Toward an understanding of haplotype inference by pure parsimony

Daniel G. Brown Ian M. Harrower

School of Computer Science, University of Waterloo,
Waterloo ON N2L 3G1 Canada
{browndg,imharrow}@cs.uwaterloo.ca
Technical Report CS-2005-27

Abstract

Haplotype inference by pure parsimony (HIPP) is known to be NP-Hard. Still, many algorithms have shown great success in solving HIPP instances on simulated data. In this paper, we present two phase transitions that illustrate why such problem instances can be classified as easy to solve. Let G be a genotype matrix generated by randomly pairing k distinct haplotypes arising from a perfect phylogeny. We show that with high probability, if G contains $(\frac{1+\epsilon}{2})k \log k$ distinct genotypes, the size of the optimal HIPP solution is the rank of G, which is computable in polynomial time. We also show that if the haplotypes have a column with minor allele frequency α, and G has $(\frac{2+\epsilon}{\alpha})k \log k$ genotypes, then with high probability, we can find an optimal set of haplotypes in polynomial time. We strengthen our results by showing that there are equivalent phase transitions when the population of haplotypes has non-uniform frequencies. Finally, we show that for a population of haplotypes of size p generated under the standard coalescent model of evolution, if $\omega(\log p)$ columns are generated, then with high probability, we will have a column with minor allele frequency α, and our phase transition bound will apply.

1 Introduction

Haplotype inference is the process of attempting to identify the sequences of parental chromosomes that have given rise to a diploid population. In the past few years, this problem has become increasingly important, as researchers attempt to identify variations responsible for inherited diseases. A particular focus has been on *single nucleotide polymorphisms*, or SNPs, single points of the genome that exist in two different common alleles.

Several haplotype inference problems have been characterized. The simplest to describe is the *haplotype inference by pure parsimony* problem, introduced by Gusfield [9]. The goal in this problem is to identify a smallest haplotype set that can explain all of the genotypes in a collection of individuals. Gusfield's original paper gives a biological intuition to the problem; the problem is also interesting from a purely combinatorial perspective.

However, the problem is NP-hard [9], and the only known polynomial-time approximation algorithms have exponentially bad performance guarantees [15]. Yet, in practice, the problem is surprisingly easy, especially when applied to synthetically generated test instances arising

from standard models of evolution. Several authors have developed algorithms based on integer programming or branch-and-bound that have shown fast performance [9, 4, 18, 5].

Here, we give some explanation of why these instances are easy to solve. Our primary results are two theorems, in Sections 3 and 4. Theorem 1 shows that if the k true ancestral haplotypes come from a perfect phylogeny, and the population consists of at least $(\frac{1+\epsilon}{2})k \log k$ distinct members obtained by randomly pairing the haplotypes (where $\epsilon > 0$), the minimum number of haplotypes to explain the genotypes will be the linear algebraic rank of the input genotype data matrix. That is, we can obtain the *size* of the optimal haplotype set in polynomial time by computing the rank, but we cannot not identify the actual haplotypes.

Our second theorem, in Section 4, gives a constructive algorithm. If all conditions of Theorem 1 hold, and in addition, there is a mutation with minor allele frequency at least α, and the size of the input genotype set is at least $(\frac{2+\epsilon}{\alpha})k \log k$, with high probability, we can find the smallest haplotype set to explain the genotype matrix in polynomial time. We extend our theorems to the case where haplotypes are not sampled uniformly from a pool, in Section 4.2.

In Section 4.4, we give an estimate of the number of mutations for which data is needed to guarantee such a common mutation will be found, under the coalescent model of evolution. We show that to obtain a guarantee that the probability of a common mutation is at least $1 - \epsilon$ from a pool of p haplotypes, the number of sites needed is $\log p$ times a constant depending only on α and ϵ.

Our results begin to explain the ease of haplotype inference in practice, despite its theoretical hardness: if interesting instances of the problem can usually be solved in polynomial time, this may indicate hidden structure that helps explain why an algorithm with theoretically exponential runtime is, in practice, surprisingly fast.

2 Background and related work

We begin with an explanation of the basic domain in which we are working. We then briefly review existing work on haplotype inference by pure parsimony, and on a phase transition in the data generation model used in this domain.

Haplotype inference and notation The input to a haplotype inference algorithm is the *genotype matrix*, G. Each of its n rows represents the diploid genotype, g_i, of a population member p_i, while the m columns represent m polymorphic sites in the genome. Thus, $G(i, j)$ is the diploid genotype of population member p_i at site s_j. (The same m sites are tested for all n population members.)

Following typical practice [9], we assume there are only two alleles, 0 and 1, at each site. Under this assumption, there are three choices for $G(i, j)$: $G(i, j) = 0$ if both parent chromosomes of p_i have allele 0 at s_j, $G(i, j) = 1$ at heterozygous positions where one parent has each allele, and $G(i, j) = 2$ if both have allele 1. (**Note**: this is **not** the standard notation for this problem, which exchanges the meanings of 1 and 2. We explain our use of this notation shortly.)

