Islands of Tractability for Parsimony Haplotyping

Roded Sharan, Bjarni V. Halldórsson and Sorin Istrail

Abstract

We study the parsimony approach to haplotype inference, which calls for finding a set of haplotypes of minimum cardinality that explains an input set of genotypes. We prove that the problem is APX-hard even in very restricted cases. On the positive side, we identify islands of tractability for the problem, by focusing on instances with specific structure of haplotype sharing among the input genotypes. We exploit the structure of those instance to give polynomial and constant-approximation algorithms to the problem. We also show that the general parsimony haplotyping problem is fixed parameter tractable.

Index Terms

A preliminary version of this paper appeared in Proc. CSB 2005 [25].

R. Sharan is with the School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. Email: roded@tau.ac.il, phone: +972-3-640-5391, fax: +972-3-640-9357.

B.V. Halldórsson is with deCODE Genetics, Sturlugata 8, 101 Reykjavik, Iceland. Email: bjarnih@decode.is, phone: +354-570-2848, fax: +354-570-1903

S. Istrail is with the Computer Science Department and the Center for Computational Molecular Biology, Brown University, Providence, RI 02912. Email: sorin@cs.brown.edu, phone: 401-863-6196, fax: 401-863-7657.

Manuscript submitted July 2005.

RS and BVH contributed equally to this work.

Correspondence should be addressed to RS.

J.3.a Biology and Genetics, G.2.2.a Graph Algorithms, F.2.0 Analysis of algorithms and Problem Complexity

I. Introduction

Single nucleotide polymorphisms (SNPs) are differences in a single base, across the population, within an otherwise conserved genomic sequence. SNPs are the most common form of variation of DNA sequences among individuals. Especially when occurring in coding or otherwise functional regions, variations in SNP content are linked to medical condition or may affect drug response.

A SNP commonly has two variants, or *alleles*, in the population, corresponding to two of the four genomic letters A, C, G, and T. The sequence of alleles in contiguous SNP positions along a chromosomal region is called a *haplotype*. For diploid organisms, the *genotype* specifies for every SNP position the particular alleles that are present at this site in the two chromosomes. Genotype data contains information only on the combination of alleles at a given site, and does not reveal the association of each allele with one of the two chromosomes–its *phase*. Current technologies, suitable for large-scale polymorphism screening only yield the genotype information at each SNP site. The actual haplotypes in the typed region can be obtained at a considerably higher cost [23]. Due to the importance of haplotype information for inferring population history and for disease association, it is desirable to develop efficient methods for inferring haplotypes from genotype information.

Numerous approaches have been suggested in the literature to resolve haplotypes from genotype data. These methods include the seminal approach of Clark [4] and related parsimony approaches [9], [10], [12]; maximum likelihood methods [5], [6], [16], [21]; Bayesian methods such as PHASE [26], HAPLOTYPER [22] and HaploBlock [8]; and perfect-phylogeny-based approaches [11], [1], [14]. The reader is referred to [13] for a survey on different formulations of the haplotyping problem.

Here we focus on the *parsimony haplotyping (PH)* problem, where the input is a set of n genotypes and the goal is to find a minimum set of haplotypes that explains them (a formal definition of PH is deferred to Section II). Parsimony is a natural criterion for choosing a solution in many domains. This is particularly true for haplotyping, since the number of distinct haplotypes observed in a population is much smaller than the number of possible haplotypes, due to population bottleneck effects and genetic drift. For example, Patil et al. report that within short genomic regions, typically, some 70-90% of the haplotypes belong to very few (2-5) common haplotypes [23].

There has been extensive research on the parsimony haplotyping problem. Hubbell has shown that the problem is NP-complete [18]. Lin et al. have investigated a related problem and showed that it is NP-complete as well [20]. A practical integer programming approach for it was devised by Gusfield [12]. Recently, Lancia et al. [19] have shown that the problem is APX-hard and have given a 2^{k-1}-approximation algorithm for the problem, for data sets in which each genotype has at most k ambiguous positions. Huang et al. [17] have given an $O(\log n)$-approximation algorithm for the problem, for data sets in which there is a polynomial number of haplotypes to be considered.

In this paper we study the complexity and approximability of parsimony haplotyping and its restrictions. We characterize instances of the problem by the number of ambiguous sites they contain and the structure of a *Clark-consistency graph* whose vertices correspond to genotypes and whose edges represent sharing of haplotypes. On the negative side, we show that parsimony haplotyping is APX-hard even when the input instances have small numbers of ambiguous sites per genotype or SNP; when the corresponding Clark-consistency graph is a clique; or when the Clark-consistency graph is bipartite. On the positive side, we show that the problem is fixed parameter tractable, and give polynomial algorithms and approximation algorithms for some of its restrictions. Specifically, we give a polynomial algorithm for PH on cliques when each SNP has at most two genotypes in which it is ambiguous. We also give a polynomial algorithm for PH when the Clark-consistency graph has bounded treewidth. Finally, we give a 1.5-approximation algorithm for PH when the input instance induces a bipartite Clark-consistency graph.

The paper is organized as follows: Section II provides background on the problem. The complexity of parsimony haplotyping is analyzed in Section III. Restrictions of the problem are studied in Sections IV-VI.

II. Preliminaries

A *haplotype* is a row vector with binary entries. Each position in the vector indicates the state (0 or 1) of a certain SNP in this haplotype. For a haplotype h, let $h[i]$ denote the ith position of h. A *genotype* is a row vector with entries in $\{0, 1, 2\}$, each corresponding to a SNP site. A *genotype matrix* is a matrix whose rows are genotypes. We denote the number of genotypes by n. Two haplotypes h_1 and h_2 *explain* a genotype g, denoted by $h_1 \oplus h_2 = g$, if for each position i the following holds: $g[i] \in \{0, 1\}$ implies $h_1[i] = h_2[i] = g[i]$; and $g[i] = 2$ implies $h_1[i] \neq h_2[i]$. If $h[i] = g[i]$ whenever $g[i] \in \{0, 1\}$ then h is said to be *consistent* with g.

