

HAPLOFREQ—Estimating Haplotype Frequencies Efficiently

ERAN HALPERIN¹ and ELAD HAZAN²

ABSTRACT

A commonly used tool in disease association studies is the search for discrepancies between the haplotype distribution in the case and control populations. In order to find this discrepancy, the haplotypes frequency in each of the populations is estimated from the genotypes. We present a new method HAPLOFREQ to estimate haplotype frequencies over a short genomic region given the genotypes or haplotypes with missing data or sequencing errors. Our approach incorporates a maximum likelihood model based on a simple random generative model which assumes that the genotypes are independently sampled from the population. We first show that if the phased haplotypes are given, possibly with missing data, we can estimate the frequency of the haplotypes in the population by finding the *global* optimum of the likelihood function in *polynomial time*. If the haplotypes are not phased, finding the maximum value of the likelihood function is NP-hard. In this case, we define an alternative likelihood function which can be thought of as a relaxed likelihood function. We show that the maximum relaxed likelihood can be found in polynomial time and that the optimal solution of the relaxed likelihood approaches asymptotically to the haplotype frequencies in the population. In contrast to previous approaches, our algorithms are guaranteed to converge in polynomial time to a global maximum of the different likelihood functions. We compared the performance of our algorithm to the widely used program PHASE, and we found that our estimates are at least 10% more accurate than PHASE and about ten times faster than PHASE. Our techniques involve new algorithms in convex optimization. These algorithms may be of independent interest. Particularly, they may be helpful in other maximum likelihood problems arising from survey sampling.

Key words: phasing, haplotypes, SNPs, expectation maximization.

1. INTRODUCTION

MOST OF THE GENETIC VARIATION AMONG DIFFERENT PEOPLE can be characterized by single nucleotide polymorphisms (SNPs) which are mutations at a single nucleotide position that occurred once in human history and were passed on through heredity. Characterization of the generic variation is an important

¹International Computer Science Institute, 1947 Center St., Suite 600, Berkeley, CA.

²Princeton University, Department of Computer Science, Princeton, NJ 08540.

research tool for trait association and disease association in particular. In order to understand the structure of this variation, we need to be able to determine the *haplotypes* of individuals, or which nucleotide base occurs at each position for each chromosome. The effort to characterize human variation, currently a major focus for the international community, will be a tremendous undertaking requiring obtaining the haplotype information from a large collection of individuals from diverse populations (NIH, 2002).

As opposed to haplotypes, the genotype gives the bases at each SNP for both copies of the chromosome, but loses the information as to the chromosome on which each base appears. Unfortunately, many sequencing techniques provide the genotypes and not the haplotypes. Haplotype analysis has become increasingly common in genetic studies of human disease. However, many of these methods rely on phase information, that is, the haplotype information versus the genotype information. Phase can be inferred by genotyping family members of each subject, but this has its downsides because of logistic and budget issues. Alternatively, laboratory techniques such as long range PCR or chromosomal isolation have been also used (Patil *et al.*, 2001; Michalatos-Beloin *et al.*, 1996), but these are often costly and are not suitable for large scale polymorphism screening. As an alternative to those technologies, many computational methods have been developed for phasing the genotypes (e.g., Clark [1990], Gusfield [2000, 2001 2002], Lancia *et al.* [2001], Stephans *et al.* [2001], Niu *et al.* [2001], Halperin and Eskin [2004], and Kimmel and Shamir [2004]).

Even though much of the attention was aimed at finding the haplotype phase, it is usually crucial to estimate correctly the haplotype frequencies in the population and not necessarily to phase the individual genotypes. For instance, in disease association studies, it is usually more informative to find the discrepancies between the control haplotype distribution and the cases haplotype distribution, than to find the phase of the haplotypes. The most likely estimation for the haplotype distribution in a population can be viewed as a weighted average over all possible phasing options. Therefore, finding the most likely phase and counting the number of occurrences of each haplotype could be used as a crude estimate for the haplotype distribution. On the other hand, in some cases, this crude estimate may be inaccurate, and more accurate frequency estimators are needed.