Haplotype inference consists of identifying a 0/1 *haplotype matrix*, H, to explain G. The k rows of H each represent a chromosome and its alleles at the m sites. Genotype g_i is explained by H if there exist two rows (possibly the same) of H with sum g_i. We can represent the

pairing of the rows of H by an $n \times k$ *pairing matrix*, Π, where row r_i of Π has value 1 in the two columns corresponding to the two parent haplotypes of g_i, and 0 in the other columns. (If genotype g_i is explained by two copies of the same haplotype, then row r_i of Π has one entry with value 2 and all others 0.) In this formulation, haplotype inference consists of finding a 0/1 haplotype matrix H and a valid pairing matrix Π such that $G = \Pi \cdot H$. (The simplicity of this formulation explains our notational choice; the idea of this formulation is due to He and Zelikovsky [12].)

Pure parsimony In the *haplotype inference by pure parsimony* (HIPP) problem, introduced by Gusfield [9], we seek the smallest set of haplotypes to explain G. For us, this corresponds to finding the smallest H such that a proper pairing matrix Π exists where $G = \Pi \cdot H$. This problem is NP-hard [9], and the only approximation algorithms for it, due to Lancia *et al.* [15], have exponential guarantees on their performance that limit their practical usefulness.

Yet in practice, many instances of this problem have been easy to solve. Gusfield [9] identified an integer linear programming formulation for the problem that, though theoretically exponential in size, solved very quickly in practice. Halldórsson *et al.* found a polynomial-sized IP formulation for the problem [10], which we independently identified and extended with further inequalities [4]; our experiments with that formulation also demonstrated that even on large instances, an IP solver could find the optimal solution. In subsequent work, we hybridized the polynomial-sized formulation and Gusfield's exponential formulation, and were able to solve even larger problem instances [5]. At the same time, He and Zelikovsky [12] gave fast heuristics for a related problem that often work on moderately large instances. Thus, we are left with a quandary: why are the instances we are solving so easy in practice?

Perfect phylogeny haplotyping To identify why the HIPP problem is often easy in practice, we briefly examine the details of how problem instances might be generated for a synthetic haplotype inference problem.

The simplest model for evolution of k parental haplotypes is a perfect phylogeny, in which all m characters evolve according to the structure of a rooted phylogenetic tree. Without loss of generality, we assume that at every sampled site s_j, the common ancestor of all haplotypes had allele 0. The site is assigned to a single edge of the tree, which represents the the mutation of that site. Leaves descendant from that edge have allele 1 at site s_j; other leaves have allele 0. Each site only mutates once on the tree; there are no back-mutations. A matrix H satisfying this condition is a PPH matrix (for *perfect phylogeny haplotype*). For ease in the rank bound of Section 3, we also include a single non-polymorphic site s_0, where all haplotypes have value 1; this has no effect on any solution, since any pairing of two such rows will have the same value, 2, for s_0.

To generate n genotypes, we must pair the haplotypes. The null model for this process is random pairing. That is, each genotype results from pairing two haplotypes sampled with replacement. This corresponds exactly to n edges being picked from a random multigraph model, where every edge (i, j) has probability $2/k^2$, and every loop (i, i) has probability $1/k^2$.

A phase transition for haplotyping? In 2003, Chung and Gusfield [6] examined the number of distinct PPH solutions to a genotype matrix G, obtained by randomly pairing the k members of a haplotype pool obtained by a perfect phylogeny. They noticed a phase transition in the

number of PPH solutions: when $n \ll k \log k$, there are many PPH solutions, while when $n \gg k \log k$, there is typically only one PPH solution. Inspired by this observation (which is similar to what we noticed about the HIPP problem), Cleary and St. John [7] studied the structure of random pairing graphs; they used the coupon collector lemma to prove that for data generated from a PPH instance, if there are $o(k \log k)$ population members, with high probability there is not a unique PPH solution. (They do not prove that above this bound, there is a unique solution with high probability, but they do give experiments that support this claim.)

The Min-PPH problem Bafna *et al.* [1] have considered the problem of finding the most parsimonious explanation H for a genotype matrix G, subject to the restriction that H is a PPH matrix.

This problem, called Min-PPH, is also NP-hard [1]. However, a polynomial-time algorithm by Bafna *et al.* [2], called DPPH, can return an implicit representation of all PPH solutions for a given genotype matrix G in $O(nm^2)$ time. Subsequently, one can examine these solutions in time that is is polynomial in the input size and proportional to the number of PPH solutions. Thus, if the number of PPH solutions is polynomial, the runtime of this approach, using DPPH, followed by enumeration, is as well.

However, it is not clear that the Min-PPH problem is necessarily relevant to HIPP: it is possible that the most parsimonious explanation for a matrix G that was derived from a PPH matrix H may actually be a smaller matrix H' that does not satisfy the PPH condition. Our first theorem, in Section 3, shows that if there are at least $(\frac{1+\epsilon}{2})k \log k$ genotypes, this is unlikely: the HIPP solution will be the same as the Min-PPH solution.