A haplotype that is consistent with two genotypes is said to be *shared* by them. Given a set of genotypes, the graph containing the genotypes as nodes and an edge between two genotypes if and only if they share a haplotype is called the *Clark-consistency graph*. This definition is inspired by Clark's rule for haplotype inference [4] as is explained below. A (k, l)-*bounded instance* is an input genotype matrix with at most k 2-entries per row and at most l 2-entries per column, where an asterisk instead of k or l indicates no constraint. An *enumerable* instance is an input genotype matrix with a polynomial number of haplotypes that are consistent with any of its genotypes or, equivalently, an ($O(\log n)$,*)-bounded instance.

The parsimony haplotyping problem is formally defined as follows:

Problem 1 (Parsimony Haplotyping (PH)): Given a set of genotypes, find a minimum set of haplotypes H such that each genotype can be explained by two haplotypes from H.

A related problem concerns identifying haplotypes that are consistent with the input set of genotypes:

Problem 2 (Minimum Haplotype Consistency (MHC)): Given a set of genotypes, find a minimum set of haplotypes H such that each genotype is consistent with some element of H.

Inference paths in the Clark-consistency graph are defined as follows: For a haplotype h and a genotype g that is consistent with it, an *inference path* is a path in the Clark-consistency graph that starts at g and is created as follows: (1) let $g = h \oplus \bar{h}$; (2) move to a genotype g' that is consistent with $\bar{h}$ if such exists and was not visited already; (3) set $g = g'$, $h = \bar{h}$ and go to step (1). The path terminates when it reaches a haplotype h whose complement is consistent with genotypes in the path only. Its *length* is defined to be its number of edges.

III. Complexity of Parsimony Haplotyping

The general parsimony haplotyping problem is known to be NP-complete [18] and APX-hard [19], and, hence, unlikely to admit a polynomial time approximation scheme. In fact, the construction in the hardness proof of Lancia et al. [19] shows that the problem is APX-hard already for $(3, *)$-bounded instances. In the following we strengthen their result and prove that parsimony haplotyping is APX-hard even for $(4, 3)$-bounded instances.

Theorem 1: Parsimony haplotyping is NP-hard for $(4, 3)$-bounded instances.

Proof: We give a reduction from 3-Dimensional Matching with each element occurring in at most 3 triples (3DM3) [7]: given disjoint sets X, Y, Z containing ν elements each, and a set $C = \{c_0, \ldots, c_{\mu-1}\}$ of μ triples in $X \times Y \times Z$ such that each element occurs in at most three triples of C, find a maximum cardinality set $C' \subseteq C$ of disjoint triples (a 3-dimensional matching).

We build a genotype matrix with $3\nu + 3\mu$ rows and $6\nu + 4\mu$ columns. The first 3ν rows are called *element genotypes* and represent the elements of the 3DM3 instance. The other 3μ rows are called *matching genotypes* and represent the triples. The first 3ν columns are used to ensure that for each element genotype, at most one of its haplotypes can be shared. The next 3ν columns ensure that element genotypes do not share haplotypes with each other; they can only share haplotypes with genotypes corresponding to triples they occur in. The next 4μ columns represent the triples and restrict the sharing of haplotypes among the matching genotypes, as described below.

The construction of the genotype matrix is based on the gadget shown in Figure 1. For each element $x_i \in X$, $y_i \in Y$, or $z_i \in Z$ we construct one genotype. In the following we specify for each genotype its non-zero entries only.

- $x_i[i] = 2$; $x_i[3\nu + i] = 1$; $x_i[6\nu + 4j] = 2$ for all j such that $x_i \in c_j$.
- $y_i[\nu + i] = 2$; $y_i[4\nu + i] = 1$; $y_i[6\nu + 4j] = 2$ for all j such that $y_i \in c_j$.
- $z_i[2\nu + i] = 2$; $z_i[5\nu + i] = 1$; $z_i[6\nu + 4j] = 2$ for all j such that $z_i \in c_j$.

For each triple $c_j \in C$ we create 3 genotypes, whose non-zero entries are:

- $c_j^x[3\nu + i] = 2$ for i such that $x_i \in c_j$; $c_j^x[6\nu + 4j] = 1$; $c_j^x[6\nu + 4j + 1] = 2$.
- $c_j^y[4\nu + i] = 2$ for i such that $y_i \in c_j$; $c_j^y[6\nu + 4j] = 1$; $c_j^y[6\nu + 4j + 2] = 2$.
- $c_j^z[5\nu + i] = 2$ for i such that $z_i \in c_j$; $c_j^z[6\nu + 4j] = 1$; $c_j^z[6\nu + 4j + 3] = 2$.

$$\begin{array}{c|ccc|ccc|cccc}
x_i & 2 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 \\
y_i & 0 & 2 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 0 \\
z_i & 0 & 0 & 2 & 0 & 0 & 1 & 2 & 0 & 0 & 0 \\
\hline
c^x & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 2 & 0 & 0 \\
c^y & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 & 2 & 0 \\
c^z & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 0 & 0 & 2
\end{array}$$

Fig. 1. Gadget for the reduction in Theorem 1.

The resulting genotype matrix A is $(4, 3)$-bounded. Indeed, each element genotype contains exactly one 2-entry in one of the first 3ν columns and at most three other 2-entries representing the triples in which the element occurs. Each matching genotype has exactly two 2-entries. For the bound on the columns, observe that the first 3ν columns contain one 2-entry; the next 3ν columns have at most three 2-entries, since their corresponding elements occur in at most three triples. The last 4μ columns contain at most three 2-entries each.

We now claim that A has a parsimony solution of cardinality $6\nu + 4\mu - \omega$ if and only if C has a matching of size ω. First, observe that every set of three matching genotypes can be phased using four haplotypes, none of which can be shared with the element genotypes, or using 6 haplotypes, 3 of which (left column) can be shared with element genotypes, as depicted in Figure 2.

