SNPs in each resulting block (Daly et al., 2001; Dawson et al., 2002; Gabriel et al., 2002; Wang et al., 2002). The other group of methods partition the haplotypes into blocks to minimize the total number of tag SNPs over a region of interest or the whole genome (Patil et al., 2001; Zhang et al., 2002b; Zhang et al., 2003). In this paper, we review the dynamic programming algorithms developed for haplotype block partitioning and tag SNP selection to minimize the total number of tag SNPs based on either haplotype data or genotype data. Our approaches fall in the second category of methods.

A Dynamic Programming Algorithm for Haplotype Block Partition for the Whole Genome

In a large-scale study of chromosome 21, Patil et al. (2001) identified 20 haplotypes by a rodent-human somatic cell hybrid technique consisting of 24,047 SNPs (with at least 10% minor allele frequency) spanning over 32.4 Mbps. They developed a greedy algorithm to partition the haplotypes into 4,135 haplotype blocks with 4,563 tag SNPs based on two criteria: (1) In each block, at least 80% of the observed haplotypes are represented more than once, and (2) the minimum set of tag SNPs that can distinguishing at least 80% of haplotypes is selected as tag SNPs. Zhang et al. (2002b) followed the definitions of haplotype block and tag SNPs as in Patil et al. (2001). For the same data, they reduced the numbers of blocks and tag SNPs to 2,575 and 3,582, respectively, using a dynamic programming algorithm.

Before recalling the dynamic programming algorithm, we adopt the notation used in Zhang et al. (2002b) and Zhang et al. (2003). Assume that we are given K haplotype samples comprised of n consecutive SNPs: s_1, s_2, \dots, s_n . For simplicity, the SNPs are referred as $1, 2, \dots, n$ in the context. Let h_1, h_2, \dots, h_K be the K haplotype samples. Each haplotype $h_k, k = 1, 2, \dots, K$, can be represented as an n -dimensional vector with the i -th component $h_k(i) = 0, 1$, or 2 being the allele of the k -th haplotype at the i -th SNP locus, where 0 indicates missing data, and 1 and 2 are the two alleles. To make the present paper self-contained, we summarize the definitions of ambiguous and unambiguous haplotypes used in Patil et al. (2001) and Zhang et al. (2002b). Consider haplotypes defined by SNPs i to j . Two haplotypes, h_k and $h_{k'}$, are compatible if the alleles for the two haplotypes are the same at the loci with no missing data, that is $h_k(i) = h_{k'}(i)$, for any $l, i \leq l \leq j$ and $h_k(i)h_{k'}(i) \neq 0$. A haplotype in a block is ambiguous if it is compatible with two other haplotypes that are themselves incompatible. For example, consider three haplotypes $h_1 = (1, 0, 0, 2)$, $h_2 = (1, 1, 2, 0)$, and $h_3 = (1, 1, 1, 2)$. Haplotype h_1 is compatible with haplotypes h_2 and h_3 , but h_2 is not compatible with h_3 because they differ at the third locus. Thus, h_1 is an ambiguous

haplotype, whereas h_2 and h_3 are unambiguous haplotypes. In Patil et al. (2001) and Zhang et al. (2002b), only unambiguous haplotypes were included in the analysis and compatible haplotypes were considered as identical haplotypes.

The following definitions of haplotype block and tag SNP were first used by Patil et al. (2001), and then generalized by Zhang et al. (2002b; 2003). A set of consecutive SNPs can form a block if at least α percent of haplotypes are common haplotypes. The tag SNPs were chosen as the minimum set of SNPs that can distinguish at least α percent of the haplotypes. Due to the small sample size (20 haplotypes) used in Patil et al. (2001), the common haplotypes are those represented more than once. In general, the common haplotypes are defined as those haplotypes with frequency at least γ . Given α and γ , the following functions were defined (Zhang et al., 2002b; Zhang et al., 2003) for a set of consecutive SNPs for SNP i to SNP j :

- $block(i, \dots, j) = 1$ if SNPs from i to j form a block. Otherwise, this function is defined as 0.
- $f(i, \dots, j)$: the number of tag SNPs within a block formed by SNPs from i to j . Given a set of disjoint blocks, $B = \{B_1, B_2, \dots, B_I\}$ and $B_1 \prec B_2 \prec \dots \prec B_I$, where $B_1 \prec B_2$ indicates that the last SNP of B_1 is located before the first SNP of B_2 , (note that the last SNP of B_1 and the first SNP of B_2 are not necessary to be consecutive, thus the interval between them is excluded from this block partition), the total number of tag SNPs for these blocks is defined by $f(B) = \sum_{i=1}^I f(B_i)$.

Using the above criterion for defining blocks and tag SNPs, Zhang et al. (2002b) designed a dynamic programming algorithm to partition haplotypes into blocks with the minimum total number of tag SNPs. Define $S(j)$ to be the number of tag SNPs for the optimal block partition of the first j SNPs, s_1, \dots, s_j and set $S(0) = 0$. Then,

$$S(j) = \min\{S(i-1) + f(i, \dots, j), \text{ if } 1 \leq i \leq j \text{ and } block(i, \dots, j) = 1\}.$$

Using the recursion, a dynamic programming algorithm was developed to compute the minimum number of tag SNPs for all of the n SNPs, $S(n)$, and trace back to find the optimal block partition.

In practice, there may exist several block partitions that give the minimum number of tag SNPs. Zhang et al. (2002b) used another dynamic programming algorithm to find the partition with the minimum number of blocks simultaneously. Let $C(j)$ be the minimum number of blocks of all the block partitions requiring $S(j)$ tag SNPs in the first j SNPs. Then, applying dynamic programming theory again,

$$C(j) = \min\{C(i-1) + 1, \text{ if } 1 \leq i \leq j \text{ and } \text{block}(i, \dots, j) = 1 \\ \text{and } S(j) = S(i-1) + f(i, \dots, j)\}.$$

where $C(0) = 0$. By this recursion, the minimum number of blocks in the partition, C_n , can be computed.

