

Haplotype block partitioning and tagSNP selection under the perfect phylogeny model

Changiz Eslahchi^{1*}, Ali Katanforoush¹, Hamid Pezeshk^{2,3}, Narjes Afzaly²

¹Faculty of Mathematical Sciences, Shahid Beheshti University, G.C., P.O. Box 198396-3113, Tehran, I.R. Iran

²School of Mathematics, Statistics and Computer Sciences, Center of Excellence in Biomathematics, College of Science, University of Tehran, P.O. Box 6455-14155, Tehran, I.R. Iran ³School of Computer Science, Institute for Research in Fundamental Sciences, P.O. Box 19395-5746, Tehran, I.R. Iran

Abstract

Single Nucleotide Polymorphisms (SNPs) are the most usual form of polymorphism in human genome. Analyses of genetic variations have revealed that individual genomes share common SNP-haplotypes. The particular pattern of these common variations forms a block-like structure on human genome. In this work, we develop a new method based on the Perfect Phylogeny Model to identify haplotype blocks using samples of individual genomes. We introduce a rigorous definition of the quality of the partitioning of haplotypes into blocks and devise a greedy algorithm for finding the proper partitioning in case of perfect and semi-perfect phylogeny. It is shown that the minimum number of tagSNPs in a haplotype block of Perfect Phylogeny can be obtained by a polynomial time algorithm. We compare the performance of our algorithm on haplotype data of human chromosome 21 with other previously developed methods through simulations. The results demonstrate that our algorithm outperforms the conventional implementation of the Four Gamete Test approach which is the only available method for haplotype block partitioning based on Perfect Phylogeny.

Keywords: Single Nucleotide Polymorphisms; haplotype; tagSNP; perfect Phylogeny

INTRODUCTION

One of the major interests of current researches in

genomics is to understanding the genomic differences in human population so as to be able to find out what makes us different rather than what we have in common. Single Nucleotide Polymorphism (SNP), i.e. single base pair difference between individuals in a population, is believed to be an important reason for variations that occur in human genome. SNP is the result of a substantiated single site mutation in population. Genome of a new child is affected by many single site mutations of which some spread over population. Currently more than 14.5 million SNPs have been reported to and validated by dbSNP which is almost 0.5% of whole genome nucleotides (www.ncbi.nlm.nih.gov/SNP/snp_summary.cgi). An SNP-haplotype is a sequence of SNP alleles in a certain region of chromosome. For simplicity, we will use haplotype rather SNP-haplotype from now onwards. Soon after the completion of Human Genome Project, researchers have shown that genome comprises regions of certain boundaries in which haplotypes are inherited through generations without any change.

The studies suggest that human genome can be viewed as a partitioning of haplotype blocks in which common variations of haplotypes within a certain population are distinguished by a relatively small number of SNPs, called haplotype tagging SNPs (tagSNPs), (Dawson *et al.*, 2002; Johnson *et al.*, 2001).

Different models have been introduced to define the block-like structure of genome. Haplotype blocks in methods of Patil *et al.* (2001) and Zhang *et al.* (2002) are subjected to get a limited Haplotype Diversity. The former approach applies a greedy algo-

*Correspondence to: Changiz Eslahchi, Ph.D.
Tel: +98 98 21 29903015; Fax: +98 21 22431652
E-mail: ch-eslahchi@sbu.ac.ir

Archive of SID

rithm to find edges of the haplotype blocks and the latter performs a dynamic programming to achieve minimum number of tagSNPs. Methods based on statistical association are essentially derived from the measures of Linkage Disequilibrium (LD) (Daly *et al.*, 2001), (Gabriel *et al.*, 2002), (Katanforoush *et al.*, 2009). The method introduced by Gabriel *et al.* (2002) is the most commonly used in this category. Since there are some parts of haplotype in which no strongly associated SNPs exist, Gabriel's method results in obtaining "islands" of haplotype blocks separated by some uncertain gaps.

In this paper we present an algorithm for haplotype block partitioning based on Perfect Phylogeny. To consider how the actual data are fitted to the model, a measure for deviation is introduced. For a given constant σ , ($0 < \sigma \leq 1$) denoting the amount of deviation, we refine haplotype samples by changing at most $(1 - \sigma) \times 100\%$ of SNPs alleles. By setting $\sigma = 1$, our method finds the partitioning in which no recombination within each block can be observed. This case corresponds to pure perfect haplotypes. In a partially perfect phylogeny, by setting $\sigma < 1$, at most $(1 - \sigma) \times 100\%$ of SNPs has to be ignored so that the rest satisfy the Perfect Phylogeny constraints. The conventional approach to find haplotype blocks under the model of Perfect Phylogeny is the Four Gamete Test approach (Hudson and Kaplan, 1985). In this approach, a pair of SNPs passes the test if no recombination occurred between them. The first and only practical implementation of this approach is attributed to Wang *et al.* (2002) in which haplotype blocks starting with a single SNP extended until δ percent of SNP pairs fail the Four Gamete Test. Gramm *et al.* (2009) have shown that problem of haplotype partitioning with minimum haplotype blocks under the Perfect Phylogeny model is NP-hard. Our approach obtains an optimal partitioning for the problem assuming that blocks are continuous regions and are governed by the model of Perfect Phylogeny.

