

A Note on efficient computation of Haplotypes via Perfect Phylogeny

Vineet Bafna* Dan Gusfield† Sridhar Hannenhalli‡ Shibu Yooseph§

September 5, 2003

1 Introduction

With the completion of the draft sequencing of the human genome [13, 18], a natural next step is to identify, and characterize, the variations that explain the diversity of the human species. Much of the variation can be explained by *Single Nucleotide Polymorphisms (SNPs)* which correspond to point mutations that have accumulated in the population. Large scale resequencing and genotyping efforts are underway that aim to construct a map of SNPs in the human genome. It is expected that such a map will be invaluable in identifying genetic causes of complex disorders.

Humans, being diploid, contain two copies of every chromosome (and therefore every SNP), inherited from the mother, and father. Most current strategies for SNP determination allow the demarcation of markers as being homozygous or heterozygous (i.e. whether the maternal and paternal alleles are identical or not), but not *haplotyping*, i.e., determining which alleles are maternal, and which are paternal. Obtaining haplotypes have been shown to increase the power of mapping of disease genes. In the absence of biological techniques, various statistical [7, 17, 15], and combinatorial algorithms [1, 4, 6, 11] have been devised to extract haplotype information from genotype data. This is of particular interest in regions where there is little, or no recombination, and the various

*Computer Science Department, UC San Diego, Email: vbafna@cs.ucsd.edu

†Dept. of Computer Science, 3051 Engineering II, University of California, One Shields Avenue, Davis, CA 95616.
Email: gusfield@cs.ucdavis.edu Research Supported by NSF grant DBI-9723346 and EIA-0220154

‡

§Informatics Research, Applied Biosystems, 45 W. Gude Drive, Rockville, MD 20850. Email:
Shibu.Yooseph@celera.com

SNPs are tightly linked. Indeed, initial studies [5] point to varying recombination rates across the genome, and reveal many regions with little or no genetic recombination in the population.

Motivated by these considerations, Gusfield [11] formulated a combinatorial version of the haplotyping problem. Given a region in which there have been 0 recombination events, and making the *infinite sites* assumption (each mutation hits a new site on the genome), the set of haplotype sequences form a *perfect phylogeny*. The PPH problem, as formulated in [11] is stated as follows: Given n diploid sequences, can they be resolved into a set of $2n$ haploid sequences which form a perfect phylogeny. If they can, find an implicit representation of all the solutions, so that each solution can be explicitly generated in $O(m)$ time. Gusfield [11] presented an efficient algorithm that solved the PPH problem in $O(nm\alpha(nm))$ time. However, the proof made use of established, difficult results in the matroid and graph theory literature, which are difficult to implement, and a more direct approach to solving the problem was desired. Subsequent work [1, 6] closed this gap, showing direct, easy to understand, yet efficient solutions. Both solutions have complexity $\Theta(nm^2)$ which is higher than the $O(nm\alpha(nm))$ solution. Another implementation [3, 2] follows the high-level idea in [11], but runs in $O(nm^2)$ time.

The algorithm in Bafna et al. [1] has four steps: *genotype graph construction*, *Edge assignment*, *haplotype matrix creation*, and *validation*. Each step, except for the first could be accomplished in time $O(nm + m^2)$, and the $\Theta(nm^2)$ bound is essentially for the first step of genotype graph construction, so any reduction in the time for the first step reduces the overall running time. Reducing that time was left as an open problem in [1]. The algorithm in Eskin et al. [6] also has a first step which is very similar to the first step in [1], also taking $\Theta(nm^2)$ time. The time-bound established for the steps needed beyond the first step is $O(nm^2)$. Additionally, if there is more than one solution to the PPH problem, there is a space efficient representation for all of the possible solutions, as well as a time efficient algorithm for exposing all solutions ([1, 6]). However, the number of PPH solutions may be very large, and it is an open problem to select one that is optimal under some reasonable criterion.

In this short note, we address the issue of whether the first steps of the algorithms in [1, 6] can be sped-up to $O(nm + m^2)$ time, which would imply an $O(nm)$ time-bound for the PPH problem. In addition, if there are multiple solutions, we address the question of finding one that is optimal w.r.t the number of haplotypes.