It is worth noting that the problem of finding the minimum number of tag SNPs within a block to uniquely distinguish all the haplotypes is known as the MINIMUM TEST SET problem, which has been proven to be NP-Complete (Garey and Johnson, 1979). Thus, there are no polynomial time algorithms that guarantees to find the optimal solution for any input. However, the number of tag SNPs in a block is generally small, so Zhang et al. (2002b) enumerated all the possible SNP combinations to find them. The complexity of this algorithm can be easily analyzed. The space complexity for this algorithm is $O(K \cdot n)$. Given a block of k SNPs: s_i, \dots, s_{i+k-1} , the computation time for $\text{Block}(i, \dots, i+k-1)$ is $O(K^2 \cdot k)$ because we need to determine if any two of the K haplotypes are compatible at these SNPs in the block. In total, there are at most $O(n \cdot N)$ blocks, which requires $O(K^2 \cdot N^2 \cdot n)$ time for computing all values of $\text{Block}(\cdot)$. N is the number of SNPs contained in the largest block, and $N \ll n$ generally. The enumeration method for computing $f(i, \dots, j)$ costs at most time $O(N^K)$ theoretically but runs much faster in practice. Thus, the overall time complexity becomes $(K^2 \cdot N^{K+2} \cdot n)$.

Other than the definitions of haplotype block and tag SNPs used in the study of Patil et al. (2001) and a series papers of Zhang et al. (2002a, 2002b, 2003), many other definitions for hapotype blocks have been used in previous studies. Daly et al. (2001) identified regions between which there is no apparent recombination event as blocks. Gabriel et al. (2002) looked for areas within which the most pairs of SNPs have high LD measures, for example D' , as blocks. Wang et al. (2002) determined the segments in which there is no evidence for historical recombinations between any pair of SNPs as blocks using the four-gamete test. Other methods for identifying tag SNPs for a given block have also been proposed. For example, Johnson et al. (2001) proposed to choose tag SNPs based on haplotype diversity. We point out that the other definitions for haplotype block and tag SNPs can be easily incorporated into the dynamic programming algorithm. Zhang et al. (2002b) adapted another method for tag SNP selection into their dynamic programming algorithm, in which $f(\cdot)$ was defined as the minimum number of SNPs required to explain a percentage of the total haplotype diversity in the block. The dynamic programming algorithm is still valid (Zhang et al., 2002b). The dynamic programming algorithm also provides a general framework to refine the blocks obtained by other methods. For instance, Wang et al. (2002) used four-gamete test to identify regions in which there are no historical recombinations between each pair of SNPs and define such regions as blocks. However, there may exist many block partitions that satisfy this criteria and it is hard to choose one without additional information. By setting $f(\cdot) = 1$ for all potential blocks, we can use the dynamic programming algorithm to find a block partition with the minimum number of blocks, which is equivalent to find a block partition with minimum number of recombination events.

Dynamic Programming Algorithms for Haplotype Block Partition with Limited Resources

In the above studies, we tried to minimize the total number of tag SNPs for the entire chromosome. However, when resources are limited, investigators may not be able to genotype all the tag SNPs and instead must restrict the number of tag SNPs used in their studies. With a given number of tag SNPs to be genotyped, some of SNPs may be excluded from the analysis. The objective now is to prioritize SNPs and corresponding chromosomal regions for genotyping in association studies with limited resources. To achieve this, Zhang et al. (2003) introduced an additional function to represent the length of a set of consecutive SNPs, s_i, \dots, s_j :

- $L(i, \dots, j)$: the length of SNP i to j . It can be defined as the number of SNPs, $L(i, \dots, j) = j - i + 1$. It can also be set as the actual length of the genome spanning from the i -th SNP to the j -th SNP. Given a set of disjoint blocks, $B = \{B_1, B_2, \dots, B_I\}$, the total length for these blocks is $L(B) = \sum_{i=1}^I L(B_i)$.

Based on the above notation, Zhang et al. (2003) proposed to find the haplotype block partition to maximize the total length of the region included with a fixed number of tag SNPs and gave the following mathematical formulation:

Block Partition with a Fixed Number of Tag SNPs (FTS): Given K haplotypes consisting of n consecutive SNPs, and an integer m , find a set of disjoint blocks $B = \{B_1, B_2, \dots, B_I\}$ with $f(B) \leq m$ such that $L(B)$ is maximized.

This problem was converted to an equivalent, "dual" problem as follows:

Block Partition with a Fixed Genome Coverage (FGC): Given a chromosome with length L , K haplotypes consisting of consecutive n SNPs, and $\beta \leq 1$, find a set of disjoint blocks $B = \{B_1, B_2, \dots, B_I\}$ with $L(B) \geq \beta L$ such that $f(B)$ is minimized.

Zhang et al. (2003) developed a 2-dimensional (2D) dynamic programming algorithm for the **FTS** problem, and then a parametric dynamic programming algorithm for the **FGC** problem.

A 2D Dynamic Programming Algorithm

Let $S(j, k)$ be the maximum length of the genome that is covered by at most k tag SNPs for the optimal block partition of the first j SNPs, $j = 1, 2, \dots, n$. Set $S(0, k) = 0$ for any $k \geq 0$ and $S(0, k) = -\infty$ for any $k < 0$. Then,

$$S(j, k) = \max \begin{cases} S(j-1, k) \\ S(i-1, k - f(i, \dots, j)) + L(i, \dots, j) \\ \text{for all } 1 \leq i \leq j \text{ where } \mathit{block}(i, \dots, j) = 1. \end{cases}$$

Let $B = \{B_1, B_2, \dots, B_I\}$ be the set of disjoint blocks for $S(j, k)$ such that $L(B)$ is maximum with the constraint $f(B) \leq k$. Then either the last block B_J ends before j , such that $S(j, k) = S(j-1, k)$, or B_J ends exactly at j and starts at some i^* , $1 \leq i^* \leq j$, such that $S(j, k) = S(i^*-1, k - f(B_J)) + L(B_J)$. Using this recursion, Zhang et al. (2003) designed a dynamic programming algorithm to compute $S(n, m)$, the maximum length of genome that is covered by m tag SNPs. The optimal block partition B can be found by back tracking the elements of S that contribute to $S(n, m)$.