There are, however, a few EM-based (expectation maximization) algorithms that directly estimate the haplotype frequencies (Excoffier and Slatkin, 1995; Fallin and Shork, 2000; Hawley and Kidd, 1995; Long *et al.*, 1995). These methods use a likelihood function based on the underlying assumption that the Hardy–Weinberg equilibrium holds (that the two haplotypes of an individual are independently drawn from the haplotype distribution in the population). In particular, those methods try to find a *haplotype* distribution which maximizes the probability of observing genotypes in the given sample, under the assumption of Hardy–Weinberg equilibrium.

One of the main drawbacks in all previous methods is that there is no guarantee that the algorithm converges to a global maximum or that the algorithm converges in polynomial time. Both the convergence of the EM algorithm to a global optimum and its running time are heavily affected by the starting point of the algorithm which is usually a “reasonable” guess or a random point.

We present a method called HAPLOFREQ which aims to overcome the above limitations of previous approaches. Similarly to previous approaches, we use a likelihood function model. Our approach is different from previous approaches in the following aspects. First, we use an algorithm which is provably guaranteed to run efficiently and to find the haplotype distribution assuming that the number of samples is large enough and assuming a uniform error model. Second, we consider two different likelihood functions, one that assumes Hardy–Weinberg equilibrium and another that does not. The latter is used in order to find the *genotype* distribution given missing data or the *haplotype* distribution given phased haplotypes with missing data. For instance, the phased haplotypes are given when sequencing chromosome X in men or when sequencing the genome of certain organisms that are either haploid or have a short life span.¹

In the case where the Hardy–Weinberg equilibrium holds, the maximum likelihood function is a multinomial of very high degree. In order to find the maximum value of this multinomial, we relax the problem by allowing the variables to be n -dimensional vectors instead of real numbers. We then use convex

¹For example, in *Drosophila*, the phased haplotypes can be obtained by breeding.

programming methods which involve linear constraints, multinomial functions, and positive semidefinite constraints in order to find the maximum value of the relaxed problem. This relaxed objective function can be thought of as an alternative likelihood function since we show that the maximum value of the relaxed function approaches asymptotically the haplotypes frequencies in the population.

We measured the performance of our algorithm over various datasets and compared it to the widely used program PHASE (Stephans *et al.*, 2001). We found that our algorithm is consistently more accurate and much faster than PHASE. In Section 6, we describe our experiments and their results.

2. ESTIMATING HAPLOTYPE FREQUENCIES

One of the most natural tools in disease association studies is the search for discrepancies in the allele distribution between the cases and the controls. A natural extension of this tool is the search for discrepancies between the haplotype distribution in each of the populations. In particular, the haplotype frequency is calculated from the samples of each of the populations, and a statistical test (e.g., chi squared) is performed in order to assess whether the haplotype distributions of the two populations are identical. If the distributions are significantly different, then the region is likely to be correlated with the disease.

In order to estimate the haplotype frequencies in a population, a geneticist would sample a set of n individuals from the population. Throughout the paper, we assume that these n individuals are independently sampled from a large population. Each sample consists of a genotype, which is the information of the two copies of the chromosome in each base. The haplotype information therefore has to be derived from the genotype information. Furthermore, the sequenced data usually contains some missing data, which adds another complexity to the problem. In this paper, we focus on estimating the haplotype frequencies from the genotype data, with missing data. Our model and algorithms have natural extensions for other types of noise, such as sequencing errors.

2.1. A maximum likelihood approach

In order to formalize the above scenario, we first need to set some notations and definitions. A *complete haplotype* is a binary string of length k . The values 0 and 1 correspond to the mutation and the wild type alleles. A *partial haplotype* is a string over $\{0, 1, *\}^k$. The character “*” corresponds to an unknown value (missing data).

We denote a genotype by a string over $\{0, 1, 2, *\}^k$, where 0,1 correspond to homozygous sites (i.e., the bases of the mother’s chromosome and the father’s chromosomes are the same), the value 2 corresponds to a heterozygous position, that is, a position where the mother chromosome carries a different base than the father chromosome and “*” corresponds to unknown values for both haplotypes. For a given genotype g or haplotype h , we denote by $g(i)$ ($h(i)$), respectively, its value in the i -th coordinate.