We were motivated to try to improve the running time of the first step by three facts: First, the the output of the first step is of size $O(m^2)$; second the result in [11] shows that $o(nm^2)$ time is possible for the PPH problem; third, the fact established in the Section 3 shows that if the PPH problem could be solved in $O(nm + m^2)$ time, then it can be solved in $O(nm)$ time. For the second problem, minimizing the number of possible haplotypes is reasonable based on the considerations of parsimony. Indeed, many haplotyping algorithms use the parsimony criterion implicitly, or explicitly.

We give reductions that suggests that the answer to both questions is “no”. For the first problem, we show that computing the output of the first step (in either method) is equivalent to Boolean matrix multiplication. Therefore, the best bound we can presently achieve is $O(nm^{\omega-1})$, where $\omega \leq 2.52$ is the exponent of matrix multiplication. Thus, any linear time solution to the PPH problem, likely requires a different approach. For the second problem of computing a PPH solution that minimizes the number of distinct haplotypes, we show that the problem is NP-hard using a reduction from Vertex Cover [8].

2 The PPH solution in [1]

A matrix-oriented approach is taken in [1] to solve the PPH problem. In our discussion we will use the same notations as in that work. The input to the problem is a $n \times m$ genotype matrix M with $M[i, j] \in \{0, 1, 2\}$. The i -th row $M[i, *]$ describes the genotype of species s_i , and each location j where $M[i, j] = 2$ is a polymorphic site. Each column $c_j = M[*, j]$ is a polymorphic locus. The goal is to generate a $2n \times m$ *haplotype-matrix* M' , with $M'[i, j] \in \{0, 1\}$. A $2n \times m$ haplotype-matrix M' is an *expansion* of a $n \times m$ genotype matrix M , if the following hold.

1. Each row $M[i, *]$ expands to two rows denoted by $M'[i, *]$, and $M'[i', *]$.
2. For all j s.t. $M[i, j] \in \{0, 1\}$, $M[i, j] = M'[i, j] = M'[i', j]$.
3. For all j s.t. $M[i, j] = 2$, $M'[i, j] \neq M'[i', j]$.

Also, as discussed in the previous section, the set of haplotypes must form a perfect phylogeny. We use the *no-four-gamtes* characterization of perfect phylogeny, which has been independently established many times (for an exposition see [9, 10]). Define a *complete-pair-matrix* as matrix with 2 columns, containing each of the rows in $\{00, 01, 10, 11\}$. The classical theorem is that a $2n \times m$

binary matrix M' defines a unrooted perfect phylogeny if and only if no submatrix $M'[*,(j_1,j_2)]$ formed by selecting the two columns j_1, j_2 , is a *complete-pair-matrix*.

Columns j, k in a genotype matrix M are *companions*, if there exists a row i such that $M[i, j] = M[i, k] = 2$. Row i is said to be a *companion* row for columns j and k . Companion columns j, k are said to be *forced in-phase* if the expansion of the non-companion rows contains $\{00, 11\}$; similarly, companion columns j, k are said to be *forced out-of-phase* if the expansion of the non-companion rows contains $\{01, 10\}$.

The forcing relations between companion columns are described using an indicator function \mathcal{E} . For companion columns j, k , $\mathcal{E}(j, k) = 0$ if these columns are forced in-phase and $\mathcal{E}(j, k) = 1$ if they are forced out-of-phase. The task is to determine the value of $\mathcal{E}(j, k)$ for those companion columns j, k that are not forced, i.e., for which the expansion of non-companion rows does not contain either $\{00, 11\}$ or $\{10, 01\}$. In [1] it is shown that a solution to the PPH instance exists iff the indicator function setting of companion column pairs satisfies certain properties.

The algorithm proposed in [1] to solve the PPH problem has four main steps:

1. **Genotype graph construction:** In this step the genotype graph $G_M = (J, E_f \cup E_n)$ is constructed from the genotype matrix M and the indicator function \mathcal{E} is set appropriately for each edge in E_f (see below).
2. **Assignment:** The indicator function \mathcal{E} is set appropriately for each edge in E_n .
3. **Haplotype matrix creation:** The haplotype matrix M' is produced from the genotype matrix M using the indicator function assignments to the edges in G_M .
4. **Validation:** M' is checked to see if it admits a perfect phylogeny.