Zhang et al. also analyzed the complexity of this algorithm for K haplotypes consisting of n SNPs. The space complexity for this algorithm is $O(m \cdot n)$. If the values of $\mathit{block}(\cdot)$, $f(\cdot)$, and $L(\cdot)$ have been pre-computed, the time complexity of this algorithm is $O(N \cdot m \cdot n)$, where N is the number of SNPs contained in the largest block, and $N \ll n$ for large n generally. Since the computation time for $L(\cdot \cdot \cdot)$ is $O(1)$, the overall time complexity becomes $O(K^2 \cdot N^{K+2} \cdot n + N \cdot m \cdot n)$.

A Parametric Dynamic Programming Algorithm

Zhang et al. (2003) proposed a parametric dynamic programming algorithm to solve the **FGC** problem. For a consecutive set of SNPs i, \dots, j , if $block(i, \dots, j) = 1$ and this block is included in the partition, the score for them is defined as $f(\cdot)$, which is the number of tag SNPs in this block. If these SNPs are excluded from the partition, the score for this exclusion is defined as $\lambda L(i, \dots, j)$, where λ is the parameter for deletion and $\lambda \geq 0$. This parameter plays a crucial role in the algorithm. λ can be regarded as the penalty for each unit length of the excluded regions. Using this scoring scheme, they scored a block partition by $f(B) + \lambda L(E)$, where B represents the included blocks, and E represents the excluded SNPs. Let the scoring function $S(j, \lambda)$ be the minimum score for the optimal block partition of the first j SNPs ($j = 1, 2, \dots, n$) with respect to the deletion parameter λ . Let $S(j, \lambda) = 0$. Zhang et al. (2003) applied the dynamic programming algorithm to obtain $S(j, \lambda)$ by the following recursion:

$$S(j, \lambda) = \min \begin{cases} S(i-1, \lambda) + \lambda L(i, \dots, j), 1 \leq i \leq j; \\ S(i-1, \lambda) + f(i, \dots, j), 1 \leq i \leq j \text{ and } block(i, \dots, j) = 1. \end{cases}$$

For any j , if there exists i^* satisfying $1 \leq i^* \leq j$ and $S(j, \lambda) = S(i^*-1, \lambda) + f(i^*, \dots, j)$, then the block $[i^*, \dots, j]$ is included in the partition. Otherwise, there must exist i^* satisfying $1 \leq i^* \leq j$ and $S(j, \lambda) = S(i^*-1, \lambda) + \lambda L(i^*, \dots, j)$, such that the interval $[i^*, \dots, j]$ is excluded from the partition.

Obviously, $S(n, 0) = 0$ since all SNPs are excluded from the block partition, and $S(n, \infty)$ equals the minimum number of tag SNPs for the entire genome because all SNPs are included in the block partition. $S(n, \infty)$ can be obtained by the dynamic programming algorithm developed by Zhang et al. (2002b). For any fixed $\lambda > 0$, the parametric dynamic programming algorithm can compute the optimal solution with included blocks and excluded intervals. Zhang et al. (2003) showed that if

the length of the included blocks equals to βL , then the number of tag SNPs is $S(N, \lambda) - \lambda(1 - \beta)L$ which must be equal to the minimum number of tag SNPs that is necessary to include at least βL of the genome length. Thus, **FGC** problem can be solved by this parametric dynamic programming algorithm.

The parametric dynamic programming algorithm is a natural extension of the classical dynamic programming algorithm and has been served as a classical computational tool in sequence alignment, where the parameters are the weight of matches, mismatches, insertions/deletions and gaps (Gusfield et al., 1994; Waterman et al., 1992). Thus, the methods developed in Gusfield et al., (1994) and Waterman et al. (1992) can be used to study the properties of block partitions according to the deletion parameter λ . According to Waterman et al. (1992), it can be shown that $S(n, \lambda)$ has the following properties:

$S(n, \lambda)$ is an increasing, piecewise-linear, and convex function of λ . The right-most linear segment of $S(n, \lambda)$ is constant. The intercept and slope for $S(n, \lambda)$ for each piecewise-linear segment are the total number of tag SNPs and the total length of excluded intervals, respectively.

Waterman et al. (1992) proposed a method to find $S(n, \lambda)$ for all $\lambda \geq 0$ efficiently, which was outlined in Zhang et al. (2003). Zhang et al. (2003) studied the complexity of the above algorithm. The calculation of $\min_{1 \leq i \leq j} \{S(i-1, \lambda) + f(i, \dots, j)\}$ depends on the block structure of the haplotypes and is the same as the dynamic programming algorithm for the haplotype block partitioning (Zhang et al., 2002b). The parametric algorithm takes $O(N \cdot n^2)$ time to compute $\min_{1 \leq i \leq j} \{S(i-1, \lambda) + \lambda L(i, \dots, j)\}$, where N is the number of SNPs contained in the largest block. However, if $L(\cdot)$ is an additive function, the time can be reduced to $O(N \cdot n)$ time (Waterman et al., 1992; Waterman, 1995). Thus, the total time for finding $S(n, \lambda)$ is $O(K^2 \cdot N^{K+2} \cdot n + N \cdot S \cdot n)$, where K is the total number of haplotype samples, and S is the number of segments in $S(n, \lambda)$, which is less than the total number of tag SNPs.

After finding all the line segments of $S(n, \lambda)$, we know the entire function of $S(n, \lambda)$.

At each intersection point $(x, S(n, x))$, several block partitions with different numbers of tag SNPs and lengths of excluded intervals may have the same score. We choose the right-most one with the maximum number of tag SNPs and the minimum length of excluded intervals. For each line segment between two intersection points, both the total number of tag SNPs and the total length of excluded intervals are constant along this segment, and both are equivalent for the low intersection point between this segment and the previous segment. We can sort the number of tag SNPs by an ascending order according to the deletion parameters at the intersection points. The gaps between these numbers give us information as to how the block partition is affected by the deletion parameter λ .