3 A first limit theorem for HIPP

Our first limit theorem for HIPP gives a bound on the number of genotypes, above which it is easy to predict the optimal number of haplotypes.

Theorem 1. *Let G be a genotype matrix derived by randomly pairing k distinct haplotypes from a perfect phylogeny. Suppose G has at least $(\frac{1+\epsilon}{2})k \log k$ distinct rows, for some constant $\epsilon > 0$, and that k^* is the smallest number of haplotypes that explain all of G. Then $k = k^* = \text{rank}(G)$ almost surely as $k \to \infty$.*

Proof. We first consider the matrix Π. It is the node-edge incidence matrix of the pairing multigraph. A standard result in graph theory shows that such a matrix is of full rank if all of its connected components are non-bipartite [17]. Thus, one way to demonstrate that the rank of Π is k is to show that the graph is connected and contains a triangle.

If the graph has ℓ distinct rows, then it results from at least $\ell - k$ random non self-loop pairings. As such, it contains a subgraph G' from the standard random graph model $G(k, \ell - k)$, with k nodes and $\ell - k$ edges, with each possible edge equally likely. Two theorems about random graphs show that if $\ell \geq (\frac{1+\epsilon}{2})k \log k$, G' is connected almost surely, and it is non-bipartite almost surely, as $k \to \infty$ [3]. As such, Π is of rank k almost surely as $k \to \infty$.

Next, consider the rank of H. Each distinct haplotype h_i corresponds to a different leaf in the phylogenetic tree, and has value 1 at positions corresponding to mutations on the path

from leaf to root. Consider two neighbouring leaves i and j. Since they have distinct sequences, at least one of i or j has a mutation on the edge from their common ancestor to it; suppose it is i. Then h_i has a 1 found in no other haplotype, and is thus clearly linearly independent of all other haplotypes. We can remove h_i and repeat this process for all k haplotypes, and thus show H is of full rank; the existence of the column s_0 of all 1s mentioned in Section 2 prevents a row of all 0s being the last row left, and ensures the matrix is of rank k, not $k-1$. We have assumed that the common ancestor of all haplotypes has a 0 at all sites except s_0, but elementary column operations allow us to complement any column without affecting the rank.

Since $G = \Pi \cdot H$, Π is almost surely of rank k, and H is of rank k, then the rank of G is also, almost surely, k. However, the rank of G is also a lower bound on the minimum number of rows k^* for any H^* such that $G = \Pi^* \cdot H^*$. (This rank bound was noted previously by Kalpakis and Namjoshi [14].)

Thus, the minimum number of haplotypes to explain G is at least k almost surely. Moreover, Π and H give a way of explaining G with exactly k haplotypes. Hence, our theorem is proved: the minimum number of haplotypes to explain G is its rank, almost surely, as $k \to \infty$. □

It is important to note that the condition that the input matrix comes from a perfect phylogeny is only used to establish that the input haplotype matrix is of full rank. The theorem holds if the input haplotypes are distinct and are the rows of a matrix that is full rank almost surely as $k \to \infty$.

This theorem does not construct H; it merely gives its likely size. We have not succeeded in using this theorem to make an efficient HIPP algorithm.

4 A second limit theorem, for slightly larger populations

For instances of the problem with a constant factor more haplotypes than the bound of Theorem 1, and with a mutation that is "common" (which we define below), we can prove a constructive result. We do this by connecting to the Min-PPH problem, which was described in Section 2, above. Our main result is the following constructive theorem.

Theorem 2. *Let G be derived from randomly pairing k distinct haplotypes, represented by the haplotype matrix H. Suppose that the haplotypes result from a perfect phylogeny, and suppose also that there exists a column of H (without loss of generality, s_1) where at least αk haplotypes have value 0 at s_1 and at least αk of the haplotypes have value 1, for some $\alpha > 0$. Then if G includes at least $(\frac{2+\epsilon}{\alpha})k \log k$ distinct genotypes, then the HIPP problem can be solved in polynomial time almost surely as $k \to \infty$.*

For example, if $\alpha = \frac{1}{4}$, so there is a SNP with minor allele frequency at least 25%, Theorem 2 says that if the number of distinct rows of G is at least $(8+\epsilon)k \log k$, the optimal set of haplotypes can be found with high probability.

4.1 Proof of the second limit theorem

We will prove Theorem 2 through several steps.

Lemma 1. *If the conditions of Theorem 2 hold, then almost surely, the set of haplotypes H used to generate genotype matrix G is an optimal HIPP solution.*

$$a)\ \begin{matrix} 1 & 1 \\ 1 & x \\ y & 1 \end{matrix} \qquad b)\ \begin{matrix} 1 & 1 \\ 0 & 0 \\ 2 & 2 \end{matrix} \qquad c)\ \begin{matrix} 1 & 1 \\ 2 & 0 \\ 0 & 2 \end{matrix}$$

Figure 1: Patterns of 3×2 submatrices which cause an edge to be added between the vertices representing the sites in $D(G)$. The values x and y are each either 0 or 2.