The genotype graph $G_M = (J, E_f \cup E_n)$ is constructed from the genotype matrix M as follows: J is the set of columns in M . An edge $(j, k) \in E_f$ iff j and k are either forced in-phase or out-of-phase. Also, $\mathcal{E}(j, k) = 0$ iff they are forced in-phase and $\mathcal{E}(j, k) = 1$ iff they are forced out-of-phase. Finally, an edge $(j, k) \in E_n$ iff j and k are companion columns but are not forced to be in-phase or out-of-phase. The graph construction phase takes $O(nm^2)$ using the obvious algorithm.

The indicator function assignment step operates on G_M . This involves a DFS-like traversal on the edges in E_f and uses the properties of G_M to assign the indicator function to edges in E_n . This

step takes $O(|J| + |E_f \cup E_n|) = O(m^2)$. The last two steps of the algorithm can both be accomplished in $O(nm)$ time.

Thus, the bottleneck to obtaining a faster algorithm for the PPH problem is the graph construction step. In Section 4 we show that constructing G_M is equivalent to boolean matrix multiplication problem.

3 $O(nm + m^2)$ -time implies $O(nm)$ -time

Theorem 1: If the PPH problem can be solved in $O(nm + m^2)$ time, by any method, then it can be solved in $O(nm)$ time.

Proof: This is certainly true if $m = O(n)$, since $O(nm + m^2) = O(nm)$ in that case. So assume that $m = \Omega(n)$.

The genotype matrix M is n by m , so a perfect phylogeny P that solves the PPH problem for input M which will have $2n$ leaves. Every internal node of P must have at least two children, so P can have at most $2n - 1$ internal nodes, and hence at most $4n - 1$ nodes and $4n - 2$ edges overall. Each mutation is assigned to only one edge in P , so for any genotype matrix M where $m > 4n - 2$, either there is no PPH solution for M , or there must be duplicate columns in M , and at most $4n - 2$ distinct columns.

Using radix sort (considering each column as a binary number), duplicate columns can be grouped together in $O(nm)$ time. If we want only a single PPH solution, we can remove all but one copy of each duplicate column, leaving at most $4n - 2$ distinct columns. This leaves a PPH problem with an input matrix M'' of size n by m'' , where $m'' = O(n)$, which can be solved in $O(n^2) = O(nm)$ time. Hence, the overall time bound to find one PPH solution, including the time for the radix sort is $O(nm)$.

If we want to find a representation of all the solutions, then we can proceed in one of two ways. It was stated in [11] that once one solution to the PPH problem is found (by whatever method), an implicit representation of all the solutions can be found in $O(m)$ additional time. However, that approach is complex, and has not been implemented. A simpler approach that is more easily implemented is shown next.

Even if there are $k > 2$ copies of a given column v in M , in any PPH solution there are at most two distinct ways that those k columns will be expanded in a solution M' . This follows from the fact that each pair v, w of the k columns is a companion pair and the columns are identical, so knowing how v is expanded and knowing the value of $\mathcal{E}(v, w)$, completely determines the expansion of w .

So assume we remove all but two copies v, w of a column that has $k > 2$ copies in M . Let \overline{M} denote the reduced matrix. A PPH solution for \overline{M} where $\mathcal{E}(v, w) = 0$ is interpreted as a PPH solution where all of the k columns are expanded the same way (in-phase). However, a PPH solution where $\mathcal{E}(v, w) = 1$ expands columns v and w differently (out-of-phase). That solution actually corresponds to $2^k - 2$ solutions, where we choose arbitrarily which of the two expansions to use for any of the k columns other than v and w .

So even when $k > 2$, we can again use $O(nm)$ -time to reduce the input to a problem of size n by m'' , where $m'' = O(n)$, and so we can find an implicit representation of all the PPH solutions in $O(nm)$ time, if the PPH problem can be solved in $O(nm + m^2)$ time. ♣

4 Equivalence of genotype graph construction and boolean matrix multiplication

Definition: Given an $n \times m$ matrix P in which the entries are drawn from the alphabet Σ , we will use \mathcal{P}_{ab} to denote the *template* matrix derived from P for the template ab (where $a, b \in \Sigma$) as follows: \mathcal{P}_{ab} is an $m \times m$ boolean matrix in which an entry $\mathcal{P}_{ab}[j, k] = 1$ iff $P[i, j] = a$ and $P[i, k] = b$, for some $1 \leq i \leq n$.