The above scoring scheme using $L(\cdot)$ treats each SNP or each chromosome segment as equally important. Functional SNPs play a crucial role in association studies because they directly code proteins. Thus, investigators may try to keep as more as possible coding regions instead of non-coding regions. Zhang et al. (2003) proposed weighted schemes for defining function $L(\cdot)$, in which a higher penalty was added to the SNPs in coding regions, to accommodate this information. For example, they gave an example of $L(\cdot)$ as follows:

$$L(i, \dots, j) = j - i + 1 - n_c + T * n_c$$

where n_c is the number of SNPs in coding regions and T is a relatively large positive number.

Haplotype Block Partitioning with Genotype Data from Unrelated Individuals

The above methods (Zhang et al., 2002b; Zhang et al., 2003), along with other methods (Patil et al., 2001; Wang et al., 2002) were initially developed based on haplotype

data. Although laboratory techniques, such as allele-specific long-range PCR (MichalatosBeloin et al., 1996) or diploid-to-haploid conversion (Douglas et al., 2001), have been used to determine haplotypes in diploid individuals, these approaches are technologically demanding and expensive, which makes it impossible to do a large scale study across the whole genome such as done by Patil et al. (2001). For these reasons, multiple large-scale genotype data rather than haplotype data are being generated. It is necessary to develop methods to extract block information from genotype data directly.

In all the three dynamic programming algorithms (Zhang et al., 2002b; Zhang et al., 2003) we studied, the boundary of blocks and the number of tag SNPs depend on three functions for a set of consecutive SNPs, s_i, \dots, s_j : $Block(i, \dots, j)$, $f(i, \dots, j)$ and $L(i, \dots, j)$. $Block(\cdot)$ and $f(\cdot)$ can be efficiently computed from a set of haplotypes and their frequencies. Thus, we define the haplotype block partitioning problem based on genotype data from unrelated individuals as follows.

Suppose we are given genotypes of a group of individuals. For a consecutive set of SNPs s_i, s_{i+1}, \dots, s_j , we can first estimate the haplotype frequencies and then infer the two haplotypes of each individual. To calculate $S(j)$ ($j = 1, 2, \dots, n$), the haplotype frequency and the haplotype pairs carried by each individual are estimated for a set of a consecutive set of SNPs: s_i, s_{i+1}, \dots, s_j , in which i is set equal to j initially. The function, $block(i, \dots, j)$ and $f(i, \dots, j)$ are then computed based on these estimates. If this set of SNPs can form an block, then we extend our inference of haplotypes and their frequencies to the set of SNPs, s_{i-1}, s_i, \dots, s_j , and compute $Block(\cdot)$ and $f(\cdot)$. Otherwise, we repeat the above procedure to compute $S(j+1)$ until $S(n)$ is obtained.

Many methods are available to infer haplotypes and their frequencies based on genotypes of unrelated individuals. They can be divided into those based on combinatorics (Clark, 1990; Gusfield, 2001) and those based on statistics (Excoffier and Slatkin, 1995; Hawley and Kidd, 1995; Lin et al., 2002; Niu et al., 2002; Stephens et al., 2001; Qin et al., 2002). Combinatorics based methods assign the two haplotypes of

an individual first and then the frequencies of haplotypes are estimated based on the assigned haplotypes. While statistical methods first estimate the frequencies of haplotypes and then two haplotypes are assigned to each individual according to the likelihood function. Although there are still debates about the optimal methods for inferring haplotype frequencies and reconstructing two haplotypes for each individual, the methods incorporated in the dynamic programming algorithms should be fast enough to infer haplotypes and their frequencies in all consecutive sets of SNPs that can form a potential block. In the study of Patil et al. (2001), the largest block contains more than 100 SNPs. In addition, we infer haplotypes and their frequencies for each consecutive set of SNPs that can form a potential block, rather than infer them for the whole set of SNPs. The haplotype inference program would be executed many times in a single round. Therefore, we chose the Partition-Ligation-Expectation-Maximization (PL-EM) algorithm for haplotype inference (Qin et al., 2002) and incorporated into dynamic programming algorithms. In the PL-EM algorithm, all of the SNP loci are broken down into "atomistic" units that only contain several SNPs (usually 5-8 SNPs) and have one or two common SNPs with adjacent units. EM algorithm is first employed within each unit to infer the frequencies of haplotypes and haplotypes for each individual, then applied to "ligate" two adjacent partial haplotypes together. In general, EM algorithm is efficient in time and space for small number of SNP markers. Thus, this strategy can solve the speed and memory constraints generally existed in EM algorithm and makes it suitable for large-scale recovery of haplotypes from genotype data.

To recover the haplotypes from genotype data in a large scale, Eskin et al. (2003) combined a local haplotype prediction algorithm and a dynamic programming algorithm together to determine the block boundaries directly from the genotype data. In their local haplotype prediction algorithm, they determined a set of possible haplotype that appear in samples based on imperfect phylogeny, in which the number of haplotypes is much less than then number of haplotypes that are compatible with the genotype of samples. This makes it possible to estimate the frequency of these

haplotypes using EM algorithm for a relative large number of SNPs. However, it is not clear if their local haplotype prediction algorithm could be extended to predict haplotypes for larger number of SNPs, especially for more than 100 SNPs. In this situation, the decreased number of haplotypes that are compatible with imperfect phylogeny could be too large to be handled by EM algorithm. As we mentioned before, there is a largest block with more than 100 SNPs in an empirical study (Patil et al. 2001). Actually, based on the same data and parameter setting, there are six blocks with more than 100 SNPs and 11 blocks with more than 80 SNPs in blocks identified by Zhang et al. (2002b). Our current implementation of EM-PL algorithm could prediction haplotypes from about 100 samples for up to 250 SNPs. This kind of scale should be large enough for most studies.