For the ’if’ part, suppose that C has a matching of size ω. For each $c \in C$ we phase the corresponding matching genotypes using the template $\mathcal{P}_6$, as shown in Figure 2. Three of those six haplotypes can be used to phase the corresponding element genotypes, where each element genotype requires one additional haplotype to complete its phasing. Overall, the phasing uses 9ω

$$\left\{\begin{array}{ccc} (0001001100) & \oplus & (0000001000) \\ (0000101010) & \oplus & (0000001000) \\ (0000011001) & \oplus & (0000001000) \end{array}\right\} \overset{\mathcal{P}_4}{\Longleftarrow} \left\{\begin{array}{c} (0002001200) \\ (0000201020) \\ (0000021002) \end{array}\right\} \overset{\mathcal{P}_6}{\Longrightarrow} \left\{\begin{array}{ccc} (0001001000) & \oplus & (0000001100) \\ (0000101000) & \oplus & (0000001010) \\ (0000011000) & \oplus & (0000001001) \end{array}\right\}$$

Fig. 2. The three matching genotypes corresponding to a triple and alternative phasings of these genotypes. $\mathcal{P}_4$ show a minimal phasing with 4 haplotypes, none of which can be shared with the element genotypes. $\mathcal{P}_6$ shows a phasing using 6 haplotypes, 3 of which can be shared with the element genotypes.

haplotypes for this set of genotypes. The remaining element genotypes can be phased arbitrarily using two haplotypes each. The remaining matching genotypes can be phased using the $\mathcal{P}_4$ template by 4 haplotypes each, as shown in Figure 2. In total, the phasing includes $9\omega + 2 \cdot 3(\nu - \omega) + 4(\mu - \omega) = 6\nu + 4\mu - \omega$ haplotypes.

Conversely, given a phasing of A using $6\nu + 4\mu - \omega$ haplotypes, we can construct a matching of size ω, by letting our matching be those triples whose corresponding matching genotypes share haplotypes with all three of their element genotypes. By construction, element genotypes cannot share haplotypes among themselves, so their phasing requires 6ν haplotypes. Consider any triple t of matching genotypes. These genotypes can only share haplotypes with each other or with the corresponding element genotypes. Furthermore, t can share at most 3 haplotypes with its element genotypes. If t shares exactly 3 haplotypes with its element genotypes (in the given phasing) then, by construction, it is phased using 6 haplotypes in total. If t shares less than 3 haplotypes with its element genotypes, it must be phased using 4 additional haplotypes that are not shared with the element genotypes. Hence, the resulting matching has size at least ω. ■

Corollary 1: Parsimony haplotyping is APX-hard for $(4, 3)$-bounded instances.

Proof: Petrank [24] has shown that it is NP-hard to determine whether a maximum matching of a 3DM3 instance is perfect or misses a constant fraction ϵ of the elements. In the first case, our genotype instance admits a solution of cardinality $5\nu + 4\mu$; in the second case, it admits a solution of cardinality at most $5\nu + 4\mu + \epsilon\nu$. The claim follows. ■

We now show that the related problem of ’covering’ the input genotypes is hard as well.

Theorem 2: MHC is NP-complete.

Proof: The problem is clearly in NP. We reduce from CLIQUE COVER [7]. Given an instance of CLIQUE COVER, consisting of a graph $G = (\{1, \ldots, n\}, E)$ and an integer k, we build an $n \times n$ genotype matrix as follows: For each vertex i we have a corresponding row r^i. We set $r^i_i = 1$. For all vertices j that are adjacent to i we set $r^i_j = 2$. All other entries of r^i are set to 0. It is easy to see that a haplotype is consistent with a set of genotypes (rows) if and only if the corresponding vertices form a clique in G. Hence, there is a 1-1 correspondence between solutions to CLIQUE COVER and solutions to the MHC instance. ■

We note that a similar reduction from CLIQUE shows that even the problem of identifying a haplotype that is consistent with a maximum number of genotypes is NP-hard. Moreover, these reductions also show that both problems are NP-hard to approximate to within a factor of $n^{1-\epsilon}$, unless NP=ZPP [15].

On the positive side, we now show that PH is fixed parameter tractable with respect to the cardinality of the solution set of haplotypes.

Theorem 3: Parsimony haplotyping is fixed parameter tractable with respect to to the number of haplotypes in the solution set.

Proof: Fixing the number of allowed haplotypes to k implies that the maximum number of distinct genotypes possible is $\frac{k(k+1)}{2}$. Let m be the length of the input genotypes. Denote the unknown haplotypes in an optimal solution by $h_1, \ldots, h_k$. For each genotype, we can enumerate the pair of indices of the solution haplotypes that explain it. The problem is then reduced to solving m sets of linear equations over GF(2). Each set of equations involves at most two variables per equation and can be viewed as a 2-SAT instance. Hence, resolving the haplotypes given their assignment to genotypes can be done in $O(mk^2)$ time, and the overall complexity of the algorithm is $O(mk^{k^2+k})$. ■

The rest of the paper concerns identifying islands of tractability for parsimony haplotyping. We show positive results for instances in which the Clark-consistency graph is a $(*, 2)$-bounded clique or has bounded treewidth, as well as approximation algorithms for several variants, including instances for which the Clark-consistency graph is bipartite.

IV. Parsimony on Cliques

In this section we study complete Clark-consistency graphs (cliques), corresponding to instances in which every two genotypes share a haplotype. We call such an instance a *clique instance*. For a clique instance, every column in the genotype matrix can contain at most two values (out of $\{0, 1, 2\}$), one of which is 2. W.l.o.g., we shall consider matrices with only 0-s and 2-s. In particular, the all-zero haplotype is shared by all the genotypes and is called *trivial*. When the input instance contains the all-0 genotype, any solution to it must contain the trivial haplotype. For ease of presentation, we assume in the following that the input instance does not contain the all-0 genotype.

Theorem 4: Parsimony haplotyping is NP-hard on cliques.