It is also shown that if there is no missing data in a perfect block, then the number of tagSNPs is equal to the number of mutually distinct haplotypes minus one and in this case, tagSNPs can be identified using a polynomial time algorithm. For blocks with missing data, we use an approximation algorithm to find minimum set of tagSNPs.

In what follows, we first introduce methods which consist of missing SNPs reduction, measurement of block perfectness, perfect block partitioning and tagSNP selection. We then assess the accuracy of the

block inference and discuss the results of applying the method on a set of real haplotype samples of human genome.

METHODS

The outline of the method is as follows. Samples of individual haplotypes are given. We assume that all the haplotypes are taken from a certain region of human genome. Each of the next procedures is performed on various sub-intervals of this region, which finally some of these sub-intervals will be assigned as haplotype blocks. Starting from a singleton interval, we extend the interval by adding one SNP to the right boundary of the interval and then re-evaluate the same procedures on this interval. An interval is considered as a block if certain criteria on Perfect Phylogeny are satisfied. The procedures include the calculation of pattern matrices, missing data reduction, perfectness measurement, and tagSNP selection. Following this section, we will provide details of the procedures.

SNPs are usually bi-allelic, i.e. every SNP variation is on two nucleotide forms. Thus haplotypes can be represented by 0/1 vectors. From now on, we suppose that haplotype samples are given by a matrix in which columns correspond to SNPs and rows correspond to individuals. We denote this matrix with A . We assume that the major and minor alleles are represented by 0 and 1, respectively and the letter N indicates missing entries.

Let c_1 and c_2 be two columns of A . Let $V(c_1, c_2)$ be the set of different combinations that the pair of columns takes on over all rows of A , so $V(c_1, c_2) = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$. It has been shown (Estabrook *et al.*, 1975) that a necessary and sufficient condition for A to be a Perfect Phylogeny is that for every pair c_1 and c_2 of columns of A , $|V(c_1, c_2)| \leq 3$. This criterion is known as Four Gamete Test. We assume that the number of 1-entries in each column that is also called Minor Allele Frequency (MAF) is less than the half. As a consequence, in every pair of columns of A , gamete (0,0) exists so we can consider only three gametes $\{(0, 1), (1, 0), (1, 1)\}$ for the Four Gamete Test, so we state the test by $|V(c_1, c_2) - \{(0, 0)\}| \leq 2$.

Reducing Missing Data using Pattern Matrix: A haplotype h' of length n is said to be a *cover of type (1)* of another haplotype h (of the same length as h') if for each SNP i , $1 \leq i \leq n$, which $h(i) \neq N$ then $h'(i) = h(i)$. We denote this by $h \ll_{(1)} h'$. The haplotype h' is called the *cover of type (2)*, if for another haplotype h for which

Archive of SID

$h(i) \neq N$, then $h'(i) = h(i)$ or $h'(i) = N$. It is denoted by $h \ll_{(2)} h'$.

A haplotype matrix $P_{m' \times n}$ is said to be a *pattern matrix* for another haplotype matrix $A_{m \times n}$ ($m' \leq m$), if it satisfies following conditions,

- (1) $\forall h \in A, \exists h' \in P \ni h \ll_{(1)} h'$
- (2) $\forall h' \in P, \exists h \in A \ni h \ll_{(1)} h'$
- (3) $\forall h \in P, \exists h' \in P \ni h \ll_{(2)} h'$

Theorem 1. Suppose that P is a pattern matrix for A . If P is perfect then A is perfect, too.

Proof. Suppose on the contrary, A is not perfect. So there exist three rows, say, r_1, r_2 and r_3 and two columns, say i and j , such that

- (1) $a_{r_1 i} = 1$ and $a_{r_1 j} = 0$,
- (2) $a_{r_2 i} = 1$ and $a_{r_2 j} = 1$,
- (3) $a_{r_3 i} = 0$ and $a_{r_3 j} = 1$.

Suppose $r_k \ll_{(1)} h_{l_k}$, in which h_{l_k} is the l_k -th row of P . Therefore,

- (1') $P_{l_1 i} = 1$ and $P_{l_1 j} = 0$,
- (2') $P_{l_2 i} = 1$ and $P_{l_2 j} = 1$,
- (3') $P_{l_3 i} = 0$ and $P_{l_3 j} = 1$.

This is a contradiction because we have assumed that

P was perfect.

Pattern Matrix: We take the following steps in constructing a pattern matrix;

- (1) Sort haplotypes of A into an increasing order depending on the number of missing SNPs, say A_1, A_2, \dots, A_n .
- (2) At the beginning, let A_1 be the first row of P . In the k -th step, $k \leq 2$, compare A_k with all the existing rows of P . If none of the rows of P covers A_k of type (2) then add A_k to P as a new row. Otherwise, for each row p_i of P for which $A_k \ll_{(2)} p_i$, take $p_i(j) = A_k(j)$ for each j that $1 \leq j \leq n$ and $p_i(j) = N$.