Haplotype Block Partitioning with Genotype Data from General Pedigrees

Although many methods have been developed for estimating haplotype frequencies using unrelated individuals, pedigree data are routinely collected in genetic studies. The genetic information from relatives in a general pedigree can help us resolve haplotype ambiguity. Even if ambiguities may still exist with data from a very large pedigree, the reduction of haplotype ambiguity can help us improve the efficiency for estimating haplotype frequencies (Becker and Knapp, 2003). Thus, genotype data from general pedigrees combined with that from unrelated individuals can be used to improve the accuracy of the estimates for haplotype frequency and to detect haplotype blocks more reliably. Suppose that the haplotype frequencies can be estimated for each set of consecutive SNPs that can form a potential block, the dynamic programming algorithms for haplotype block partitioning using genotype data from general pedigrees are essentially the same with algorithms outlined in the previous section. Thus, we focus on deriving an EM algorithm for estimating haplotype frequencies from general pedigrees in the rest of this section.

We make two assumptions. First, we assume Hardy-Weinberg equilibrium (HWE) for haplotypes carried by each individual. This is the fundamental assumption of EM algorithm for haplotype frequency estimations (Exofficer et al., 1995; Hawley et al., 1995). Second, we assume that there is no recombination event within a family for a set of consecutive SNPs within the family, s_i, \dots, s_j . The rationale behind this assumption is that there is high LD and relatively low recombination in each block. Under this assumption, each haplotype is recoded as an allele at a single multiallelic locus. The multi-locus genotypes can be represented as single-locus genotypes using the recoded alleles. Laws of Mendelian inheritance (O’Connell et al., 2000) are assumed to be hold.

We introduce the following notation in our computations. Suppose that there are total of K haplotypes comprised of SNPs s_i, \dots, s_j : $\mathbb{H} = \{h_1, h_2, \dots, h_K\}$. Their frequencies in the population are represented as $\Theta = \{\theta_1, \theta_2, \dots, \theta_K\}$. Assume that we have a total of F families. We define an individual in a pedigree as a non-founder if both parents are available and others are referred as founders. In a family f ($1 \leq f \leq F$), let n_f be the number of founders and m_f be the total number of individuals in the family. The n_f founders are indexed as $1, \dots, n_f$ and the $m_f - n_f$ non-founder individuals are indexed as $n_f + 1, \dots, m_f$. The genotype and two haplotypes of individual r in family f are denoted as G_{f_r} and $H_{f_r} = \{h_{f_r,1}, h_{f_r,2}\}$, respectively. It is likely that many haplotype pairs are compatible with the genotypes of families under the assumption of no recombination events. Let \mathbb{S}_{f_r} denote all the compatible haplotype pairs for individual r in family f . It is worth noting that the unrelated individuals can be included into our model. Each individual forms a family and is the only founder member in this family. In this case, \mathbb{S} contains all possible haplotype pairs compatible with the genotype of this individual.

The objective is to estimate the haplotype frequencies based on the genotypes of the

families. For a family f , the likelihood of the genotypes of the family is:

$$\begin{aligned}
L_f(G_{f_1}, \dots, G_{f_{m_f}} | \Theta) &= \sum_{H_{f_1} \in \mathbb{S}_{f_1}} \dots \sum_{H_{f_{m_f}} \in \mathbb{S}_{f_{m_f}}} Pr(G_{f_1}, \dots, G_{f_{m_f}}, H_{f_1}, \dots, H_{f_{m_f}} | \Theta) \\
&= \sum_{H_{f_1} \in \mathbb{S}_{f_1}} \dots \sum_{H_{f_{m_f}} \in \mathbb{S}_{f_{m_f}}} Pr(G_{f_1}, \dots, G_{f_{m_f}} | H_{f_1}, \dots, H_{f_{m_f}}, \Theta) \\
&\quad Pr(H_{f_1}, \dots, H_{f_{m_f}} | \Theta) \\
&= \sum_{H_{f_1} \in \mathbb{S}_{f_1}} \dots \sum_{H_{f_{m_f}} \in \mathbb{S}_{f_{m_f}}} \prod_{r=1}^{n_f} Pr(H_{f_r} | \Theta) \prod_{r'=n_f+1}^{m_f} Pr(H_{f_{r'}} | H_{f_{r'}}^F, H_{f_{r'}}^M)
\end{aligned}$$

where $Pr(H_{f_r} | \Theta) = 2\theta_{f_r,1}\theta_{f_r,2}$ if $h_{f_r,1} \neq h_{f_r,2}$ and $Pr(H_{f_r} | \Theta) = \theta_{f_r,1}\theta_{f_r,2}$ if $h_{f_r,1} = h_{f_r,2}$ for $r = 1, \dots, n_f$ under the assumption of HWE; and $Pr(H_{f_{r'}} | H_{f_{r'}}^F, H_{f_{r'}}^M)$ ($r' = n_f + 1, \dots, m_f$) is the gamete transmission probabilities for unordered genotypes where $H_{f_{r'}}^F$ and $H_{f_{r'}}^M$ are the haplotype pairs for the father and mother, respectively (Elston and Stewart, 1971). The likelihood of all the data is obtained by multiplying $L_f(G_{f_1}, \dots, G_{f_{m_f}} | \Theta)$ across all the families. We then estimate θ using the maximum likelihood estimation (MLE) approach. It is difficult to directly obtain the MLE of Θ . EM algorithms are frequently used to estimate Θ .

Suppose that we know that $\Theta = \Theta^{(k)}$ and we want to estimate $\Theta^{(k+1)}$. In the E-step, we introduce the following notation. Let

$$\begin{aligned}
\alpha_{l,f}(H_{f_1}, \dots, H_{f_{n_f}}) &= \#\{h_l \text{ in } H_{f_1}, \dots, H_{f_{n_f}}\} \\
&= \sum_{i=1}^{n_f} (I_{h_{f_i,1}}(h_l) + I_{h_{f_i,2}}(h_l)),
\end{aligned}$$

which is the number of haplotype l present in the founders of family f ($l = 1, \dots, K$).