Proof. The number of distinct population members is greater than $(\frac{1+\epsilon}{2})k \log k$, so Theorem 1 applies, and the rank of G almost surely equals k, the number of distinct haplotypes in the set H used to generate G. Any solution must use at least rank(G) members. Thus, H is an optimal HIPP solution for G. □

Corollary 1. *With high probability, the optimal solution to a Min-PPH problem whose input matrix satisfies the conditions of Theorem 2 is also optimal for HIPP on the same input matrix.*

Proof. Lemma 1 allows us to restrict our search to classes of haplotype matrices that include the optimal solution H. Since H satisfies a perfect phylogeny, one smallest PPH solution for G will be H, which is also optimal for HIPP. □

Corollary 1 allows us to restrict our search to PPH solutions. Lemma 2 shows there exists only one, with high probability; Corollary 2 shows it can be found in polynomial time. This will complete the proof of Theorem 2.

Lemma 2. *Given a genotype matrix G satisfying the conditions of Theorem 2, with high probability there exists only one set of haplotypes that satisfies the perfect phylogeny condition and can generate G.*

Proof. We prove the lemma via a property of the DPPH algorithm of Bafna *et al.* [2]. This algorithm constructs a graph with one vertex for each column of G. The main result we use is that the number of PPH solutions for G is 2^{c-1}, where c is the number of connected components in a specific subgraph [2] of this graph. We show that if G satisfies the conditions of Theorem 2, then with high probability, $c = 1$, and there is only one PPH solution.

Consider a graph $D(G)$, which has one vertex for each column in the genotype matrix G. We add an edge between two vertices s and s' if there exits a row g of G with value 1 in columns s and s', and the resolution of g at sites s and s' is restricted by the perfect phylogeny condition.

More precisely, we connect s and s' if we can find three genotypes g_1, g_2 and g_3 such that the 3×2 submatrix of G induced by these genotypes and sites has one of the forms from Figure 1. In each of these forms, one of the possible resolutions of sites s and s' would violate the perfect phylogeny condition, so the possible space of PPH solutions is restricted. The restriction is such that the total number of PPH solutions for G equals 2^{c-1}, where c is the number of connected components of $D(G)$ [2].

To show that $D(G)$ is almost surely connected, we will show that almost surely, there is an an edge between each node and the node for the site s_1 with the common mutation. Let e_1 be the tree edge containing the mutation at site s_1. We partition the set of haplotypes into two classes: C, containing all haplotypes descendant of e_1, and B, containing all haplotypes not descendant of e_1. For any site s, let e_s be the tree edge where the mutation at s occurs, and A_s be the set of all haplotypes below e_s in the tree. When considering the random pairings to make

genotypes, we use $\langle X, Y \rangle$ to denote a pairing of a haplotype from class X with a haplotype from class Y.

Consider a site s. We consider a few cases on s, depending on whether e_s is below e_1 or not, and on the size of A_s. First, suppose e_s is not below e_1. If A_s has fewer than $\alpha k/2$ haplotypes, we will have an edge between s and s_1 in $D(G)$ if the events $\langle A_s, C \rangle$, $\langle A_s, B \setminus A_s \rangle$ and $\langle B \setminus A_s, C \rangle$ occur. These events produce the rows (1 1), (1 0), and (0 1), a submatrix of type (a) in Figure 1. For a random pairing, each event has probability at least $\frac{\alpha}{2k}$. If $|A_s| > \alpha k/2$, the events $\langle A_s, C \rangle$, $\langle A_s, A_s \rangle$ and $\langle C, C \rangle$ produce the rows (1 1), (2 0) and (0 2), which are of type (c) in Figure 1. Again, all events have probability at least $\frac{\alpha}{2k}$.

Suppose instead that e_s is below e_1. If $|A_s| < \alpha k/2$, events $\langle A_s, B \rangle$, $\langle A_s, C \setminus A_s \rangle$ and $\langle B, C \setminus A' \rangle$ give the rows (1 1), (1 2), and (0 1), of form (a), while if $|A_s| \geq \alpha k/2$, events $\langle A_s, B \rangle$, $\langle A_s, A_s \rangle$ and $\langle B, B \rangle$ give rows (1 1), (2 2), and (0 0), of form (b). In both cases, the events all have probability at least $\frac{\alpha}{2k}$.

Therefore, for each column s, three events, each with probability at least $\frac{\alpha}{2k}$ occurring is sufficient to connect s and s_1. This totals less than $6k$ events, since there are at most $2k - 2$ distinct columns in a perfect phylogeny with k leaves. The coupon collector lemma [16] shows that after $(\frac{2+\epsilon}{\alpha})k \log k$ random pairings, the probability that a needed event has not yet occurred is less than $6(k^{-\epsilon/2})$.