Once the above procedure is completed, each row of A is covered, of type (1) by some rows of P . From now on, when we use the term "cover" we mean cover of type (1).

In what follows, we show that the matrix P is a pattern matrix for A .

1. To prove the condition (1) of pattern matrix definition, let us assume that we arrived at the k -th step. We consider two cases:

Case (I): If there is no $p \in P$ such that $A_m \ll_{(2)} p$ then we add A_k to P as a new row. The 0-1 entries of A_k will never change through the next steps. Therefore this row ultimately will cover A_k .

Case (II): If there exists $p \in P$ such that $A_m \ll_{(2)} p$ then

Table 1. Example of haplotype matrix and the covering pattern matrix.

Pattern id	Pattern haplotaype	Haplotype matrix	Covering pattern
1	111111011111011101111111111111	0000000000000000100NNNNNNNNNN	3
2	0000001000001000100000000000	001101011010000101110N00N01NN	5
3	0000000000000000100000000000	0011N1011010000101110000000NN	4
4	0011N1011010000101110000000NN	0011010010100001011NNNNNNNNNN	8
5	001101011010000101110N00N01NN	0000000000000000100NNNNNNNNNN	3
6	N0N101011N10010101110N00N01NN	00000000000000001000000000NN	3
7	0011110111110101011NNNNNNNNNN	0011110111110101011NNNNNNNNNN	7
8	0011010010100001011NNNNNNNNNN	0000000000000000N00NNNNNNNNNN	3
9	00N000001010000101NNNNNNNNNN	00N000001010000101NNNNNNNNNN	9
		1111110111110111011NNNNNNNNNN	1
		00000000000000001000000N0000	3
		1111N101111101110111111111111	1
		0000000000000000100000000000	3
		0000001000001000100000000000	2
		0000001000001000100000000000	2
		N0N101011N10010101110N00N01NN	6
		111111011111011101111111111111	1
		0000000000000000NN010000000000	3

Archive of SID

the row corresponding to P will ultimately cover A_k . So we conclude that the condition (1) of pattern matrix holds.

2. Since each row of P is obtained from a row in A , so the condition (2) holds.

3. In every step that we intend to add a new row, say q , to P we check that if for an arbitrary row q' of P there exists at least a column j for which $q(j) \neq q'(j)$ and none of these entries are missing that yields none of rows of P covers each other.

Table 1 presents an example of haplotype matrix and its associated pattern matrix. As shown in Table 1, missing data could be inferred. Based on Theorem 1, many of missing SNPs, N, are replaced by 0 or 1. This is done so that the condition of perfectness holds.

Measure of "Perfectness": The conventional method to measure the deviation from "perfectness" considers the ratio of Four Gamete Test failures to all pairs of columns of a haplotype matrix (Wang *et al.*, 2002). For example, in Figure 1, the deviation from "perfectness" is 0.2. Instead, we consider 1-entries while changing them to 0, a test failure is resolved. We then obtain a minimal set of such entries that addresses all failures. We use the size of the minimal set divided by the number of all entries as our measure of deviation from "perfectness". As an example, it is 1/18 in Figure 1. It should be noted that changing all entries of the set does not necessarily transform the matrix to a perfect matrix and in fact, we are just interested in the least necessary number of such changes.

In detail, for every column i and j ($i < j$) of a pattern matrix P , we define three conflict sets, C_{ij}^1 , C_{ij}^2 and

C_{ij}^3 as follows;

$$\begin{aligned} C_{ij}^1 &= \{r \mid p_{ri} = p_{rj} = 1\}, \\ C_{ij}^2 &= \{r \mid p_{ri} = 1 \wedge p_{rj} = 0\}, \\ C_{ij}^3 &= \{r \mid p_{ri} = 0 \wedge p_{rj} = 1\}. \end{aligned}$$

Now, the problem is to find the minimal set of coordinates, $S = \{(r, s) \mid p_{rs} = 1\}$ such that for every pair (i, j) which has conflict, one of the following conditions is satisfied;

- (1) if $r \in C_{ij}^1 \Rightarrow (r, i) \in S$ or $(r, j) \in S$,
- (2) if $r \in C_{ij}^2 \Rightarrow (r, i) \in S$,
- (3) if $r \in C_{ij}^3 \Rightarrow (r, j) \in S$.

Therefore, for columns i and j which fail in Four Gamete Test, there exist 1-entries in S such that changing them to 0 resolves the failure. In what follows, a heuristic approach is introduced which approximately finds the minimal set S .

Algorithm: Let \sim be a total order on the set of column pairs, $\Sigma = \{(i, j) \mid 1 \leq i < j \leq n\}$. Based on the ordering of Σ and in each step, the algorithm considers a pair of columns of the pattern matrix and adds coordinates of some of the 1-entries to the solution set. The subset of 1-entries in each step is selected among subsets of 1-entries which resolve the test failure that possibly occurs in the pair of columns. We select the subset of 1-entries for each column pair by using a greedy algorithm as follows;

Suppose that (i_0, j_0) is the minimum element of (Σ, \sim) . At the first step of the algorithm, let $S_0^1 = \{(r, i_0) \mid r \in C^1_{i_0 j_0}\}$, $S_0^2 = \{(r, i_0) \mid r \in C^2_{i_0 j_0}\}$, and $S_0^3 = \{(r, j_0) \mid r \in C^3_{i_0 j_0}\}$.