Proof: We give a reduction from 3DM3, similar to that in the proof of Theorem 1. The input to the 3DM3 instance includes disjoint sets X, Y, Z containing ν elements each, and a set $C = \{c_0, \ldots, c_{\mu-1}\}$ of μ triples in $X \times Y \times Z$. Let ν_1, ν_2 denote the number of elements that only occur in 1 or 2 triples, respectively. We build a genotype matrix A with $21\nu + 6\mu$ rows and $6\nu + 4\mu + 8\nu_1 + 4\nu_2$ columns. The first 21ν rows are called *element genotypes* and represent the elements of the 3DM3 instance. The other 6μ rows are called *matching genotypes* and represent the triples.

To ensure that every element occurs in exactly three sets, we start by constructing $2\nu_1 + \nu_2$ singleton sets. Each element that occurs in two triples is assigned to one singleton set, and each element that occurs in one triple is assigned to two singleton sets. We label the singleton sets $c_{\mu+1}$ through $c_{\mu+2\nu_1+\nu_2}$.

For the ith element $\gamma \in X \cup Y \cup Z$, occurring in sets (triples or singleton sets) c_{j_1}, c_{j_2} and c_{j_3}, we construct seven element genotypes $\gamma_1, \ldots, \gamma_7$ (see Figure 3). Let $l_\gamma = 1$ if $\gamma \in X$, $l_\gamma = 2$ if $\gamma \in Y$, and $l_\gamma = 3$ if $\gamma \in Z$. The 2-entries of γ_k, $1 \leq k \leq 7$, are as follows:

- $\gamma_k[2i] = 2$ if $k \leq 3$; $\gamma_k[2i+1] = 2$.
- $\gamma_k[6\nu + 4j_1] = \gamma_k[6\nu + 4j_1 + l_\gamma] = 2$ if $k \in \{2, 3, 5, 6, 7\}$.
- $\gamma_k[6\nu + 4j_2] = \gamma_k[6\nu + 4j_2 + l_\gamma] = 2$ if $k \in \{1, 3, 4, 6, 7\}$.
- $\gamma_k[6\nu + 4j_3] = \gamma_k[6\nu + 4j_3 + l_\gamma] = 2$ if $k \in \{1, 2, 4, 5, 7\}$.

For each triple $c_j \in C$ we create six matching genotypes $c_j^1, \ldots, c_j^6$, whose 2-entries are:

- $c_j^k[6\nu + 4j] = 2$ if $k \leq 3$.
- $c_j^k[6\nu + 4j + r + 1] = 2$, where $r = k - 1 \pmod 3$.

Note that the construction of the matching genotypes implies that the trivial haplotype and each of the haplotypes that have a single 1-entry in one of the columns $6\nu + 4j + r, r \in \{1, 2, 3\}, j \in \{0, \ldots, \mu - 1\}$ will necessarily be included in any solution to the PH instance.

The construction ensures that if genotypes of different elements share a non-trivial haplotype, then the elements are members of the same triple c_j and the haplotype has a single 1-entry at column $6\nu + 4j$. Also, only the trivial haplotype can be shared between matching genotypes that are not part of the same triple, or between a matching genotype and an element genotype that is

γ_1	2	2	0	0	2	2	2	2
γ_2	2	2	2	2	0	0	2	2
γ_3	2	2	2	2	2	2	0	0
γ_4	0	2	0	0	2	2	2	2
γ_5	0	2	2	2	0	0	2	2
γ_6	0	2	2	2	2	2	0	0
γ_7	0	2	2	2	2	2	2	2

Fig. 3. Gadget for the construction of element genotypes in the proof of Theorem 4.

not a member of the corresponding triple. Finally, the set of non-trivial haplotypes that can be shared by a set of genotypes for a single element γ and the matching genotypes of a triple c_j, where $\gamma \in c_j$ includes: (1) the haplotypes having a single 1-entry at either column $6\nu + 4j$ or $6\nu + 4j + l_\gamma$; and (2) the haplotype that has two 1-entries at columns $6\nu + 4j$ and $6\nu + 4j + l_\gamma$.

We now claim that A admits a phasing of size $15\nu+4\mu-\omega+1$ if and only if C has a matching of size ω. Suppose that C has a matching of size ω. We phase the genotypes of each triple in the matching using the $\mathcal{P}_6$ template shown in Figure 4. We phase the genotypes of elements in each such triple using 4 additional haplotypes using the $\mathcal{P}_5$ template shown in Figure 5. The remaining sets of matching genotypes can be phased using 4 haplotypes each, according to the $\mathcal{P}_4$ template shown in Figure 4. The remaining sets of element genotypes can be phased using 5 haplotypes each, according to the $\mathcal{P}_5$ template shown in Figure 5. Overall, the phasing includes $18\omega + 5 \cdot 3(\nu - \omega) + 4(\mu - \omega) + 1 = 15\nu + 4\mu - \omega + 1$ haplotypes.

Conversely, suppose that A admits a phasing of cardinality $15\nu + 4\mu - \omega + 1$. We let the matching include those triples that share haplotypes with all their elements in this phasing. We first show that any phasing of the set of genotypes of an element, γ, must contain at least 5 non-trivial haplotypes, 4 of which cannot be shared with any other genotype. Furthermore, if the fifth haplotype can be shared then it must contain at least two 1-entries at positions $6\nu + 4j$ and $6\nu + 4j + l_\gamma$, for $\gamma \in c_j$, implying the only genotypes it can be shared with are the genotypes of the triple c_j where $\gamma \in c_j$. We distinguish between three cases:

- If only one haplotype has a 1-entry at position $2i$ then the first three genotypes imply three other haplotypes that must occur in the phasing. All four haplotypes cannot be shared with element genotypes of other elements. A fifth haplotype, satisfying the constraints above, is required to phase the seventh genotype.
- If a single haplotype has a 1-entry at position $2i + 1$ and a 0-entry at position $2i$ then genotypes four through seven imply four additional haplotypes that must occur in the phasing, none of which can be shared with element genotypes.
- If none of the above holds, then there are at least two haplotypes that have a 1-entry at position $2i$ and two haplotypes that have a 0-entry at position $2i$ and a 1-entry at position $2i + 1$. All four haplotypes cannot be shared with element genotypes. A fifth haplotype, satisfying the constraints above, is necessary for phasing genotypes four through seven.