Thus, if the conditions of Theorem 2 are satisfied, then with high probability, the graph $D(G)$ is connected and the DPPH results of Bafna *et al.* [2] show that there exists a unique PPH solution for G. □

Corollary 2. *A Min-PPH instance G satisfying the conditions of Theorem 2 can be solved in polynomial time with high probability.*

Proof. The algorithm DPPH of Bafna *et al.* [2] gives a representation of all PPH solutions for a given genotype matrix G in polynomial time, and allows their enumeration in time polynomial in the input matrix size and proportional to the number of PPH solutions. Since Lemma 2 shows that there is a unique PPH solution with high probability, it can be recovered in polynomial time. □

4.2 Non-uniform sampling of a haplotype pool

Theorems 1 and 2 apply to genotypes derived from random pairings of haplotypes, each of identical frequency. However, various population genetic mechanisms may cause some haplotypes to be more common in a population than others; we will see this, for example, in Section 4.4, when we discuss sequences derived from the coalescent model.

Theorems 1 and 2 can be extended to this case as well: given a pool of p haplotypes that satisfy the perfect phylogeny condition and that are not necessarily unique, the results of Theorems 1 and 2 apply, except that the bound depends on p, the haplotype pool size, as well as k, the number of unique haplotypes. This change makes sense: if, for example, we randomly sample from a pool that includes k haplotypes, some of them very rare, the random pairing graph may include components consisting solely of rare haplotypes, which may still be bipartite well past when there are $\frac{1}{2}k \log k$ edges.

Theorem 3. *Suppose we are given a haplotype matrix H with p rows, possibly not unique, but with k distinct rows, and suppose that the haplotypes can be derived from a perfect phylogeny.*

If we create a genotype matrix G by pairing random rows of H, and if at least $(\frac{1+\epsilon}{2})p\log p$ different pairings of rows in H give rise to G, then the size of the optimal solution to the HIPP instance G is the rank of G, almost surely as $p \to \infty$.

Proof. The proof is similar to the proof for Theorem 1. If we have at least $f(p)$ unique genotypes, then we include a subgraph from the standard random graph model $G(p, f(p) - p)$; again, it is non-bipartite and connected almost surely as $p \to \infty$, if $f(p) = (\frac{1+\epsilon}{2})p\log p$, so the pairing matrix Π has full column rank p almost surely. Since $G = \Pi \cdot H$, and H has rank k, G has rank k as well; since this is a lower bound on the size of the optimal HIPP instance, and can be achieved with the k distinct haplotypes of H, the optimal value of the HIPP instance is rank(G), as desired. □

Theorem 2 can also be extended to this domain.

Theorem 4. *Suppose we are given a haplotype matrix H with p rows, possibly not unique, and with k distinct rows. Suppose also that H is derivable from a perfect phylogeny, and that there exists a column of H such that at least αp of the rows have value 1 in that column, and at least αp of the rows have value 0 in that column, for $\alpha > 0$. If we generate a genotype matrix G by randomly pairing rows of H, and if G arises from at least $\max(\frac{1+\epsilon}{2}p\log p, \frac{2+\epsilon}{\alpha}p\log k)$ distinct pairings of members of H, then we can solve the HIPP problem in polynomial time almost surely as $k \to \infty$ (and consequently $p \to \infty$).*

Proof. The proof follows the structure of the proof for Theorem 2. Since the number of different pairings is large enough, Theorem 3 allows us to restrict ourselves to PPH solutions while still remaining optimal for HIPP.

We create the same graph, $D(G)$, as before: nodes correspond to sites (distinct columns of G), and edges to pairs of sites, with each edge arising from three rows of G. There are k unique haplotypes, so there are at most $2k-2$ possible columns; thus the graph has at most $2k-2$ nodes. To connect each node to the one for the site with minor allele frequency at least α, three pairings suffice, each with probability at least $\frac{\alpha}{2p}$. The coupon collector lemma again applies: if the number of pairings is at least $(\frac{2+\epsilon}{\alpha})p\log k$, for some $\epsilon > 0$, then almost surely as $k \to \infty$, there is only one PPH solution, the one that generated G, and the algorithm described in Section 4.1 finds it in polynomial time. □

4.3 Applicability to Small Populations

The previous theorems are asymptotic results that require that the number of unique haplotypes approaches infinity. However, to evaluate algorithms synthetic instances with relatively few population members are often used. We evaluate the applicability of Theorem 4 to small populations of haplotypes. We use Hudson's program `ms` [13] to generate p haplotypes with $2p$ sites. The haplotypes are not necessarily distinct. We pair the haplotypes randomly to generate n distinct genotypes. We generate 200 instances for each n and test for a unique PPH solution using Ding, Filkov and Gusfield's LPPH [8], and test if the rank of the input genotype matrix equals the number of unique haplotypes. For each p we find the first value of n (in increments of 5) that passes this test in all 200 tests. The results are summarized in table 1. We can see that for p evaluated it was sufficient to generate $p\log p$ genotypes, which is much lower than the estimated $8p\log p$ that we would predict for $\alpha = \frac{1}{4}$.

Number of haplotypes p	30	50	75	100	150	200
Sufficient number of genotypes n	95	165	290	385	650	885

Table 1: The smallest number of genotypes n for which all 200 trials passed the rank and PPH tests. The values are all less than $p \log p$, but are growing faster than linearly.