We say that a genotype $g \in \{0, 1, 2, *\}^k$ and a pair of complete haplotypes $h^1, h^2 \in \{0, 1\}^k$ are *compatible* if for every position i , if $g(i) \in \{0, 1\}$ then $h^1(i) = h^2(i) = g(i)$ and if $g(i) = 2$ then $h^1(i) \neq h^2(i)$.

For a genotype g , we define $\mathcal{C}(g)$ to be the set of pairs of haplotypes that are compatible with g . We assume that the genotypes admit a Hardy–Weinberg equilibrium, that is, that the two haplotypes of each individual are independently picked from the distribution of haplotypes in the population.

Let \mathcal{P} be a distribution over the set of all possible complete haplotypes of length k . We denote by $p(h)$ the probability assigned to the haplotype h by \mathcal{P} . We consider the following likelihood function (Excoffier and Slatkin, 1995) of a set of partial genotypes \mathcal{G} and a distribution \mathcal{P} :

$$\mathcal{L}(\mathcal{G}, \mathcal{P}) = \prod_{g \in \mathcal{G}} \sum_{(h_1, h_2) \in \mathcal{C}(g)} p(h_1)p(h_2). \quad (1)$$

The function $\mathcal{L}(\mathcal{G}, \mathcal{P})$ is simply the probability of observing the genotypes \mathcal{G} in a random sample of the population under Hardy–Weinberg equilibrium, given that the distribution of complete haplotypes in

the population is \mathcal{P} and that the distribution of missing data in a genotype g does not depend on the contents of g .

When the sample size approaches infinity, the maximum likelihood is attained when \mathcal{P} is the actual distribution of haplotypes in the population.² Therefore, it is only natural to aim for finding the distribution \mathcal{P} which maximizes the likelihood and to estimate the distribution of haplotypes in the population as \mathcal{P} . Previous methods (Excoffier and Slatkin, 1995; Fallin and Shork, 2000; Stephens *et al.*, 2001) use expectation maximization (EM) in order to find the maximum likelihood. When using EM, both the running time and the convergence to a global maximum depend on the starting point. In particular, these algorithms may be exponential, and they may give a nonoptimal solution, even when the number of samples is large. In Section 4, we introduce an alternative approach which is guaranteed to converge to a global optimum of another likelihood function $\mathcal{L}_2(\mathcal{G}, \mathcal{P})$. We further show in Section 4.1 that \mathcal{L} and \mathcal{L}_2 have the same asymptotic behavior under Hardy–Weinberg.

2.2. Working with phased data

In some cases, we are given the phased genotypes, possibly with missing data. For instance, some sequencing techniques provide the haplotypes and not the genotypes. In haploid organisms, or in diploid organisms with short life span such as drosophila,³ we can get the phased haplotypes. It is therefore interesting to estimate the haplotype frequencies given a sample of haplotypes with missing data. This approach may also be useful to estimate the *genotype* frequencies, given a sample of genotypes with missing data. The latter may be particularly important when there are departures from Hardy–Weinberg equilibrium in the underlying genotype distribution.

In order to formalize the above scenario, we need to introduce a few more notations and definitions. We say that a partial haplotype $h_1 \in \{0, 1, *\}^k$ is *consistent* with a complete haplotype $h_2 \in \{0, 1\}^k$ if they share the same values whenever $h_1(i) \neq *$. Given a partial haplotype h , we define $\mathcal{C}(h)$ to be the set of complete haplotypes that are consistent with h .

As before, let \mathcal{P} be a distribution over the set of all possible complete haplotypes of length k . Given the set of partial haplotypes \mathcal{H} , the likelihood of \mathcal{P} is given by

$$\mathcal{L}(\mathcal{H}, \mathcal{P}) = \prod_{h \in \mathcal{H}} \sum_{h' \in \mathcal{C}(h)} p(h').$$

The function $\mathcal{L}(\mathcal{H}, \mathcal{P})$ is simply the probability of observing the partial haplotypes \mathcal{H} in a random sample of the population, given that the distribution of complete haplotypes in the population is \mathcal{P} and that the distribution of missing data in a haplotype h does not depend on the contents of h .