Let

$$\beta_{l,f}^{(k)} = \sum_{H_{f_1} \in \mathbb{S}_{f_1}} \dots \sum_{H_{f_{m_f}} \in \mathbb{S}_{f_{m_f}}} \alpha_{l,f}(H_{f_1}, \dots, H_{f_{n_f}}) \prod_{r=1}^{n_f} Pr(H_{f_r} | \Theta^{(k)}) \prod_{r'=n_f+1}^{m_f} Pr(H_{f_{r'}} | H_{f_{r'}}^F, H_{f_{r'}}^M),$$

which is the weighted number of haplotype l appeared in the founders of family f . We define an normalization constant $C_f^{(k)}$ satisfying $C_f^{(k)} \sum_{l=1}^N \beta_{l,f}^{(k)} = 2n_f$ to evade the calculation of Mendelian likelihood of the whole family. In the M-step, we can

estimate Θ^{k+1} from all families as follows:

$$\theta_l^{(k+1)} = \left(\sum_{f=1}^F C_f^{(k)} \beta_{l,f}^{(k)} \right) / \left(2 \sum_{f=1}^F n_f \right) \quad (l = 1, 2, \dots, K).$$

The EM-based method of estimating haplotype frequencies can be very computationally intensive, since the likelihood for each configuration of founder haplotype pairs is computed during each iteration. Even if a large pedigree is very helpful to eliminate the number of compatible haplotype configurations to reduce the computation, this number could still be too large to make the computation practical, especially for a large number of SNP loci with the presence of missing data. In general, the genotype elimination method (Lange and Goridia, 1987) can be used to accelerate the likelihood evaluation (O’Connell, 2000) in the first step. However, the genotype elimination algorithm itself can be very time consuming too. We propose to use a rule-based method to partially assign the haplotypes to each individual (Wijsman, 1987) first and then perform genotype elimination. Several other techniques, such as set recoding (O’Connell and Weeks, 1995) and the partition-ligation technique (Qin et al., 2002) can further reduce the complexity in performing genotype elimination and EM algorithm.

Discussion

Several recent studies have suggested that the human genome can be divided into blocks with high LD within each block. Due to this feature, a relative small fraction of SNPs can capture most of the haplotypes in each block. In this paper, we review a set of dynamic programming algorithms for haplotype block partitioning and tag SNP selection (Zhang et al., 2002b; Zhang et al., 2003). These algorithms guarantee to find the blocks with minimum number of tag SNPs. Thus, genotyping efforts can be reduced as much as possible. These algorithms also provided a general framework to accommodate other block definitions and criteria for tag SNP selection. For example, Schwartz et al. (2003) examined the robustness of inference of haplotype

block structure for three methods based on two optimal criteria: minimizing the total number of blocks and minimizing the total number of tag SNPs. Both approaches were achieved by the dynamic programming algorithm (Zhang et al., 2002b).

Most methods for block partitioning and tag SNP selection were developed primarily based on haplotype data. Recently, block-detection methods based on genotype data have been developed (Eskin et al., 2003; Greenspan and Geiger, 2003). In this paper, we discuss extensions of dynamic programming algorithms to genotype data from unrelated individuals as well as genotype data from general pedigrees. The most difficult part in this step is to estimate the haplotype frequency and two haplotypes carried by each individual efficiently. Although many methods have been developed in this area, the haplotype inference and analysis from large pedigrees remain a challenging problem. Furthermore, with the accumulation of different types of data, such as case-control data, sibship data, pedigree data, and pooled DNA genotype data (Sham et al., 2002), new tools for haplotype inference from these combined data need to be developed.

The extent of LD varies among different populations, and thus haplotype block structure also show substantial variation, especially among African and other populations (Gabriel et al., 2002). However, the available methods are all based on the assumption that the population under study is homogeneous. A possible way around this problem is to first use unrelated SNPs to divide a general population into several homogeneous populations (Pritchard and Rosenberg 1999), and then obtain the haplotype block partitions and the tag SNPs for each population. On the other hand, the current HapMap will be constructed based on several major populations and applied to the other unstudied populations. Obviously, this may be problematic and more data are needed to evaluate the validity of such generalizations.

Many methods for haplotype block detection have been developed. The haplotype block structures and the tag SNPs identified in each study depend heavily on the method used. Even using same definition of haplotype blocks and tag SNPs, the

boundary of haplotype blocks and the tag SNPs in each block identified by the dynamic programming algorithm and other methods may not be unique, making it very difficult to compare them. Undoubtedly, this comparison along with the biological relevance of haplotype blocks, such as their relations with recombination hot spot, genetic drift, population history and other factors (Jeffreys et al., 2001; Phillips et al., 2003), are of great interest. Our interest of haplotype block is its potential utility in association studies: to reduce the genotyping effort and preserve high power in mapping disease genes responsible for human complex diseases. Under this criterion, different methods can be compared. Zhang et al. (2002a) compared the power of different tests using tag SNPs and all SNPs by extensive simulations. They found that the power using 20% tag SNPs is only reduced less than 10% under certain conditions. However, they did not extend it to other tag SNP identification methods. Obviously, the assessment of power using tag SNPs conducted from different methods would be of great interest and extremely helpful for association studies.

One of the most important implications of haplotype block is the genotyping effort can be largely reduced without much loss of power using only tag SNPs. Markers within the same block generally show a strong LD pattern. This advantage, along with the relatively smaller number of haplotypes defined by tag SNPs in each block provides a possible way to resolve the complexity of haplotypes. Thus, we can use each block as a unit in association studies (Daly et al., 2001). However, haplotype block boundaries are not unique and substantial LD can be found between loci in different blocks (Gabriel et al., 2002). Therefore, the effectiveness of using each block as a unit to study the association between complex traits and candidate regions needs to be further investigated.

Acknowledgements

This research was partially supported by National Institutes of Health Grant DK53392, National Institutes of Health Grant 1 R01-RR16522-01, National Science Foundation

EIA-0112934, and the University of Southern California. The part on haplotype block partition with genotype data was in collaboration with Peter Qin and Jun S. Liu of Harvard University.