4.4 A bound on finding a common mutation in a standard model

Our theorems now show that with high probability, if there exists a common mutation studied (with minor allele frequency at least α), we will likely be able to solve HIPP if there are at least $(\frac{2+\epsilon}{\alpha})p \log k$ distinct genotypes. How often does such a mutation occur?

Certainly, it is unlikely that one would be studying a population if all mutations in the population were rare. However, we can give a partial answer to this question probabilistically, in the standard infinite-sites constant-population coalescent model from population genetics.

Our results show that to guarantee that with probability $1 - 4\epsilon$, there is a site chosen with minor allele frequency at least α, the number of sites needed is only $\log p$ times a constant depending only on α and ϵ. Our bounds are coarse, but again help to explain the high success in solving synthetic HIPP instances.

An introduction to the coalescent model Here, we give a brief discussion of this model, focusing on details we need; for full detail, see Hartl and Clark [11].

The coalescent model describes the descent of a population under neutral evolution. We use it to generate rooted trees with p leaves, where each leaf represents a haplotype.

The model generates the tree topology as follows: Initially, there is one population member, which eventually has p descendants. The model begins with a bifurcation of that member into 2 distinct taxa. The number of descendants of the first of these is uniformly distributed over $[1, \ldots, p-1]$; the second taxon has the remaining descendants. A taxon is *active* if it has more than one eventual descendants; if it has only 1 descendant, it undergoes no further bifurcations.

There will be $p-2$ more bifurcations. For each, one active taxon is chosen uniformly (and independently of all previous choices) and bifurcated; the number of descendants of the new offspring are uniformly chosen so each has at least one, maintaining the proper total. After $p-1$ bifurcations, a tree with p leaves has been built. (We note that typical descriptions of this model give the process as operating from the leaves to the root; we have used this equivalent description because it makes our analysis easier.)

The tree edges are all assigned lengths, which correspond to time between bifurcation events (the time unit is not relevant to our needs). The time between the ith and $i+1$st bifurcation events is an exponentially distributed random variable r_i, with mean $\frac{1}{i(i+1)}$, while the time between the last bifurcation and the end of the process is a random variable r_{p-1}, exponentially distributed with mean $\frac{1}{(p-1)p}$; all lengths are independent. An edge of the phylogenetic tree between the ith and jth bifurcation has length $\sum_{k=i+1}^{j} r_k$; if it starts with the ith bifurcation and ends in a leaf, its length is $\sum_{k=i+1}^{p-1} r_k$.

To generate a haplotype matrix, we assume that the root of the tree has value 0 at every character, that every infinitesimal region of the tree is equally likely to have a mutation, that every site only undergoes at most one mutation (an *infinite sites* model), and that all characters are independent. Under these assumptions, for a given polymorphic site s_i, the probability that

the mutation in that site is on an edge e of the phylogenetic tree (and thus, s_i has value 1 only in leaves descendant from that edge) is proportional to the length of the edge.

Infinite-site constant-populaton coalescent models are a standard population genetics model: they are used, for example, in the program `ms` [13], which has been used by Gusfield [9] and by us [4, 5] to generate HIPP problem test instances, by randomly pairing the resultant haplotypes. More complicated models of the same sort allow population size change and recomination to be modeled.

Common mutations We now connect the coalescent model to our HIPP theorems. Since the model generates p not necessarily unique haplotypes, we can apply Theorem 4. Yet, how often do we have a studied polymorphic site where the minor allele is found in at least αp haplotypes?

Theorem 5. *Suppose that we produce p haplotypes by an infinite-site constant-population coalescent model. For any $\epsilon > 0$, if we choose at least*

$$\log(1/\epsilon)((2\log p + \log(1/\epsilon))(p/(p-1))\left(\frac{\epsilon^{2/(2\alpha-1)} + \epsilon^{1/(2\alpha-1)}}{\log(1/(1-\epsilon))}\right)$$

independent polymorphic sites, with probability at least $1 - 4\epsilon$, then we will have chosen a site with minor allele frequency at least α. In particular, if the number of sites is $\omega(\log p)$, then such a site is chosen almost surely as $p \to \infty$.

Note that this bound has a weak dependency on p: it can be bounded by $\log p$ times a constant depending on ϵ and α .

We prove Theorem 5 by studying the probability a single site meets our needs, which is the fraction of the total tree edge length in edges with between αp and $(1-\alpha)p$ descendants. (In what follows, we call these "good" edges.)

First, consider the total edge length. Between the ith and the $i+1$st bifurcation, there are $i+1$ lineages. Each contributes r_i to the total edge length. The variable r_i is exponentially distributed with mean $\frac{1}{i(i+1)}$; multiplying it by $(i+1)$ yields an exponential random variable with mean $1/i$. Thus, the total edge length is the sum of i independent exponentially distributed random variables q_i for $i = 1 \ldots p-1$, each with mean $1/i$.