At the k -th step of the algorithm, define

$$S^1_k = \min_{1 \leq l \leq 3} (S^l_{k-1} \cup \{(r, i_k) \mid r \in C^l_{i_k j_k} \text{ and } (r, j_k) \notin S^l_{k-1}\}),$$

$$S^2_k = \min_{1 \leq l \leq 3} (S^l_{k-1} \cup \{(r, i_k) \mid r \in C^2_{i_k j_k}\}),$$

$$S^3_k = \min_{1 \leq l \leq 3} (S^l_{k-1} \cup \{(r, j_k) \mid r \in C^3_{i_k j_k}\}),$$

in which $\min(\cdot)$ denotes the set with minimum number $1 \leq l \leq 3$

of elements. After the final step, the solution set S is obtained by $S = \min_{1 \leq l \leq 3} (S^l_K)$, where $K = |\Sigma|$.

$$1 \leq l \leq 3$$

Our measure of deviation from "perfectness" is now obtained by $\delta(A) = |S| / (mn)$. Obviously, the result of the algorithm depends on the total order of Σ . Thus,

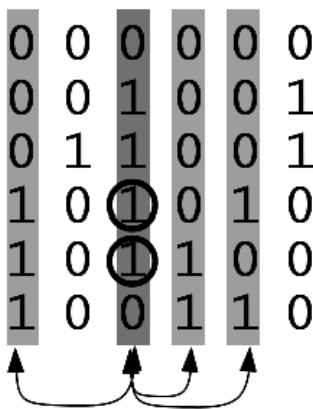


Figure 1. Measuring the deviation from "perfectness". Three pairs of columns (shaded gray) fail in Four Gamete Test that is counted 3 failures out of 15. As another approach, we consider two entries (in circles) which changing them from 1 to 0 turn whole matrix into a perfect matrix.

Archive of SID

the algorithm should be performed several times using different total orders to find the size of the optimal set S . In practice, we examined few total orders among those in which $(i, j) \sim (i', j')$ implies that $j < j'$. The following algorithm obtains haplotype blocks of a chromosomal region of n SNPs and m haplotype samples in $O(n^2m)$.

Block partitioning: In this section we present an algorithm for partitioning a haplotype matrix into blocks. Given a set of m haplotypes on n SNPs by an $m \times n$ matrix A , we find an increasing sequence $\langle \xi_1=0, \xi_2, \dots, \xi_k=n \rangle$ indicating boundaries of haplotype blocks in such a way that $\delta(P_i) \leq 1-\sigma$, for $i=1, \dots, k-1$, where P_i is the pattern matrix of the submatrix starting from column ξ_{i+1} to column ξ_{i+1} and σ is an arbitrary threshold which denotes how much a certain block is near to perfect. We also add another option to the algorithm by which at least $100\alpha\%$ of individuals share “common haplotypes” in each block. In detail, the percentage of “common haplotypes” in a block is defined as $1-t/m$, where t is the number of patterns which cover exactly one haplotype.

For a given haplotype matrix, our algorithm considers the first column as the current block. In each iteration, the pattern matrix of the current block is obtained, and then the percentage of common haplotypes and the measure of “perfectness” are calculated. The algorithm extends the current block to the next right column until at least one of conditions on perfectness or “common haplotypes” fails. The same procedure iterates on the rest of columns.

Selection of tagSNPs: Let B be a block of haplotypes with n SNPs. We call $S \subseteq \{1, \dots, n\}$ a set of tagSNPs if for each two different haplotypes i and j in B , there exists an element t of S such that $B_{it} \neq B_{jt}$, $B_{it} \neq N$ and $B_{jt} \neq N$.

Theorem 2: Let B be a set of perfect haplotypes with no missing data. If $T(B)$ is the set with the minimum tagSNPs then $|T(B)|=m-1$ and $T(B)$ could be found by a polynomial time algorithm of order $O(mn)$.

Proof: We apply strong induction on the size of B . The result is obvious for the case where $|B|=2$. Assume that the induction hypothesis holds for every set A of size of at most $m-1$. Also assume that the set $B=\{h_1, h_2, \dots, h_m\}$ satisfies the condition of the theorem. Then there exists a SNP, x , for which none of sets $X_0=\{h \in B|h(x)=0\}$ and $X_1=\{h \in B|h(x)=1\}$ are empty;

this SNP is found in $O(n)$. Therefore $|X_0| \leq m-1$ and $|X_1| \leq m-1$. So by induction assumption, $|T(X_0)|=|X_0|-1$ and $|T(X_1)|=|X_1|-1$. These sets can be found by an algorithm of $O(n(|X_0|+|X_1|))$. It is obvious that $x \notin T(X_0) \cup T(X_1)$. Now we show that $T(X_0) \cap T(X_1) = \emptyset$. In contrary, let us assume $y \in T(X_0) \cap T(X_1)$ then there exist four haplotypes $h_i, h_j \in X_0$ and $h_k, h_l \in X_1$ such that,

$$\begin{aligned} h_i(x)=0, & \quad h_i(y)=1, \\ h_j(x)=0, & \quad h_j(y)=0, \\ h_k(x)=1, & \quad h_k(y)=1, \\ h_l(x)=1, & \quad h_l(y)=0. \end{aligned}$$