References

- [Becker and Knapp, 2003] Becker T, Knapp M (2003) Efficiency of haplotype frequency estimation when nuclear family information is included. *Hum Hered* 54: 45-53.
- [Clark 1990] Clark AG (1990) Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol Biol Evol* 7: 111-112.
- [Daly et al., 2001] Daly MJ, Rioux JD, Schaffner SF, Hudson TJ, Lander ES (2001) High-resolution haplotype structure in the human genome. *Nat Genet* 29: 229-232.
- [Dawson et al., 2002] Dawson E, Abecasis GR, Bumpstead S, Chen Y, Hunt S, Beare DM, Pabialal J, Dibling T, Tinsley E, Kirby S, Carter D, Papaspyidonos M, Livingstone S, Ganske R, Löhmmussaar E, Zernant J, Tõnisson N, Remm M, Mgi R, Puurand T, Vilo J, Kurg A, Rice K, Deloukas P, Mott R, Metspalu A, Bentley DR, Cardon LR, Dunham I (2002) A first-generation linkage disequilibrium map of human chromosome 22. *Nature* 418: 544-548.
- [Douglas et al., 2001] Douglas JA, Boehnke M, Gillanders E, Trent JM, Gruber SB (2001) Experimentally-derived haplotypes substantially increase the efficiency of linkage disequilibrium studies. *Nat Genet* 28: 361-364.
- [Dunning et al., 2000] Dunning AM, Durocher F, Healey CS, Teare MD, McBride SE, Carlomagno F, Xu CF, Dawson E, Rhodes S, Ueda S., Lai E, Luben RN, Van Rensburg EJ, Mannermaa A, Kataja V, Rennart G, Dunham I, Purvis I, Easton D, Ponder BAJ (2000) The extent of linkage disequilibrium in four

populations with distinct demographic histories. *Am J Hum Genet* 67: 1544-1554.

[Eisenbarth et al., 2001] Eisenbarth I, Striebel AM, Moschgath E, Vodel W, Assum G (2001) Long-range sequence composition mirrors linkage disequilibrium pattern in 1.13 MB region of human chromosome 22. *Hum Mol Genet* 24: 2833-2839.

[Elston and Stewart, 1971] Elston RC, Stewart J (1971) A general model for the genetic analysis of pedigree data. *Hum Hered* 21: 523-542.

[Eskin et al., 2003] Eskin E, Halperin E, Eskin E (2003) Large scale recovery of haplotypes from genotype data using imperfect phylogeny. In Miller W, Vingron M, Sorin I, Pevzner P, Waterman M (eds), *Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB 2003)*. ACM, New York, pp104-113.

[Excoffier et al., 1995] Excoffier L, Slatkin M (1995) Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol Biol Evol* 12: 921-927.

[Gabriel et al., 2002] Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, Higgins J, DeFelice M, Lochner A, Faggart M, Liu-Cordero SN, Rotimi C, Adeyemo A, Cooper R, Ward R, Lander ES, Altshuler D. (2002) The structure of haplotype blocks in the human genome. *Science* 296: 2225-2229.

[Garey and Johnson, 1979] Garey MR, Johnson DS (1979) *Computers and Intractability*. Freeman, New York, p222.

[Greenspan and Geiger 2003] Greenspan G Geiger D (2003) Model-based inference of haplotype block variation. In Miller W, Vingron M, Sorin I, Pevzner P, Waterman M (eds), *Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB 2003)*. ACM, New York, pp131-137.

- [Gusfield, 2001] Gusfield D (2001) Inference of haplotypes from samples of diploid populations: Complexity and algorithms. *J Comp Biol* 8: 305-323.
- [Gusfield et al., 1994] Gusfield D, Balasubramanian K, Naor D (1994) Parametric optimization of sequence alignment. *Algorithmica* 12: 312-326.
- [Hawley and Kidd, 1995] Hawley ME, Kidd KK (1995) HAPLO: a program using the EM algorithm to estimate the frequencies of multi-site haplotypes. *J Hered* 86: 409-411.
- [Jeffreys et al., 2001] Jeffreys AJ, Kauppi L, Neumann R (2001) Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex. *Nat Genet* 29:217-222.
- [Johnson et al., 2001] Johnson GCL, Esposito L, Barratt BJ, Smith AN, Heward J, Genova GD, Ueda H, Cordell HJ, Eaves IA, Dudbridge F, Twells RCJ, Payne F, Hughes W, Nutland S, Stevens H, Carr P, Tuomilehto-Wolf E, Tuomilehto J, Gough SCL, Clayton DG, Todd JA (2001) Haploype tagging for the identification of common disease genes. *Nat Genet* 29: 233-237.
- [Kruglyak,1999] Kruglyak L (1999) Prospects for whole-genome linkage disequilibrium mapping of common disease genes. *Nat Genet* 22: 139-144.
- [Lange and Goradia, 1987] Lange K, Goradia TM (1987) An algorithm for automatic genotype elimination. *Am J Hum Genet* 40: 250-256.
- [Lin et al., 2002] Lin S, Cutler DJ, Zwick ME and Chakravarti A (2002) Haplotype inference in random population samples. *Am J Hum Genet* 71: 1129-1137.
- [Long et al., 1995] Long JC, Williams RC, Urbanek M (1995) An E-M algorithm and testing strategy for mutiple-locus haplotypes. *Am J Hum Genet* 56: 799-810.
- [MichalatosBeloin et al., 1996] MichalatosBeloin S, Tishkoff SA, Bentley KL, Kidd KK, Ruano G (1996) Molecular haplotyping of genetic markers 10 kb apart by allelic-specific long-range PCR. *Nucleic Acids Res* 24: 4841-4843.