Lemma 3. *Let X be the total edge length in a coalescent tree with p leaves. For any $\delta > 1$, $X < \delta \log p$ almost surely as $p \to \infty$. In particular, $Pr[X > \delta \log p] < p^{\gamma+1-\delta+\delta p^{-\gamma}}$ for any $\gamma > 0$; if we use $\gamma = 1$, we can show that $Pr[X > t_{p,\epsilon}] < \epsilon$, where $t_{p,\epsilon} = (2\log p + \log(1/\epsilon))(p/(p-1))$.*

Proof. A weaker version of this lemma can be proved using Chebyshev's inequality: $E[X]$ is between $\log(p-1)$ and $1 + \log(p-1)$, and $\mathrm{Var}[X] = \sum_{i=1}^{p-1} \mathrm{Var}(q_i) = \sum_{i=1}^{p-1} \frac{1}{p^2}$, which converges to the constant $\pi^2/6$. Chebyshev's inequality will bound the probability that X is much greater than $\log p$.

However, we can give a sharper Chernoff-style tail bound. We are interested in $Pr[X > \delta \log p] = Pr[e^{tX} > p^{\delta t}]$ for any $t > 0$; by Markov's inequality, this is at most $p^{-\delta t} E[e^{tX}]$. Since the q_i are independent, the second factor equals $\prod_{i=1}^{p-1} E[e^{tq_i}]$. The expectation $E[e^{tq_i}]$ equals $\frac{i}{i-t}$, assuming $t < 1$, so the probability $Pr[X > \delta \log p]$ is at most $p^{-\delta t} \prod_{i=1}^{p-1} \frac{i}{i-t}$. If we set t to $1 - 1/p^\gamma$ for any $\gamma > 0$, we can upper bound the bad probability by $p^{\gamma+1-\delta+\delta/p^\gamma}$ as desired. □

Next, we show that there is likely a good edge reasonably high in the tree.

Lemma 4. *Consider a coalescent tree with p leaves, and assume $0 < \alpha < 1/3$. The probability that the top of a good edge, with between αp and $(1-\alpha)p$ descendant leaves, exists in the tree at or before the ℓth bifurcation is at least $1 - \ell^{2\alpha-1}$. Given an $\epsilon > 0$, with probability at least $1-\epsilon$, there is a good edge whose start is at or above level $\ell^*_{\alpha,\epsilon} = \epsilon^{1/(2\alpha-1)}$.*

Proof. At each layer, there exists a lineage with the most descendants. If it has fewer than $(1-\alpha)p$ descendants, we have already seen a good edge. To see this, consider the first time this happens: we divided a value greater than $(1-\alpha)p$ into two parts, both smaller than $(1-\alpha)p$. Since $\alpha < 1/3$, one is at least αp.

Thus, we can concern ourselves with bifurcations on the lineage with the most descendants. At step i in the coalescent process, this lineage has probability at least $1/i$ of being chosen to bifurcate; it may be more if there are inactive lineages. If it bifurcates, the probability of a good edge being produced is at least $1 - 2\alpha$. Since all bifurcations are independent, we can upper bound the probability of no good edges occurring by level ℓ by $\prod_{i=1}^{\ell}(1 - \frac{1-2\alpha}{i}) < \prod_{i=1}^{\ell} e^{\frac{2\alpha-1}{i}} < e^{(2\alpha-1)\log \ell} = \ell^{2\alpha-1}$. The bound as a function of the probability $1-\epsilon$ of a good edge at or above level $\ell^*_{\alpha,\epsilon}$ is easily shown by arithmetic. □

Corollary 3. *With probability at least $1 - 2\epsilon$, the total length of all good edges is at least $f_{\epsilon,\alpha} = \frac{\log(1/(1-\epsilon))}{\ell^*_{\alpha,\epsilon}(\ell^*_{\alpha,\epsilon}+1)}$.*

Proof. By Lemma 4, there is a good edge by level $\ell^*_{\alpha,\epsilon}$ with probability at least $1-\epsilon$. The length of an edge is the sum of a sequence of exponentially distributed variables, one of which has mean at least $\frac{1}{\ell^*_{\alpha,\epsilon}(\ell^*_{\alpha,\epsilon}+1)}$. The probability that this variable takes value at least $\log(1/(1-\epsilon))$ times its mean is $1-\epsilon$, so with probability at least $1-2\epsilon$, the total good edge length is at least $f_{\epsilon,\alpha}$. □

Now, we can finish the proof of Theorem 5.

Proof. With probability at least $1-\epsilon$, the total tree length is less than $t_{p,\epsilon}$, and with probability at least $1-2\epsilon$, the total good tree edge length is at least $f_{\epsilon,\alpha}$. Thus, with probability at least $1-3\epsilon$, the probability that the mutation at any single polymorphic site is on a good edge is at least $\frac{f_{\epsilon,\alpha}}{t_{p,\epsilon}}$. After $\log(1/\epsilon)\frac{t_{p,\epsilon}}{f_{\epsilon,\alpha}}$ random sites, the probability of no good mutation is less than ϵ. Thus, with probability at least $1-4\epsilon$, in a coalescent tree with p leaves, if we choose $m = \log(1/\epsilon)\frac{t_{p,\epsilon}}{f_{\epsilon,\alpha}}$ columns, we have chosen a column with minor allele frequency greater than α. □

Our theorem gives an unpleasantly complex bound on the minimum number of columns needed to satisfy the conditions of Theorem 4. However, the important feature is that for every α, if we want to guarantee that the probability of a column with minor allele frequency at least α is at least $1-\epsilon$, it suffices to have $\log p$ columns times a constant depending only on ϵ and α; that is, the number of needed columns grows only with $\log p$.