Lemma 2: The genotype graph $G_M = (J, E_f \cup E_n)$ can be constructed by querying template matrices $\mathcal{M}_{00}, \mathcal{M}_{01}, \mathcal{M}_{10}, \mathcal{M}_{11}, \mathcal{M}_{20}, \mathcal{M}_{02}, \mathcal{M}_{21}, \mathcal{M}_{12}$, and \mathcal{M}_{22} .

Proof: An edge $(j, k) \in E_f \cup E_n$ iff $\mathcal{M}_{22}[j, k] = 1$. Edge $(j, k) \in E_f$ and $\mathcal{E}(j, k) = 0$ iff $(\mathcal{M}_{22}[j, k] = 1$ and $(\mathcal{M}_{00}[j, k] = 1$ or $\mathcal{M}_{20}[j, k] = 1$ or $\mathcal{M}_{02}[j, k] = 1)$ and $(\mathcal{M}_{11}[j, k] = 1$ or $\mathcal{M}_{21}[j, k] = 1$ or $\mathcal{M}_{12}[j, k] = 1)$). Similarly, $(j, k) \in E_f$ and $\mathcal{E}(j, k) = 1$ iff $(\mathcal{M}_{22}[j, k] = 1$ and $(\mathcal{M}_{10}[j, k] = 1$ or $\mathcal{M}_{20}[j, k] = 1)$ and $(\mathcal{M}_{01}[j, k] = 1$ or $\mathcal{M}_{02}[j, k] = 1)$). We assume that an edge $(j, k) \in E_f$ cannot have both $\mathcal{E}(j, k) = 0$ and $\mathcal{E}(j, k) = 1$, as this implies that M does not have a PPH solution [1].

Finally, edge $(j, k) \in E_n$ iff $(j, k) \in E_f \cup E_n$ and $(j, k) \notin E_f$. ♣

Lemma 3: Denote by $T(h, k, l)$, the time to multiply two boolean matrices of dimensions $h \times k$, and $k \times l$ respectively. Given an $n \times m$ matrix M , a template matrix \mathcal{M}_{ab} can be computed in time $O(T(m, n, m))$.

Proof: Define boolean matrices $X_{m \times n}$ and $Y_{n \times m}$ as follows. Entry $X[j, i] = 1$ iff $M[i, j] = a$, where $1 \leq i \leq n$ and $1 \leq j \leq m$. Entry $Y[i, j] = 1$ iff $M[i, j] = b$, where $1 \leq i \leq n$ and $1 \leq j \leq m$. Let $Z = X.Y$. We claim that $\mathcal{M}_{ab} = Z$. To see why, note that entry

$$Z[j, k] = \sum_{i=1}^n X[j, i].Y[i, k], \quad \text{where } 1 \leq i \leq n \text{ and } 1 \leq j \leq k \leq m$$

Thus

$$\begin{aligned} Z[j, k] = 1 & \quad \text{iff} \quad X[j, i] = 1 \text{ and } Y[i, k] = 1, \text{ for some } 1 \leq i \leq n \\ & \quad \text{iff} \quad M[i, j] = a \text{ and } M[i, k] = b \\ & \quad \text{iff} \quad \mathcal{M}_{ab}[j, k] = 1 \end{aligned}$$

♣

Corollary 4: $G_M = (J, E_f \cup E_n)$ can be constructed in $O(nm^{\omega-1})$, where ω is the constant in boolean matrix multiplication.

Proof: Consider the boolean matrices $X_{m \times n}$ and $Y_{n \times m}$ defined in Lemma 3. Since $m \leq n$, we can partition them as shown

$$X = [X(1) \quad X(2) \quad \dots \quad X(n/m)] \text{ and } Y = \begin{bmatrix} Y(1) \\ Y(2) \\ \dots \\ Y(n/m) \end{bmatrix}$$

where each $X(i)$ is an $m \times m$ matrix and each $Y(j)$ is also an $m \times m$ matrix.