Again, in order to estimate the haplotype frequencies, we find the distribution \mathcal{P} that maximizes the likelihood $(\mathcal{H}, \mathcal{P})$. In Section 3 we introduced an efficient polynomial time algorithm that finds the global maximum of $(\mathcal{H}, \mathcal{P})$. This is quite surprising given that we essentially find a maximum point of a polynomial of potentially high degree. In general, finding an extremum of a polynomial is an intractable problem.

3. ESTIMATING HAPLOTYPE FREQUENCIES FROM A PHASED SAMPLE

In this section, we introduce an algorithm which estimates the haplotype frequencies in a population given a sample of phased haplotypes with missing data.

Formally, given a set \mathcal{H} of n partial haplotypes, we are interested in finding a distribution \mathcal{P} which maximizes the function $\mathcal{L}(\mathcal{H}, \mathcal{P})$, which is given in the previous section. Thus, finding the distribution of

²This is true under some reasonable assumptions on the distribution of the missing data.

³In diploid organisms the haplotype data is found through breeding.

maximum likelihood can be done by solving the following mathematical programming problem:

$$\begin{aligned} & \text{Maximize} && \prod_{h \in \mathcal{H}} \sum_{h' \in \mathcal{C}(h)} p(h') \\ & \text{s.t.} && \sum_{h \in \{0,1\}^k} p(h) = 1 \\ & && p(h) \geq 0, \quad h \in \{0,1\}^k \end{aligned}$$

We will use the following definition in order to simplify the notations.

Definition 1. Given a partial haplotype $h \in \{0, 1, *\}^k$ and a set of haplotypes $S = \{h_1, \dots, h_m\} \subseteq \{0, 1\}^k$, define the compatibility vector of h with respect to S as a vector $A_h \in \{0, 1\}^m$ such that $A_h(i) = 1$ if $h_i \in \mathcal{C}(h)$ and $A_h(i) = 0$ otherwise.

Note that in practice the values of k are relatively small and the set of possible haplotypes is limited to a reasonable size. A typical value for k is in the range of 10–50, and there are typically at most a few hundreds of possible haplotypes, that is, haplotypes that are compatible with one of the genotypes.

Using this definition, the maximum likelihood formulation above is equivalent to solving the following problem:

Definition 2. FREQUENCY ESTIMATION OF PHASED GENOTYPES.

Input: A matrix $A \in \{0, 1\}^{n \times m}$ consisting of n row vectors $\{A_1, \dots, A_n\} \in \{0, 1\}^m$

Goal: Find a vector $\vec{p} \in \mathfrak{R}_+^n$, such that:

1. $\sum_{i=1}^m p_i = 1$; $\forall i \ p_i \geq 0$
2. Let $\vec{q} \stackrel{\text{def}}{=} A \cdot \vec{p}$. Then the following quantity is maximized: $f(\vec{p}) = \prod_{i=1}^n q_i$

3.1. Algorithms for FREQUENCY ESTIMATION OF PHASED GENOTYPES

We first prove that the problem is solvable in polynomial time via the ellipsoid method (Grötschel *et al.*, 1988).

Theorem 1. FREQUENCY ESTIMATION OF PHASED GENOTYPES is solvable in polynomial time.

Proof. We prove that the problem is in \mathcal{P} by providing a separation oracle that can be used with Khachiyan’s ellipsoid algorithm (Khachiyan, 1980) to provide a solution.

Given some vector $\vec{p} \in \mathfrak{R}_+^n$, let $\vec{q} = A \cdot \vec{p} \in \mathfrak{R}_+^m$, define the function

$$g_p(\vec{y}) \stackrel{\text{def}}{=} (A \cdot \vec{y})_1 q_2 \dots q_n + b \left(\sum_{i=2}^n \frac{(A \cdot \vec{y})_i}{q_i} \right)$$

and the corresponding hyperplane

$$H_p \stackrel{\text{def}}{=} \{\vec{x} \in \mathfrak{R}_+^n \mid g_p(\vec{x}) \geq n \cdot b\}.$$

Claim 1. Given a point $\vec{x} \in \mathfrak{R}_+^m$ for which $f(\vec{x}) = a < b$, the hyperplane H_p is a separating hyperplane with respect to \vec{x} . That is, it has \vec{x} on one side and all points \vec{z} such that $f(\vec{z}) \geq b$ on the other side.