Hence, four gametes 00, 01, 10, and 11 could be observed at SNPs x and y that is a contradiction because we have already assumed that B is perfect. Therefore, $T=T(X_0) \cup T(X_1) \cup \{x\}$ is a set of tagSNPs of size $m-1$. It means that $|T(B)| \leq m-1$. Now suppose that $z \in T(B)$. Similar to the above argument we define two sets Z_0 and Z_1 to obtain $T(B)=T(Z_0) \cup T(Z_1) \cup \{z\}$. So $|T(B)|=|T(Z_0)|+|T(Z_1)|+1$ that is equal to $m-1$ by the induction hypothesis.

Based on the proof of Theorem 2, we introduced an algorithm to select a set of tagSNPs for blocks of haplotype samples. For a general haplotype block, the problem of minimum tagSNP set is NP-hard (Huang and Chao, 2008; Vinterbo *et al.*, 2006). However, for blocks of perfect haplotypes, the following algorithm finds a set of tagSNPs with minimum size in polynomial time. The result is also a reasonable approximation of minimum tagSNPs for general cases.

Assume that $T(B)=\min\{T|T \text{ is a set of tagSNPs for } B\}$. Recall that in Theorem 2, we proved that for each SNP x if $X_0=\{h \in B|h(x)=0\} \neq \emptyset$ and $X_1=\{h \in B|h(x)=1\} \neq \emptyset$ then there exists a minimal set of tagSNPs for which $x \in T(B)$. We use this proposition to derive an algorithm for tagSNP selection.

For each SNP i , define $A_i=\{p \in P|p(i)=0\}$, $B_i=\{p \in P|p(i)=1\}$, and let $m_i=\min(|A_i|, |B_i|)$ and $M_i=\max(|A_i|, |B_i|)$. Choose one SNP of maximum M_i among those maximizing m_i , say i_1 , and let $T=\{i_1\}$. Assume that after the k -th step of algorithm, we have $T=\{i_1, i_2, \dots, i_k\}$, and $D_k=\{(x_1, \dots, x_k) | Q_{x_1, \dots, x_k} \neq \emptyset \text{ for } 1 \leq j \leq k \text{ and } x_j \in \{0, 1\} \text{ for } 1 \leq i \leq k\}$, where $Q_{x_1, \dots, x_k} = \{p \in P|p(i_r) \in \{x_r, N\} \text{ for } 1 \leq r \leq k\}$.

For each $(x_1, \dots, x_k) \in D_k$ and each $i \in \{1, 2, \dots, n\} - T$, we define $A_{x_1, \dots, x_k; i} = \{p | p \in Q_{x_1, \dots, x_k} \text{ and } p(i)=0\}$ and $B_{x_1, \dots, x_k; i} = \{p | p \in Q_{x_1, \dots, x_k} \text{ and } p(i)=1\}$.

Let

$$m_{ik} = \sum_{(x_1, \dots, x_k) \in D_k} \min(|A_{x_1, \dots, x_k; i}|, |B_{x_1, \dots, x_k; i}|), \text{ and}$$

Archive of SID

$$M_{ik} = \sum_{(X_1, \dots, X_k) \in D_k} \max(|A_{x_1, \dots, x_k; i}|, |B_{x_1, \dots, x_k; i}|),$$

Similarly for the next step, we choose one SNP of maximum M_{ik} among those maximizing m_{ik} , say i_{k+1} , and add it to T . The algorithm stops at step r if for each $(x_1, \dots, x_r) \in D_r$, we have $|Q_{x_1, \dots, x_k}| \leq 1$.

Method evaluation: In this section, we assess our proposed methods through evaluations on real and simulated haplotype data. In the first assessment, we estimate the accuracy of methods to detect perfect and partially perfect blocks using simulation generated samples. By the other evaluations, we compare general features of haplotype blocks of two previously reported block structures of human chromosome 21 with results obtained by our algorithms. In the assessment, length of blocks and its distribution on genome, coverage of “common haplotypes” and number of tagSNPs are compared among different partitioning for chromosome 21.

Accuracy of block inference: To assess how much a haplotype partitioning method accurately detects boundaries of perfect haplotype blocks, we evaluated our proposed method and Wang’s approach of Four Gamete Test through simulations. At first, we developed a simple algorithm to produce a library of randomly generated haplotype samples under the model of perfect phylogeny and with various sample size. Each entry set of the library contained a set of perfect haplotypes on three to 25 SNPs. The sets comprised haplotype blocks of the final dataset. Precisely, we randomly chose haplotype sets and joined them to each other until samples with at least 200 SNPs were obtained. We rearranged the order of SNPs in each block such that at least a failure on Four Gamete Test among pairs of two neighboring columns of two different blocks was assured. We recorded coordinates of the block boundaries for further assessment. We also added noise to the sample by changing 0 entries to 1 with a noise ratio 0.02.