We observe that the set of genotypes of a triple can share at most three haplotypes with the genotype sets of its elements; if less than three haplotypes are shared, then four additional haplotypes are needed for the phasing of this set of genotypes. We conclude that the constructed matching must be of cardinality at least ω, as the trivial haplotype will be included in the optimal solution, 5 non-trivial haplotypes are required for each set of element genotypes, 4 additional non-trivial haplotypes are required for each set not sharing three haplotypes with its elements, and 3 additional non-trivial haplotypes are required for each triple assigned to the matching. ■

$$\left\{\begin{array}{ccc}(1000) & \oplus & (0100)\\ (1000) & \oplus & (0010)\\ (1000) & \oplus & (0001)\\ (0000) & \oplus & (0100)\\ (0000) & \oplus & (0010)\\ (0000) & \oplus & (0001)\end{array}\right\} \overset{\mathcal{P}_4}{\Longleftarrow} \left\{\begin{array}{c}(2200)\\ (2020)\\ (2002)\\ (0200)\\ (0020)\\ (0002)\end{array}\right\} \overset{\mathcal{P}_6}{\Longrightarrow} \left\{\begin{array}{ccc}(0000) & \oplus & (1100)\\ (0000) & \oplus & (1010)\\ (0000) & \oplus & (1001)\\ (0000) & \oplus & (0100)\\ (0000) & \oplus & (0010)\\ (0000) & \oplus & (0001)\end{array}\right\}$$

Fig. 4. Templates for phasing the set of genotypes corresponding to a triple in the proof of Theorem 4. $\mathcal{P}_4$ shows a minimal phasing with 4 non-trivial haplotypes. $\mathcal{P}_6$ shows a phasing with 6 non-trivial haplotypes, three of which can be used can be shared with element genotypes.

$$\left\{\begin{array}{c}(22002222)\\ (22220022)\\ (22222200)\\ (02002222)\\ (02220022)\\ (02222200)\\ (02222222)\end{array}\right\} \overset{\mathcal{P}_5}{\Longrightarrow} \left\{\begin{array}{ccc}(10000000) & \oplus & (01001111)\\ (10000000) & \oplus & (01110011)\\ (10000000) & \oplus & (01111100)\\ (00000000) & \oplus & (01001111)\\ (00000000) & \oplus & (01110011)\\ (00000000) & \oplus & (01111100)\\ (00110000) & \oplus & (01001111)\end{array}\right\}$$

Fig. 5. A template for phasing the set of genotypes of an element in the proof of Theorem 4 using 5 non-trivial haplotypes.

Since the PH problem is NP-hard with respect to clique instances, our main focus in this section is on identifying clique sub-instances for which PH is tractable. We start with several observations on the constraints imposed by a clique instance on the sharing among its genotypes.

Lemma 5: In a $(*, k)$-bounded clique instance every non-trivial haplotype is shared by at most k genotypes.

Proof: Consider a non-trivial haplotype. By definition, such a haplotype must have a 1-entry in some position, and that is consistent with at most k genotypes. ■

Given a clique instance with n genotypes, any solution to it must have at least $L \equiv \frac{\sqrt{8n+9}-3}{2} \sim \sqrt{2n}$ non-trivial haplotypes. To see this, consider a solution with l non-trivial haplotypes. Since the l haplotypes, together with the trivial haplotype, can form at most $l + \binom{l}{2}$ distinct genotypes, we must have $l \geq L$. We now show a lower bound on solutions to $(*, k)$-bounded clique instances, for $k \leq L$.

Lemma 6: For $k \leq L$, any solution to a $(*, k)$-bounded clique instance has cardinality at least $\frac{2n}{k+1} + 1$.

Proof: Consider a solution with l non-trivial haplotypes. Since all n genotypes in the input instance are distinct, the trivial haplotype participates in the phasing of at most l of them in this solution. By Lemma 5, the solution explains at most $l + l(k-1)/2$ genotypes, implying that $l \geq \frac{2n}{k+1}$. The claim follows. ■

Corollary 2: For $(*, k)$-bounded clique instances, the trivial solution yields an approximation ratio of $\frac{k+1}{2}$.

We now present a polynomial algorithm for $(*, 2)$-bounded clique instances. Clearly, an upper bound of $n + 1$ is easy to achieve. By Lemma 6, $\frac{2n}{3} + 1$ is a lower bound on the cardinality of any solution. We shall use the following auxiliary lemma.

Lemma 7: Let G be a $(*, 2)$-bounded clique instance and let g, g', g'' be three genotypes of G such that g and g' share h and g and g'' share $\bar{h}$, where $g = h \oplus \bar{h}$. Then h has 1 in every position in which both g and g' have 2.

Proof: Suppose to the contrary that h has 0 in some position in which both g and g' have 2. Hence, $\bar{h}$ has 1 in that position and, thus, cannot be consistent with g'', since this would imply that the instance is not $(*, 2)$-bounded, a contradiction. ■

Note that for a $(*, 2)$-bounded clique instance, an inference path that starts from a given genotype and a given haplotype is uniquely defined if we terminate its construction upon encountering the trivial haplotype. An inference path that is constructed in this manner is said to *avoid the trivial haplotype*. Now, for a $(*, 2)$-bounded clique instance and a haplotype h, we define a *clique inference path* as follows. If h is consistent with a single genotype g then its clique inference path is the inference path that starts at g and avoids the trivial haplotype. If h is consistent with two genotypes g_1 and g_2, its clique inference path is created by: (1) computing an inference path with respect to each of the two genotypes that avoids the trivial haplotype; (2) merging these paths by adding an edge between g_1 and g_2; and (3) adding an edge between the two other ends of the paths if both paths were terminated at the trivial haplotype. Note that the resulting clique inference path may form a *cycle*. This happens if both paths identify, or both terminate at the trivial haplotype.