Thus

$$X.Y = \sum_{i=1}^{n/m} X(i).Y(i)$$

Let $T(m)$ denote the time taken to multiply two $m \times m$ boolean matrices. Thus, the total time taken to compute $X.Y$ is $O((\frac{n}{m})T(m))$. Since $T(m) = O(m^\omega)$, it follows that the total time taken is $O(nm^{\omega-1})$. ♣

Lemma 5: Let $T_G(n, m)$ be the time to construct a genotype graph, given an $n \times m$ genotype matrix M . two boolean matrices with dimensions $m_1 \times n$, and $n \times m_2$ respectively can be multiplied in time $O(T_G(n, m_1 + m_2))$.

Proof: Consider two boolean matrices $X_{m_1 \times n}$ and $Y_{n \times m_2}$. Let $A_{m_1 \times n}$ be a matrix derived from X by replacing each 1 by a 2. Similarly, let $B_{n \times m_2}$ be a matrix derived from Y by replacing each 1 by a 2. We define the genotype matrix $M_{n \times (m_1 + m_2)} = [A^T B]$, where A^T denotes the transpose of A . Note that the j^{th} column of B appears as the $(m_1 + j)^{th}$ column in M .

Let $G_M = (J, E_f \cup E_n)$ denote the genotype graph for M and let $Z_{m_1 \times m_2} = X.Y$. We claim that $Z[j, k] = 1$ iff $(j, m_1 + k) \in E_f \cup E_n$. The proof follows.

$$\begin{aligned} Z[j, k] = 1 & \quad \text{iff} \quad X[j, i] = 1 \text{ and } Y[i, k] = 1, \text{ for some } 1 \leq i \leq n \\ & \quad \text{iff} \quad A[j, i] = 2 \text{ and } B[i, k] = 2 \\ & \quad \text{iff} \quad M[i, j] = 2 \text{ and } M[i, m_1 + k] = 2 \\ & \quad \text{iff} \quad (j, m_1 + k) \in E_f \cup E_n \end{aligned}$$

For genotype matrix $M_{a \times b}$, let $T(a, b)$ denote the time taken to construct the genotype graph G_M . From the above reduction, it follows that the time taken to compute the product of $X_{m_1 \times n}$ and $Y_{n \times m_2}$ is $O(T(n, m_1 + m_2))$.

♣

5 Complexity of MIN-PPH problem

When the solution to the PPH problem is not unique, we seek a criteria to identify a solution that is more biologically compelling than the others. One criteria that has strong empirical and theoretical support is that of *maximum parsimony*, i.e., a solution that uses the fewest number of distinct haplotypes to explain the given genotypes. The parsimony criteria is discussed in [14, 12, 16, 19] in the context of the haplotyping problem, where a set of haplotypes must explain the genotypes but are not required to solve the PPH problem. In particular, in [12, 16] the parsimony criteria was shown to be effective in identifying highly accurate haplotyping solutions (in simulations and in the few cases where we know the true underlying haplotypes). A natural conjecture is that this is also

true in the context of the PPH problem. Hence, if there is more than one PPH solution, it is of interest to obtain and examine a most-parsimonious PPH solution, if only to test (in cases when we know the true underlying haplotypes) the conjecture that a most-parsimonious PPH solution is most often highly accurate. Unfortunately, obtaining a most-parsimonious PPH solution is likely to be much more difficult than finding an arbitrary PPH solution, as we show next.

The MIN-PPH problem:

Input: An instance of the PPH problem, and an integer d .

Output: Does a PPH solution exist with d or fewer distinct haplotypes?

Theorem 6: MIN-PPH is NP-Complete.

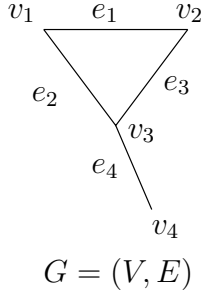
Proof: The problem is clearly in the class NP. To show that it is NP-Hard, we show a reduction from the Vertex Cover Problem [8].

Let graph $G = (V, E)$ be the input to the Vertex Cover Problem. We will construct a genotype set \mathcal{G} with $|V| + |E|$ genotypes and $2|V| + |E|$ sites, as the corresponding instance of the MIN-PPH problem. We define $\mathcal{G} = \{a_1, a_2, \dots, a_{|V|}\} \cup \{b_1, b_2, \dots, b_{|E|}\}$, where genotype a_i is associated with vertex v_i and genotype b_j is associated with edge $e_j = (v_{i_1}, v_{i_2})$. We now define the settings at each site for each genotype. Genotype a_i , where $1 \leq i \leq |V|$, has a 1 at sites i and $|V| + i$, and a 0 at each of the remaining sites. Genotype b_j , where $1 \leq j \leq |E|$, has a 2 at sites i_1, i_2 , and $2|V| + j$, and a 0 at each of the remaining sites.