Proof. First, notice that

$$\begin{aligned} g_p(\vec{p}) &= (A \cdot \vec{p})_1 q_2 \dots q_n + b \left(\sum_{i=2}^n \frac{(A \cdot \vec{p})_i}{q_i} \right) \\ &= q_1 q_2 \dots q_n + b \left(\sum_{i=2}^n \frac{q_i}{q_i} \right) \\ &= a + (n-1)b < nb. \end{aligned}$$

Now consider any point $\vec{z} \in \mathfrak{R}_+^m$ for which $f(\vec{z}) \geq b$. This implies that $\prod_{i=1}^n (A\vec{z})_i \geq b$, which implies $(A\vec{z})_1 \geq \frac{b}{\prod_{i=2}^n (A\vec{z})_i}$. Therefore,

$$\begin{aligned} g_p(\vec{z}) &= (A\vec{z})_1 q_2 \dots q_n + b \left(\sum_{i=2}^n \frac{(A\vec{z})_i}{q_i} \right) \\ &\geq \frac{b}{\prod_{i=2}^n (A\vec{z})_i} \cdot q_2 \dots q_n + b \left(\sum_{i=2}^n \frac{(A\vec{z})_i}{q_i} \right) \\ &= b \cdot \left[\prod_{i=2}^n \frac{q_i}{(A\vec{z})_i} + \sum_{i=2}^n \frac{(A\vec{z})_i}{q_i} \right]. \end{aligned}$$

Denote $c_i = \frac{(A\vec{z})_i}{q_i}$. Then we have $g_p(\vec{z}) \geq b \cdot [\prod_{i=2}^n \frac{1}{c_i} + \sum_{i=2}^n c_i]$. From symmetry, this function is minimized when $\forall i$ $c_i = c$ for some $c > 0$. So we get $g_p(\vec{z}) \geq b \cdot [\frac{1}{c^{n-1}} + (n-1)c]$. This in turn is minimized for $c = 1$ (left as an exercise for the reader), and therefore

$$g_p(\vec{z}) \geq b \cdot \left[\frac{1}{c^{n-1}} + (n-1)c \right] = nb.$$

Hence, all such vectors \vec{z} for which $f(\vec{z}) \geq b$ are on the other side of the hyperplane H_p than \vec{p} itself. ■

Given this separation oracle, the ellipsoid method can be used to find the optimal vector \vec{p} by binary search on the values of b , to within any needed precision. ■

Given Theorem 1, we can deploy the ellipsoid algorithm with the separation oracle devised above to solve FREQUENCY ESTIMATION OF PHASED GENOTYPES without assumption of Hardy–Weinberg equilibrium. Alternatively, Theorem 1 provides proof that the corresponding mathematical program is convex; therefore, more efficient interior point methods can be applied (Nesterov and Nemirovskii, 1994).

Both approaches have drawbacks: the ellipsoid method suffers from poor performance in practice for many applications. The general interior point framework does not take into account the specific structure of the problem at hand and is rather cumbersome to analyze rigorously.

We proceed to provide an efficient combinatorial algorithm that approximates the solution to FREQUENCY ESTIMATION OF PHASED GENOTYPES to within any required (constant) precision parameter. This algorithm,

```

Procedure HAPLOFREQ( $\varepsilon$ )
 $\vec{p} \leftarrow \vec{1} \cdot \frac{1}{m}$ 
 $\forall i$  set  $q_i \leftarrow A_i \vec{p}$ 
 $\vec{\delta} \leftarrow \text{FINDDELTA}(p, q, A)$ 
while  $\sum_i \frac{A_i \vec{\delta}}{q_i} \geq \varepsilon$  do
    Update  $p$  to be:  $\vec{p} \leftarrow \vec{p} + \frac{\tau^2 \varepsilon}{8n} \vec{\delta}$ 
     $\vec{\delta} \leftarrow \text{FINDDELTA}(p, q, A)$ 
return  $\vec{p}$ 

```

FIG. 1. Algorithm HAPLOFREQ.