Lemma 8: In a $(*, 2)$-bounded clique instance, any non-trivial genotype belongs to at most one clique inference cycle.

Proof: By definition, a clique inference cycle contains at least three genotypes. Let g be a non-trivial genotype and suppose to the contrary that g occurs in two distinct cycles. Let g_a, g_b and g_c, g_d be its neighbors on each of the cycles, respectively. Then there are four haplotypes h_a, h_b, h_c, h_d such that $g = h_a \oplus h_b = h_c \oplus h_d$, $g_a = h_a \oplus \bar{h_a}$, $g_b = h_b \oplus \bar{h_b}$, $g_c = h_c \oplus \bar{h_c}$ and $g_d = h_d \oplus \bar{h_d}$.

Let s be a non-zero position in g. Then w.l.o.g. we can assume that $h_a[s] \neq 0$ and $h_c[s] \neq 0$, implying that g_a and g_c are non-zero at position s. Since the instance is $(*, 2)$-bounded, and since by construction $g \neq g_a$ and $g \neq g_c$, we must have $g_a = g_c$. We further claim that $h_a = h_c$.

Suppose to the contrary that $h_a \neq h_c$. Let i be some position at which the two haplotypes differ and w.l.o.g. $h_a[i] = 1$. Then $h_d[i] = 1$, implying that g, g_a and g_d have a 2-entry at position i. However, $g_a \neq g_d$ since $g_a = g_c$, a contradiction. We conclude that both cycles correspond to the clique inference path of h_a, proving the claim. ■

Lemma 9: The most parsimonious solution for a $(*, 2)$-bounded clique instance that contains no clique inference cycles is of cardinality $n + 1$.

Proof: The existence of such a solution is immediate. Suppose to the contrary that there exists a solution of smaller cardinality. Construct a graph G on the input genotypes with edges connecting genotypes that share a haplotype in that solution. If the trivial haplotype is not used, then every vertex in the graph has degree 2, so G must contain a clique inference cycle, a contradiction. If the trivial haplotype is used, there must be a connected component of G in which the number of genotypes exceeds the number of non-trivial haplotypes that are used to phase them. Hence, this connected component contains a clique inference cycle, a contradiction.

■

Theorem 10: Parsimony can be solved in polynomial time on a $(*, 2)$-bounded clique instance.

Proof: First, observe that in a $(*, 2)$-bounded clique instance, the genotypes comprising a clique inference cycle of length k can be optimally phased using k haplotypes. The algorithm finds all clique inference cycles in the Clark-consistency graph; phases them optimally; and then phases the remaining genotypes using the trivial haplotype and one additional haplotype for each remaining genotype. The correctness of the algorithm follows from Lemmas 8 and 9.

The identification of clique inference cycles relies on Lemma 7, and is done by iterating the following steps until all genotype pairs that share some haplotype have been processed:

(a) Choose two genotypes g_1, g_2 that share some haplotype.

(b) Let h be the haplotype with 1 in position i if and only if $g_1[i] = g_2[i] = 2$.

(c) Construct the clique inference path of h.

(d) If this is a cycle, add the haplotypes found to the optimal solution and remove the genotypes found from consideration. ■

V. BOUNDED TREEWIDTH GRAPHS

A graph G is said to have ***treewidth*** k (cf. [2]) if G admits a cover $\{X_i\}_{i \in I}$ of its vertices such that: (a) $|X_i| \leq k + 1$ for all i; (b) for every edge (g, g') of G, some X_i contains both g and g'; and (c) the sets X_i can be assigned to nodes i of a rooted binary tree $T = (I, F)$ such that if j is on a path between i and k in T then $X_i \cap X_k \subseteq X_j$.

In this section we consider the case when the input instance gives rise to a Clark-consistency graph with bounded treewidth. We shall present a polynomial dynamic-programming algorithm for such graphs on enumerable input instances. We assume that the Clark-consistency graph is connected, as otherwise we can operate on each connected component independently.

Theorem 11: There is a polynomial algorithm for PH on enumerable instances when the Clark-consistency graph has bounded treewidth.

Proof: Since the input instance is enumerable, there are $O(n^c)$ haplotypes that are consistent with any genotype in the input instance (for some constant c). Let G be a Clark-consistency graph of bounded treewidth for the input instance. Thus, G admits a cover $\{X_i\}_{i \in I}$ of its vertices such that a tree T on the sets X_i has the properties described above. We give a dynamic programming algorithm for PH on G. Let r be the root of T. For a node v, let v_1 and v_2 be its two children, and let X_v denote the set of genotypes assigned to this node. We say that a multi-set of haplotypes H *resolves* a node v if $H = \{h_1, \ldots, h_{|X_v|}\}$ and genotype i in X_v is consistent with h_i.

Denote the optimum solution for the sub-instance induced by the genotypes in the subtree rooted at v by $D(v)$. Denote by $D(v,H)$ the optimum solution to this sub-instance for a multi-set H that resolves v.

Clearly, $D(r) = \min_H D(r,H)$ where H ranges over all $O(n^{c(k+1)})$ multi-sets of haplotypes of cardinality $|X_r|$ that resolve r. The following recursive formula can be used to compute $D(r,H)$:

$$D(r,H) = \min_{H_1,H_2}\{D(r_1,H_1) + D(r_2,H_2) + \Delta(r,r_1,r_2,H,H_1,H_2)\}$$

where $H_i, i = 1,2$ resolves r_i and agrees with H on the haplotypes explaining each genotype in $X_r \cap X_{r_i}$. $\Delta(r,r_1,r_2,H,H_1,H_2)$ is a correction factor: let x be the number of haplotypes that are used in phasing $X_{r_1} \cap X_{r_2}$ according to H_1 (or H_2). Let y be the number of haplotypes that are used to phase $X_r \setminus (X_{r_1} \cup X_{r_2})$ according to H. Then $\Delta(r,r_1,r_2,H,H_1,H_2) = y - x$.