Figure 1 shows an example of the reduction. In this example, the genotype set \mathcal{G} has eight genotypes - the four a_i 's are associated with V and the four b_j 's are associated with E . Each genotype has $2|V| + |E| = 12$ sites. Vertex cover of $\{v_1, v_3\}$ for graph G corresponds to PPH solution H with 10 distinct haplotypes for \mathcal{G} , where $H = \{100010000000, 010001000000, 001000100000, 000100010000, 010000001000, 100000000100, 010000000010, 000100000001, 100000000000, 001000000000\}$. The last two haplotypes in H correspond to vertices v_1 and v_3 respectively.

We make two observations about the nature of any solution for the MIN-PPH instance \mathcal{G} -

- (Observation 1) Each genotype a_i is homozygous in every site, i.e. each a_i is also a haplotype. Note that this haplotype is unique to a_i ; that is, it cannot be used to resolve any other genotype in \mathcal{G} . The reason for this is due to the 1 at site $|V| + i$.



$$\begin{aligned}
 a_1 &= 100010000000 \\
 a_2 &= 010001000000 \\
 a_3 &= 001000100000 \\
 a_4 &= 000100010000 \\
 b_1 &= 2200000002000 \\
 b_2 &= 2020000000200 \\
 b_3 &= 0220000000020 \\
 b_4 &= 0022000000002
 \end{aligned}$$

$$\mathcal{G} = \{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4\}$$

Figure 1: An example showing the reduction.

- (Observation 2) For each genotype b_j , one of the two haplotypes that resolve it, namely the one that has a 1 at site $2|V| + j$, is unique to b_j .

We claim that the graph G has a vertex cover of size k iff the MIN-PPH instance has a solution with $d = |V| + |E| + k$ distinct haplotypes. The proof follows.

(\rightarrow) Suppose that G has a vertex cover of size k . We will construct a set of haplotypes H as follows. From observation 1, it follows that the a_i 's will contribute $|V|$ distinct haplotypes to H . Now, consider the genotype b_j that is associated with edge $e_j = (v_{i_1}, v_{i_2})$. W.l.o.g., assume that v_{i_1} is a vertex that is part of the vertex cover.

Then b_j will be resolved into two haplotypes as follows. The first haplotype (denote it as a *VC-haplotype*) has a 1 at site i_1 and a 0 at each of the remaining sites, and the second haplotype has a 1 at sites i_2 and $2|V| + j$, and a 0 at each of the remaining sites. From observation 2, the second haplotype is unique to b_j . Furthermore, since the vertex cover has size k , there will be k distinct VC-haplotypes. Thus, H has $|V| + |E| + k$ distinct haplotypes. By construction, each genotype in \mathcal{G} is resolved by haplotypes from H .

We now show that H is a PPH solution. Let's consider the haplotypes which have two 1's. There is one for each a_i genotype and one for each b_j genotype. In the case of a_i , the second 1 is in column $|V| + i \leq 2|V|$, which has only 0's in the other a -derived haplotypes (because, the only a -derived haplotype with a 1 in column $|V| + i$ is an a_i -derived haplotype), and all the b -derived haplotypes also have a 0 in column $|V| + i$ (because every entry in a b -derived haplotype is 0 in columns $|V| + 1$ through $2|V|$). Hence a pair of columns which are 1,1 in an a -derived haplotype, cannot be 0,1 in

any of the other haplotypes. Now consider the b_j -derived haplotype that has two 1's. Its second 1 is in column $2|V| + j$. Every a -derived haplotype has a 0 entry in that column, and among the b -derived haplotypes, only one b_j -derived haplotype has a 1 in that column. Hence a pair of columns which are 1,1 in an b -derived haplotype, cannot be 0,1 in any of the other haplotypes. It follows that H contains no complete-pair submatrix, and so is a PPH solution.