- [Niu et al., 2002] Niu T, Qin Z, Xu X, Liu JS (2002) Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Am J Hum Genet* 70: 157-159.
- [Nordborg and Tavaré, 2002] Nordborg M, Tavaré S (2002) Linkage disequilibrium: what history has to tell us. *Trends Genet* 18: 83-90.
- [O'Connell 2000] O'Connell JR (2000) Zero-Recombinant haplotyping: applications to fine mapping using SNPs. *Genet Epidemiol* 19(Suppl 1): S64-S70.
- [O'Connell and Weeks, 1995] O'Connell JR, Weeks DE (1995) The VITESSE algorithm for rapid exact multilocus linkage analysis via genotype set-recoding and fuzzy inheritance. *Nat Genet* 11: 402-408.
- [Patil et al., 2001] Patil N, Berno AJ, Hinds DA, Barrett WA, Doshi JM, Hacker CR, Kautzer CR, Lee DH, Marjoribanks C, McDonough DP, Nguyen BT, Norris MC, Sheehan JB, Shen N, Stern D, Stokowski RP, Thomas DJ, Trulson MO, Vyas KR, Frazer KA, Fodor SPA, Cox DR (2001) Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science* 294: 1719-1723.
- [Phillips et al., 2003] Phillips MS, Lawrence R, Sachidanandam R, Morris AP, Balding DJ, Donaldson MA, Studebaker JF, Ankener WM, Alfisi SV, Kuo FS, Camisa AL, Pazorov V, Scott KE, Carey BJ, Faith J, Katari G, NBhatti HA, Cyr JM, Derohannessian V, Elosua C, Forman AM, Grecco NM, Hock CR, Kuebler JM, Lathrop JA, Mockler MA, Nachtman EP, Restine SL, Varde SA, Hozza MJ, Gelfand CA, Broxholme J, Abecasis GR, Boyce-Jacino MT, Cardon LR. 2003. Chromosome-wide distribution of haplotype blocks and the role of recombination hot spots. *Nat Genet* 33: 382-387.
- [Pritchard and Rosenberg, 1999] Pritchard JK, Rosenberg NA (1999) Use of unlinked genetic markers to detect population stratification in association studies. *Am J Hum Genet* 65: 220-228.

- [Qin et al., 2002] Qin Z, Nu T, Liu J (2002) Partitioning-Ligation-Expectation-Maximization Algorithm for haplotype inference with single-nucleotide Polymorphisms. *Am J Hum Genet* 71: 1242-1247.
- [Reich et al., 2002] Reich DE, Schaffner SF, Daly MJ, McVean G, Mullikin JC, Higgins JM, Richter DJ, Lander ES, Altshuler D (2002) Human genome sequence variation and the influence of gene history, mutation and recombination. *Nat Genet* 32: 135-142.
- [Risch and Merikangas, 1996] Risch, N and K. Merikangas (1996) The future of genetic studies of complex human diseases. *Science* 273: 1516-1517.
- [Schwartz et al., 2003] Schwartz R, Halldórsson, Bafna V, Clark AG, Sorin I (2003) Robustness of inference of haplotype block structure. *Journal of Comput Biol* 10:13-19.
- [Sham et al., 2002] Sham P, Bader JS, Craig I, O'Donovan M, Owen M (2002) DNA pooling: a tool for larger-scale association studies. *Nat Rev Genet* 3: 862-871.
- [Stephens et al., 2001] Stephens M, Smith NJ, Donnelly P (2001) A new statistical method for haplotype reconstruction from population data. *Am J Hum Genet* 68: 978-989.
- [Taillon-Miller et al., 2000] Taillon-Miller P, Bauer-Sardina I, Saccone NL, Putzel J, Laitinen T, Cao A, Kere J, Pilia G, Rice JP, Kwork PY (2000) Juxtaposed regions of extensive and minimal linkage disequilibrium in human Xq25 and Xq28. *Nat Genet* 25: 324-328.
- [Wang et al., 2002] Wang N, Akey JM, Zhang K, Chakraborty K and Jin L (2002) Distribution of recombination crossovers and the origin of haplotype blocks: The interplay of population history, recombination, and mutation. *Am J Hum Genet* 71: 1227-1234.
- [Wang et al., 1998] Wang DG, Fan JB, Siao CJ, Berno A, Young P, Sapolsky R, Ghandour G, Perkins N, Winchester E, Spencer J, Kruglyak L, Stein L, Hsie

L, Topaloglou T, Hubbell E, Robinson E, Mittmann M, Morris MS, Shen NP, Kilburn D, Rioux J, Nusbaum C, Rozen S, Hudson TJ, Lipshutz R, Chee M, Lander ES (1998) Large-scale identification, mapping and genotyping of single-nucleotide polymorphisms in the human genome. *Science* 280: 1077-1082.

[Waterman, 1995] Waterman M.S (1995) Introduction to computational biology: maps, sequences and genomes. Chapman & Hall/CRC Press, Boca Raton FL.

[Waterman et al., 1992] Waterman MS, Eggert M, Lander EL (1992) Parametric sequence comparisons. *Proc Natl Acad Sci USA* 89: 6090-6093.

[Weiss and Clark, 2002] Weiss KM, Clark AG (2002) Linkage disequilibrium and the mapping of complex human traits. *Trends Genet* 18: 19-24.

[Wijsman 1987] Wijsman EM (1987) A deductive method of haplotype analysis in pedigrees. *Am J Hum Genet* 51:356-373.

[Zhang et al., 2002a] Zhang K, Calabrese P, Nordborg M, Sun F (2002a) Haplotype structure and its applications to association studies: power and study design. *Am J Hum Genet* 71: 1386-1394.

[Zhang et al., 2002b] Zhang K, Deng M, Chen T, Waterman MS, Sun F (2002b) A dynamic programming algorithm for haplotype block partitioning. *Proc Natl Acad Sci USA* 95: 7335-7339.

[Zhang et al., 2003] Zhang K, Sun F, Waterman MS, Chen T (2003) Dynamic programming algorithms for haplotype block partitioning: applications to human chromosome 21 haplotype data. In Miller W, Vingron M, Sorin I, Pevzner P, Waterman M(eds), *Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB 2003)*. ACM, New York, pp332-340. In press.