For a leaf v at the base of the recursion, $D(v,H)$ is defined as the number of distinct haplotypes in the set composed of the haplotypes in H and their mates (with respect to X_v). Thus, $D(r)$ can be computed using a bottom-up traversal of the tree T in polynomial time. ■

Lemma 12: Let G be the Clark-consistency graph of an enumerable input instance. Any k edges whose removal makes G of bounded treewidth can be used to approximate parsimony to within an additive term of k.

Proof: Suppose we are given a set of k edges, whose removal makes G of bounded treewidth. By removing those edges we can apply the above dynamic programming algorithm to the resulting graph. Since each additional pair of genotypes that share a haplotype can reduce the number of required haplotypes by at most 1, we obtain a solution with at most $opt + k$ haplotypes, where opt is the size of an optimum solution. ■

VI. Bipartite Graphs

In this section we study the parsimony problem when the Clark-consistency graph is bipartite. We note that this implies that each haplotype can be shared by at most two genotypes. Hence, the lower bound on the cardinality of any solution is the number of genotypes n. We prove that parsimony haplotyping on bipartite graphs is hard to approximate even in the case that the longest inference path is of length 2. We complement this result by giving a polynomial algorithm for the case that the longest inference path is of length 1, and an approximation algorithm for paths of length greater than 1.

Theorem 13: Parsimony haplotyping is NP-hard when the Clark-consistency graph is bipartite and the longest inference path is of length 2.

Proof: We reduce from 3DM3. Consider a 3DM3 instance with disjoint sets X, Y, Z containing ν elements each, and a set $C = \{c_0, \ldots, c_{\mu-1}\}$ of μ triples in $X \times Y \times Z$. We construct a PH instance with $3\nu + 3\mu$ genotypes and $6\nu + 5\mu$ SNPs.

For each element $x_i \in X$, $y_i \in Y$, or $z_i \in Z$ we construct one genotype, whose non-zero entries are (see Figure 6):

- $x_i[i] = 2$; $x_i[3\nu + i] = 1$; $x_i[6\nu + 5j] = 2$ for every j such that $x_i \in c_j$.
- $y_i[\nu + i] = 2$; $y_i[4\nu + i] = 1$; $y_i[6\nu + 5j + 1] = 2$ for every j such that $y_i \in c_j$.
- $z_i[2\nu + i] = 2$; $z_i[5\nu + i] = 1$; $z_i[6\nu + 5j + 2] = 2$ for every j such that $z_i \in c_j$.

For each triple $c_j \in C$ we create 3 genotypes, whose non-zero entries are:

- $c^x_j[3\nu+i] = 2$ for every i such that $x_i \in c_j$; $c^x_j[6\nu+5j] = 1$; $c^x_j[6\nu+5j+2] = c^x_j[6\nu+5j+3] = 2$.
- $c^y_j[4\nu + i] = 2$ for every i such that $y_i \in c_j$; $c^y_j[5\nu + i] = 2$ for every i such that $z_i \in c_j$; $c^y_j[6\nu + 5j + 1] = 1$; $c^y_j[6\nu + 5j + 2] = c^y_j[6\nu + 5j + 4] = 2$.
- $c^z_j[5\nu+i] = 2$ for every i such that $z_i \in c_j$; $c^z_j[6\nu+5j+2] = 1$; $c^z_j[6\nu+5j] = c^z_j[6\nu+5j+1] = 2$.

$$\begin{array}{c|ccc|ccc|ccccc}
x_i & 2 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
y_i & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 \\
z_i & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\
\hline
c^x & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 2 & 2 & 0 \\
c^y & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 1 & 2 & 0 & 2 \\
c^z & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 1 & 0 & 0
\end{array}$$

Fig. 6. Gadget for the reduction in the proof of Theorem 13.

The graph is bipartite as the genotypes c_j^z,x_i,y_i can be assigned to one side of the bipartition, and the genotypes z_i, c_j^x, c_j^y can be assigned to the other side. The only possibilities for haplotype sharing between genotypes are: (1) γ with c^γ for $\gamma \in \{x, y, z\}$; and (2) c^x or c^y with c^z. By construction of columns $6\nu + 5j$ and $6\nu + 5j + 1$, if c_j^z shares a haplotype with some z_i, it cannot share its complement with c_j^x or c_j^y. Thus, the longest haplotype inference path has length 2.

Let A be the resulting genotype matrix. We claim that A admits a phasing of size $6\nu + 4\mu - \omega$ if and only if C has a matching of size ω. The proof is similar to that in Theorem 1 using the phasing templates given in Figure 7. ■

$$\left\{\begin{array}{ccc}(00010010010) & \oplus & (00000010100) \\ (00000101100) & \oplus & (00001001001) \\ (00000101100) & \oplus & (00000010100)\end{array}\right\} \overset{\mathcal{P}_4}{\Longleftarrow} \left\{\begin{array}{c}(00020010220) \\ (00002201202) \\ (00000222100)\end{array}\right\} \overset{\mathcal{P}_6}{\Longrightarrow} \left\{\begin{array}{ccc}(00010010000) & \oplus & (00000010110) \\ (00001001000) & \oplus & (00000101101) \\ (00000100100) & \oplus & (00000011100)\end{array}\right\}$$

Fig. 7. The three matching genotypes corresponding to a triple in the proof of Theorem 13 and alternative phasings of these genotypes. $\mathcal{P}_4$ show a minimal phasing with 4 haplotypes, none of which can be shared with the element genotypes. $\mathcal{P}_6$ shows a phasing using 6 haplotypes, 3 of which can be shared with the element genotypes.

Corollary 3: Parsimony haplotyping is APX-hard when the Clark-consistency graph is bipartite and the longest inference path is of length 2.

We note that since a haplotype can be shared by at most two genotypes, any phasing will give a 2-approximation to PH. The following two lemmas improve on this trivial ratio.

When the longest inference path is of length 1, one can reduce PH to a maximum matching problem, giving rise to the following result:

Observation 1: If the length of the longest inference path is 1 then parsimony haplotyping can be optimally solved in polynomial time.