(\leftarrow) Suppose that \mathcal{G} has a PPH solution H containing $|V| + |E| + k$ distinct haplotypes. We will show how to construct a vertex cover of size k in G . First, consider the genotype a_i . From observation 1, each a_i contributes a distinct haplotype in H . Next, consider the genotype b_j that is associated with edge $e_j = (v_{i_1}, v_{i_2})$. From observation 2, we have that b_j contributes one distinct haplotype to H . The second haplotype (denote it by h_j) contributed by b_j may or may not be unique. Furthermore, this second haplotype has a 1 in exactly one of site i_1 or i_2 , and a 0 at each of the remaining sites. This is because for H to be a PPH solution, the sites (columns) i_1 and i_2 have to be forced out-of-phase because genotype a_{i_1} has a 1 at site i_1 and 0 at site i_2 while genotype a_{i_2} has a 0 at site i_1 and a 1 at site i_2 . Let H' denote the set of h_j 's. From the preceding arguments, $|H'| = k$. Define the vertex set $V' = \{v_i | \exists h_j \in H' \text{ that has a 1 at site } i\}$. For each $1 \leq j \leq |E|$, genotype b_j (which is associated with edge e_j) has some $h \in H'$ that resolves this genotype; it follows that V' is a vertex cover of size k for the graph G .



References

- [1] V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogeny: A direct approach. Technical report, U.C. Davis, 2002. To appear in *Jnl. Computational Biology*.
- [2] R.H. Chung and D. Gusfield. Empirical exploration of perfect phylogeny haplotyping and haplotypers. In *Proceedings of the 9th International Conference on Computing and Combinatorics*, volume 2697 of *LNCS*, pages 5–19, 2003.
- [3] R.H. Chung and D. Gusfield. Perfect phylogeny haplotyper: Haplotype inference using a tree model. *Bioinformatics*, 19(6):780–781, 2003.
- [4] Andrew Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol. Biol. Evol.*, 7:111–122, 1990.

- [5] M. Daly, J. Rioux, S. Schaffner, T. Hudson, and E. Lander. High-resolution haplotype structure in the human genome. *Nature*, 29:229–232, 2001.
- [6] E. Eskin, E. Halperin, and R.M. Karp. Large scale reconstruction of haplotypes from genotype data. In *Proceedings of Seventh Annual International Conference on research in Computational Molecular Biology (RECOMB)*, pages 104–113, 2003.
- [7] L. Excoffier and M. Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol. Bio. Evolution*, 12:921–927, 1995.
- [8] M. Garey and D. Johnson. *Computers and intractability*. Freeman, San Francisco, 1979.
- [9] D. Gusfield. Efficient algorithms for inferring evolutionary history. *Networks*, 21:19–28, 1991.
- [10] D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
- [11] D. Gusfield. Haplotyping as Perfect Phylogeny: Conceptual Framework and Efficient Solutions (Extended Abstract). In *Proceedings of RECOMB 2002: The Sixth Annual International Conference on Computational Biology*, pages 166–175, 2002.
- [12] D. Gusfield. Haplotype inference by pure parsimony. In R. Baeza-Yates, E. Chavez, and M. Chrochemore, editors, *14'th Annual Symposium on Combinatorial Pattern Matching (CPM'03)*, volume 2676 of *Springer LNCS*, pages 144–155, 2003.
- [13] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.
- [14] G. Lancia, C. Pinotti, and R. Rizzi. Haplotyping populations: Complexity and approximations, technical report dit-02-082. Technical report, University of Trento, 2002.
- [15] T. Niu, Z. Qin, X. Xu, and J.S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Am. J. Hum. Genet*, 70:157–169, 2002.
- [16] S. Orzack, D. Gusfield, , J. Olson, S. Nesbitt, and V. Stanton. Analysis and exploration of the use of rule-based algorithms and consensus methods for the inferral of haplotypes. *Genetics*, to appear.

- [17] M. Stephens, N. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *Am. J. Human Genetics*, 68:978–989, 2001.
- [18] J.C. Venter, M.D. Adams, G.G. Sutton, A.R. Kerlavage, H.O. Smith, and M. Hunkapillar. Shotgun sequencing of the human genome. *Science*, 280:1540–1542, 1998.
- [19] L. Wang and Y. Xu. Haplotype inference by maximum parsimony. *Bioinformatics*, to appear.