We considered the following correlation coefficient of partitioning as the measure of accuracy,

$$R^2_{\text{partitioning}} = \frac{\sum w_i^2}{\sum u_i^2 + \sum v_i^2 - \sum w_i^2}$$

where u_j , v_k and w_i are lengths of the simulation generated blocks, the inferred blocks and the length of overlaps between partitions, respectively. In Figure 2,

results of evaluation of Wang’s method (Four Gamete Test), and our proposed method for block partitioning (PerfectBlock) are shown. Each method was performed on simulated samples, before and after introducing noise. Independently, each method was also performed with two different settings for “perfectness” threshold.

It is shown that when haplotype samples are purely governed under the perfect phylogeny model, the inference of PerfectBlock with $\sigma=1$ is persistently exact. This denotes that the proposed measure fits the model as it should be expected. In contrast, it seems that Wang’s method lacks total compatibility with the model, at least in its available implementation in Haploview (Barrett *et al.*, 2005). The noise added to the samples makes haplotype blocks to lose features of perfect phylogeny. As shown by Figure 2, in the presence of noise, the accuracy of our method is higher than Wang’s application of Four Gamete Test. It is noticeable that in this case, accuracy of both methods drops when the parameter of pure perfect model is used.

Evaluation on chromosome 21 haplotypes: We

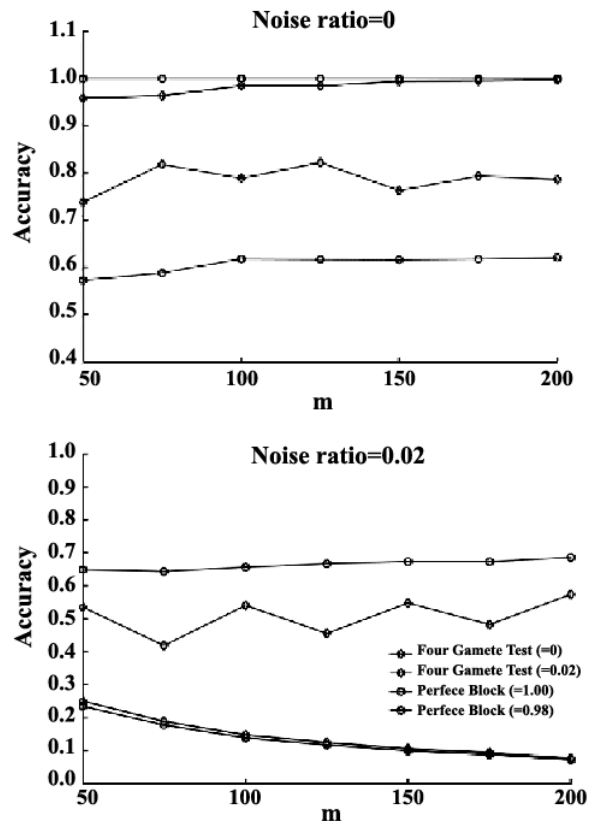


Figure 2. Accuracy of perfect block inference versus number of haplotype samples.

Table 2. Haplotype blocks of human chromosome 21 obtained by different methods.

Method	SNPs/ block	No. of blocks	Ave. block size, (SNPs)	Coverage of common hap.	Block distribution	tagSNPs
$\sigma = 1.00$	>10	657	17.8	0.91	0.17	11217
	3-10	1964	5.4	0.95	0.53	
	<3	1116	1.5	0.99	0.30	
	Total	3737	6.4	0.95	1.00	
$\sigma = 0.98$	>10	744	24.2	0.76	0.40	7873
	3-10	1019	5.8	0.87	0.55	
	<3	97	2.0	0.93	0.05	
	Total	1860	12.9	0.83	1.00	
Patil <i>et al.</i> (2001)	>10	589	23.2	n/a	0.14	4563
	3-10	408	5.2	n/a	0.34	
	<3	2138	1.4	n/a	0.51	
	Total	4135	5.8	n/a	1.00	
Zhang <i>et al.</i> (2002)	>10	742	24.5	n/a	0.29	3582
	3-10	909	5.2	n/a	0.35	
	<3	924	1.3	n/a	0.36	
	Total	2575	9.3	n/a	1.00	

The results of block partitioning of Patil *et al.* (2001), based on a greedy algorithm, and those of Zhang *et al.* (2002), based on dynamic programming are given in Table 2. These results are based on the criteria that 80% of haplotypes within each block are common.

applied our algorithm to haplotype data of human chromosome 21 from Patil *et al.* (2001). The data set included 20 haplotypes of 24, 407 SNPs of minor allele frequency at least 0.1, spanning over 32.4 MB that were located in four contigs.

Table 2 represents the properties of haplotype blocks for two different settings for parameters; $(\alpha, \sigma) = (0, 1.00)$ and $(\alpha, \sigma) = (0, 0.98)$. The first setting implies that no recombination may occur within each block (i.e. pure perfect) and there is no bound on haplotype diversity. By the second setting, we assumed that at most 2% of SNPs within each block are allowed to change so that a perfect tree could be obtained (i.e. partially perfect) and again no bounds on haplotype diversity impose.