For general bipartite graphs we can use this fact to devise a 1.5-approximation algorithm: (1) Find a maximum matching in the Clark-consistency graph; (2) phase each genotype pair in the matching using a shared haplotype; and (3) arbitrarily phase the remaining genotypes.

Lemma 14: The above algorithm gives a 1.5-approximation for PH on instances that induce a bipartite Clark-consistency graph.

Proof: Consider an instance of PH with a bipartite Clark-consistency graph G. Let m be the size of a maximum matching in G and let n be the number of genotypes. By definition, the solution returned by the approximation algorithm will have size $2n - m$. Let T be an optimum solution to the PH instance and let H be the subgraph of G which contains an edge between two genotypes if and only if they share a haplotype in the solution. Denote by e the number of edges in H, i.e., e is the number of genotype pairs that share a haplotype in this solution. Then $|T| = 2n - e$, and the approximation guarantee is $\frac{2n-m}{2n-e} \leq \frac{2n-m}{2n-2m} \leq \frac{3}{2}$. The first inequality follows from the fact that each vertex has degree at most 2 in H, and the second inequality follows from the fact that $2m \leq n$, and that the worst bound is obtained for $n = m/2$. ■

VII. Conclusions

In this paper we have studied the complexity and approximability of parsimony haplotyping. We have shown that the problem is APX-hard even in very restricted cases. On the positive side, we have introduced a characterization of input instances by the Clark-consistency graphs they induce, and identified classes of these graphs with specific structure of haplotype sharing, which admit polynomial or constant-approximation algorithms.

Common methods for solving parsimony haplotyping include integer programming [12], [3], which is often solved using a branch and bound approach, and direct branch and bound methods [27]. Our results may be of use when incorporated within these branch and bound procedures, by terminating when the examined sub-instance has the characteristics of one of the problems studied here. The sub-instance can then be efficiently solved using the algorithms we have described.

VIII. Acknowledgements

RS was supported by an Alon fellowship. The work of BH and SI was in large part performed while working at Applied Biosystems/Celera Genomics.

References

[1] V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as a perfect phylogeny. A direct approach. *Journal of Computational Biology*, 10(3):323–340, 2003.

[2] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Computing*, 25:1305–1317, 1996.

[3] D. G. Brown and I. M. Harrower. A new integer programming formulation for the pure parsimony problem in haplotype analysis. In *Proceedings of the Fourth International Workshop on Algorithms in Bioinformatics (WABI)*, pages 254–265, 2004.

[4] A. G. Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7(2):111–122, 1990.

[5] L. Excoffier and M. Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):921–927, 1995.

[6] D. Fallin and N.J. Schork. Accuracy of haplotype frequency estimation for biallelic loci, via the expectation-maximization algorithm for unphased diploid genotype data. *American Journal of Human Genetics*, 67(4):947–59, 2000.

[7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W.H. Freeman and Company, 1979.

[8] G. Greenspan and D. Geiger. Model-based inference of haplotype block variation. *Journal of Computational Biology*, 11:493–504, 2004.

[9] D. Gusfield. A practical algorithm for optimal inference of haplotypes from diploid populations. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 183–189, 2000.

[10] D. Gusfield. Inference of haplotypes from samples of diploid populations: Complexity and algorithms. *Journal of Computational Biology*, 8(3):305–324, 2001.

[11] D. Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions (Extended abstract). In *Proceedings of the Sixth Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 166–175, 2002.

[12] D. Gusfield. Haplotyping by pure parsimony. In *Proceedings of the Fourteenth Symposium on Combinatorial Pattern Matching (CPM '03)*, pages 144–155, 2003.

[13] B. V. Halldórsson, V. Bafna, N. Edwards, R. Lippert, S. Yooseph, and S. Istrail. A survey of computational methods for determining haplotypes. In *Computational Methods for SNPs and Haplotype Inference (LNCS 2983)*, pages 26–47, 2004.

[14] E. Halperin and E. Eskin. Haplotype reconstruction from genotype data using imperfect phylogeny. *Bioinformatics*, 20:1842–1849, 2004.

[15] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.

[16] M. E. Hawley and K. K. Kidd. HAPLO: A program using the EM algorithm to estimate the frequencies of multi-site haplotypes. *Journal of Heredity*, 86:409–411, 1995.

[17] Y.-T. Huang, K.-M. Chao, and T. Chen. An approximation algorithm for haplotype inference by maximum parsimony. In *Proc. ACM Symposium on Applied Computing*, pages 146–150, 2005.

[18] E. Hubbell. Finding a maximum parsimony solution to haplotype phase is NP-hard. Personal communication, 2001.

[19] G. Lancia, M. C. Pinotti, and R. Rizzi. Haplotyping populations by pure parsimony. Complexity, exact and approximation algorithms. *INFORMS Journal on Computing*, 16:348–359, 2004.

[20] H. Lin, Z.-F. Zhang, DQ.-F. Zhang, D.-B. Bu, and M. Li. A note on the single genotype resolution problem. *J. Comput. Sci. and Technol.*, 19:254–257, 2004.

[21] J. C. Long, R. C. Williams, and M. Urbanek. An E-M algorithm and testing strategy for multiple-locus haplotypes. *American Journal of Human Genetics*, 56(2):799–810, 1995.

[22] T. Niu, Z. S. Qin, X. Xu, and J. S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *American Journal of Human Genetics*, 70:157–169, 2002.

[23] N. Patil et al. Blocks of limited haplotype diversity revealed by high resolution scanning of human chromosome 21. *Science*, 294:1719–1723, 2001.

[24] E. Petrank. The hardness of approximations: gap location. *Computational Complexity*, 4:133–157, 1994.

[25] R. Sharan, B.V. Halldórsson, and S. Istrail. Islands of tractability for parsimony haplotyping. In *Proc. IEEE Computational Systems Bioinformatics Conference (CSB'05)*, 2005. In press.

[26] M. Stephens, N. J. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics*, 68:978–989, 2001.

[27] L. Wang and Y. Xu. Haplotype inference by maximum parsimony. *Bioinformatics*, 19:1773–1780, 2003.