5 Conclusion

We have concerned ourselves in this work with explaining why finding optimal solutions to the haplotype inference by pure parsimony problem seems easier than one might predict given its NP-hardness. Our answer is that the problem is actually fairly easy for a standard population genetic model, where haplotypes are generated by a perfect phylogeny, and then randomly paired to create genotypes, as long as there are a moderate number of genotypes.

In this domain, if the number of unique haplotypes is k, and pairings are independent, and uniformly chosen from all pairings, we have shown that with high probability, $(\frac{1+\epsilon}{2})k \log k$ unique genotypes suffice to make it likely that the rank of the genotype matrix is exactly the optimal number of haplotypes. Further, if there exists a common mutation among the haplotypes, whose minor allele frequency is at least α, then $(\frac{2+\epsilon}{\alpha})k \log k$ genotypes suffice to make it likely that this optimal set of haplotypes is retrievable in polynomial time.

In Section 4.2, we showed that this result extends to the case where the haplotype pool includes some repeated haplotypes: there, we obtain a bound that depends on p, the size of the haplotype pool from which we are randomly pairing, as well as the number of unique haplotypes, which may be smaller.

In Section 4.4, we extended these theorems to the specific case of data generated by the coalescent model often studied in population genetics; this is relevant, for example, for data generated with Hudson's popular `ms` package [13]. We give a bound, in terms of the number of sites needed to analyze, before it is likely that we can find a common mutation of minor allele frequency α in the pool under study. Our bound is logarithmic in the size of the population pool, with complicated dependency on the frequency α and the desired probability bound.

In summary, we can give modest bounds on the number of distinct rows of the genotype matrix and the number of columns, above which, with high probability, the optimal solution to HIPP can be found in polynomial time.

Acknowledgements This research has been supported by the Natural Science and Engineering Research Council of Canada, through a Discovery Grant to D.B. and a Postgraduate Scholarship to I,H. We would like to thank Katherine St. John for sending us a copy of her paper with Sean Cleary [7].

References

[1] V. Bafna, D. Gusfield, S. Hannenhalli, and S. Yooseph. A note on efficient computation of haplotypes via perfect phylogeny. *J. Comp. Biol.*, 11:858–866, 2004.

[2] V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogeny: A direct approach. *J. Comp. Biol.*, 10:323–340, 2003.

[3] B. Bollobás. *Random Graphs.* Cambridge Press, 2nd edition, 2001.

[4] D. G. Brown and I. M. Harrower. A new integer programming formulation for the pure parsimony problem in haplotype analysis. In *Proceedings of WABI 2004*, pages 254–265, 2004.

[5] D. G. Brown and I. M. Harrower. A new formulation for haplotype inference by pure parsimony. Technical Report CS-2005-03, University of Waterloo School of Computer Science, March 2005.

[6] R. H. Chung and D. Gusfield. Empirical exploration of perfect phylogeny haplotyping and haplotypers. In *Proceedings of COCOON 2003*, pages 5–19, 2003.

[7] S. Cleary and K. St. John. Analyses of haplotype inference algorithms. 2005. Manuscript under review.

[8] Zhihong Ding, Vladimir Filkov, and Dan Gusfield. A linear-time algorithm for the perfect phylogeny haplotyping (pph) problem. In *Proceedings of RECOMB 2005*, pages 585–600, 2005.

[9] D. Gusfield. Haplotype inference by pure parsimony. In *Proceedings of CPM 2003*, pages 144–155, 2003.

[10] 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: DIMACS/RECOMB Satellite Workshop*, volume 2983 of *LNCS*, pages 26–47, 2004.

[11] D. Hartl and A. Clark. *Principles of Population Genetics.* Sinauer, 3rd edition, 1997.

[12] J. He and A. Zelikovsky. Linear reduction for haplotype inference. In *Proceedings of WABI 2004*, pages 242–253, 2004.

[13] R. R. Hudson. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, 2002.

[14] K. Kalpakis and P. Namjoshi. Haplotype phasing using semidefinite programming. Techical report TR CS-04-10, University of Maryland Baltimore County, July 2004.

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

[16] R. Motwani and P. Raghavan. *Rondomized Algorithms.* Cambridge Press, 1995.

[17] C. Van Nuffelen. On the incidence matrix of a graph. *IEEE Transactions on Circuits and Systems*, 23(9):572–572, Sep 1976.

[18] L. Wang and Y. Xu. Haplotype inference by maximum parsimony. *Bioinformatics*, 19(14):1773–1780, 2003.