As shown by Table 2, total number of blocks considerably reduces from 3,737 to 1,860 when the parameters are changed to $(\alpha, \sigma) = (0, 0.98)$. Also, total number of tagSNPs capturing all haplotype information in all chromosomes reduced from 11,217 to 7,837. In pure perfect analysis, $(\alpha, \sigma) = (0, 1.00)$, 657 blocks containing more than 10 SNPs per block accounted for 17.6% of all blocks, 48.6% of common SNPs and 40.8% of all nucleotides. The largest purely perfect block contained 95 SNPs.

In partial perfect analysis, $(\alpha, \sigma) = (0, 0.98)$, the

number of blocks containing more than 10 SNPs increased to 744 which covered 74.8% of common SNPs and 68.6% of all nucleotides. In this case, the largest block contained 165 SNPs. Generally, purely perfect haplotype blocks were smaller than those restricted by weaker constraint, as it was expected.

Comparison of the greedy algorithm with the pure perfect analysis showed that the number of 4135 blocks obtained by the greedy algorithm reduced to 3737 blocks and also the number of blocks containing more than 10 SNPs increased from 589 to 657. The average number of SNPs reduced from 23.2 to 17.8. The number of blocks containing less than 3 SNPs reduced from 2138 to 1116 without significant changes in average block size. Comparison of dynamic programming algorithm and pure perfect analysis showed an increase in the number of blocks from 2575 to 3737 and also a decrease in the number of blocks containing more than 10 SNPs. Also it is observed that the number of blocks containing less than 3 SNPs was increased. It is notable that block partitions in pure perfect analysis covered all 20 chromosomes in the data set but the results of two others were based on 80% coverage.

Comparison of the results obtained by pure perfect analysis with the dynamic programming algorithm

Archive of SID

with 80% coverage shows an increase of only 4% in the number of blocks. The results of block partitioning using $(\alpha, \sigma)=(0, 0.98)$ and the other methods indicated a considerable decrease in the number of blocks up to 55% compared to greedy algorithm and a decrease of 27% compared to the dynamic programming algorithm. Also the number of blocks containing more than 10 SNPs increased and the number of blocks containing less than 3 SNPs decreased significantly.

It is notable that $\sigma=0.98$ indicates that for a block with size of , e.g. 5 SNPs at most 2 SNPs are allowed to change so as to be able to have a perfect block whereas the results obtained by 80% coverage indicate that 4 haplotypes which contain 20 SNPs are lost. We next examined haplotype diversity within blocks. It should be noted that although our block definition, unlike diversity based definitions, was based on recombination but we used diversity as an additional factor.

Nevertheless, in pure perfect analysis with no recombination and no bound on haplotype diversity within blocks $(\alpha, \sigma)=(0, 1.00)$, we observed that more than 90% of haplotypes within each block were common haplotypes and they appeared more than once. The average number of distinct common haplotypes in each block was less than four. In partial perfect analysis with $(\alpha, \sigma)=(0, 0.98)$ more than 80% of haplotypes in each block were common and the average number of distinct common haplotype was less than five. Thus, the low haplotype diversity is an important feature of regions with low rates of historical recombination. This low haplotype diversity makes it possible to capture all haplotype information in all chromosomes using a small number of Tag SNPs.

Missing data: Missing data are common in haplotype data sets. The missing SNPs will cause ambiguities in haplotypes and affect block partitioning. The number

of missing data in the haplotype matrix of chromosome 21 is 97513 which are 18.5% of data. As explained in methods, using pattern matrix, our algorithm infers many of the missing SNPs in each block. After block partitioning by using our algorithm with parameter values $(\alpha, \sigma)=(0, 0.98)$ the number of missing data was reduced to 18641 which were 3.5% of whole data set. It is considerable that by parameter values $(\alpha, \sigma)=(0, 1.00)$ the number of missing data was reduced to 6451 or 1.2% of whole data set, which means that we inferred almost all of the missing data.

TagSNP Selection: One of our main concerns in this work was defining a subset of haplotype Tag SNPs that characterized the haplotype diversity of a data set. These tags can be used to discriminate haplotypes within the same block. After partitioning SNP data into blocks, Tag SNPs should be selected for each block. It should be noted that our method for tagSNP selection finds tagSNPs that discriminates all the haplotypes of a block whereas the tagSNP selection which is applied by Patil *et al.* (2001) and Zhang *et al.* (2002) discriminates only a partial subset of haplotypes (i.e. common haplotypes). On the other hand, they used an exhaustive search for tagSNP selection while we applied a greedy algorithm that in blocks with missing data or partial perfect it may overestimate the number of tagSNPs. Therefore, the number of tagSNPs obtained by our algorithm is more than those previously reported (the last column, Table 2).

However, in pure perfect blocks 46% of SNPs are required to capture the information of the all SNPs in data set. In partial perfect blocks with $(\alpha, \sigma)=(0, 0.98)$, the number of Tag SNPs were reduced to 32%.

Block dissimilarity: Aspects of the result sensitivity to the algorithm parameters can be figured out by the block dissimilarities which are shown in Table 3.

Table 3. Dissimilarity between haplotype block structures defined by various parameters.

(α, σ)	(0.8, 1)	(0.9, 1)	(0, 0.98)	(0.8, 0.98)	(0.9, 0.98)	(0.8, 0.8)	(0.9, 0.8)	(1, 0)	Patil <i>et al.</i>
(0, 1)	2438	14769	66609	43778	34074	58669	36911	50847	67911
(0.8, 1)		13277	67645	24606	32710	58248	35479	48303	67145
(0.9, 1)			80869	50411	22356	68552	25629	36040	69969
(0, 0.98)				35546	65810	52241	67952	132494	94620
(0.8, 0.98)					39033	29790	42163	84687	64762
(0.9, 0.98)						54007	7506	45942	61786
(0.8, 0.8)							49646	111429	63903
(0.9, 0.8)								48756	59705
(1, 0)									98399

Archive of SID

Assume that $A=\{a_1, a_2, \dots, a_n\}$ and $B=\{b_1, b_2, \dots, b_m\}$ are two block partitions, where a_i and b_j denote intervals on the chromosome. We defined the distance between A and B by

$$d(A, B) = \sum_{a_i \cap b_j \neq \emptyset} \left| a_i \cup b_j - a_i \cap b_j \right|$$

Using our algorithm with parameters $(\alpha, \sigma)=(0, 0.98)$, we almost obtained the results of greedy algorithm of Patil *et al.* (2001). If we take $(\alpha, \sigma)=(0, 1.00)$ (i.e. only the common haplotype matters) the nearest block partitioning to this partition is obtained when the parameter values are $(\alpha, \sigma)=(0.9, 1.00)$. It is noticeable that if $\alpha=0.90$ and σ varies between 0.8 up to 0.9 we obtained almost the same block partitioning. This is the case for $\sigma=0.8$. It turns out from Table 2 and Table 3 that when we ignored the condition on common haplotypes, a small change in the condition on perfectness resulted in big change in block partitioning. On the other hand, for $0 \leq \alpha \leq 0.8$ and $\sigma=1$ we obtained almost the same block partitioning. This implies that the condition on perfectness, forces to obtain a high percentage of common haplotypes.

CONCLUSIONS

In this paper, we introduced a polynomial time algorithm to partition genome into haplotype blocks under the perfect phylogeny model. We extended the application of Four Gamete Test, a previously known measure of haplotype perfectness and introduced a new measure of perfectness which was shown more accurate in recognition of perfect blocks. In presence of noise, our method was able to find semi-perfect blocks with an acceptable accuracy.

We applied our algorithms on a set of real samples of human chromosome 21. It was shown that haplotype diversity within the perfect blocks is close to 0.8 coverage of common haplotype which was the presumed threshold to block identification in other methods.

Acknowledgments

This work is supported in part by a grant from Iranian National Science Foundation (INSF-844405).

References

Barrett JC, Fry B, Maller J, Daly MJ (2005). Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*.

- 21: 263-265.
- 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 E, Abecasis GR, Bumpstead S, Chen Y, Hunt S, Beare DM, Pabial J, Dibling T, Tinsley E, Kirby S (2002). First-generation linkage disequilibrium map of human chromosome 22. *Nature* 418: 544-548.
- Estabrook G, Johnson C, McMorris F (1975). An idealized concept of the true cladistic character. *Math Bioscience*. 23: 263-272.
- Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, Higgins J, DeFelice M, Lochner A, Faggart M, Liu-Cordero SN (2002). The structure of haplotype blocks in the human genome. *Science* 296: 2225-2229.
- Gramm J, Hartman T, Nierhoff T, Sharan R, Tantau T (2009). On the complexity of SNP block partitioning under the perfect phylogeny model. *Discrete Mathematics*. 309: 5610-5617.
- Huang Y, Chao K (2008). A new framework for the selection of tag SNPs by multimarker haplotypes. *J Biomedical Informatics*. 41: 953-961.
- Hudson RR, Kaplan NL (1985). Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics*. 111: 147-164.
- Johnson GC, Esposito L, Barratt BJ, Smith AN, Heward J, Di Genova G, Ueda H, Cordell HJ, Eaves IA, Dudbridge F (2001). Haplotype tagging for the identification of common disease genes. *Nature Genet*. 29: 233-237.
- Katanforoush A, Sadeghi M, Pezeshk H, Elahi E (2009). Global haplotype partitioning for maximal associated SNP pairs. *BMC Bioinformatics*. 10: 269.
- Patil N, Berno AJ, Hinds DA, Barrett WA, Doshi JM, Hacker CR, Kautzer CR, Lee DH, Marjoribanks C, McDonough DP (2001). Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science* 294: 1719-1723.
- Vinterbo S, Dreiseitl S, Ohno-Machado L (2006). Approximation properties of haplotype tagging. *BMC Bioinformatics*. 7: 8.
- Wang N, Akey JM, Zhang K, Chakraborty R, 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.
- Zhang K, Deng M, Chen T, Waterman MS, Sun F (2002). A dynamic programming algorithm for haplotype block partitioning. *Proc Natl Acad Sci USA*. 99: 7335-7339.