HAPLOFREQ, finds for every $\varepsilon > 0$ a $(1 + \varepsilon)$ -approximate solution in time polynomial in the input size and $\frac{1}{\varepsilon}$. The algorithm is interior-point based, tailored to the specific mathematical program at hand. As we shall see in the next section, the rather simple analysis allows us to generalize it to the case of genotypes.

Let the input matrix be $A \in \{0, 1\}^{n \times m}$. Denote $\vec{q} = A \cdot \vec{p}$ and denote the solution vector (optimal probabilities assigned to the haplotypes) by \vec{o} . Also define $\vec{w} = A \cdot \vec{o}$ as the “weights” vector of the optimum solution. Let f be the objective function; that is, $f(\vec{x}) = \prod_i (A_i \vec{x})$.

It is easy to see that we can always obtain an objective value of $f(\vec{p}) \geq (\frac{1}{n})^n$ by picking a nonempty column from each row and then assigning the uniform distribution over all columns picked. Alternatively, we can obtain an initial value of $f(\vec{p}) \geq (\frac{1}{m})^n$ by the uniform distribution over all columns (assigning all probabilities to be $\frac{1}{m}$).

Let $\tau = \min_{p_i > 0} p_i$. Note that every nonzero p_i or q_i value is at least τ . We will show later that one can assume that $\tau \geq 1/m^2$. In practice, τ is normally a constant in the order of 0.001.

Our algorithm is a hill-climbing algorithm. We start from one of the trivial solutions above and then make a series of improvement steps until the required performance guarantee is reached. In each improvement step, we amend the current vector of probabilities \vec{p} to $\vec{p}' = \vec{p} + \vec{\delta}$, so as to improve the overall value. The algorithm, called HAPLOFREQ, which takes as an input a precision parameter ε , is given in Fig. 1.

The only missing part in the description of HAPLOFREQ is the way we find the vector δ . This part is done in the procedure FINDDELTA. We need to make sure that the procedure FINDDELTA finds a vector $\vec{\delta}$ such that $\vec{p} + \frac{\tau^2 \varepsilon}{8n} \vec{\delta}$ is an improved solution to the problem.

Definition 3. Define an ε -good vector with respect to a current solution \vec{p} as a vector $\vec{\delta}$ that satisfies

$$\sum_{i=1}^m \delta_i = 0 \quad ; \quad 0 \leq \delta_i + p_i \leq 1 \quad ; \quad \sum_{i=1}^n \frac{A_i \vec{\delta}}{q_i} \geq \varepsilon.$$

The procedure FINDDELTA returns an ε -good vector if one exists. Note that FINDDELTA can be implemented using linear programming, but this is not very efficient. The following combinatorial method runs in $O(nm + m \log m)$ time.

Procedure FINDDELTA($\mathbf{p}, \mathbf{q}, \mathbf{A}$)

Let $\vec{\alpha}$ such that $\forall i \cdot \alpha_i = (\vec{1} \cdot A)_i / q_i$

Suppose w.l.o.g that $\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_m$ (o/w sort $\vec{\alpha}$)

Set $\delta_m = 1 - p_m$

Set $\forall i < m \cdot \delta_i = -p_i$

return $\vec{\delta}$

Lemma 1. The procedure FINDDELTA finds an ε -good vector for the largest possible ε and can be implemented in time $\tilde{O}(nm)$. Furthermore, it returns a vector $\vec{\delta}$ such that $|A_i \cdot \vec{\delta}| \leq 2$.

(Here and in the rest of the paper, the \tilde{O} notation is used to suppress polylogarithmic factors.)

Proof. The vector returned by FINDDELTA obviously satisfies the second of the conditions of an ε -good vector.

As for the first condition, note that

$$\sum_{i=1}^m \delta_m = - \sum_{i < m} p_i + (1 - p_m) = 1 - \sum_{i=1}^m p_i = 0.$$

In addition, we claim that the $\vec{\delta}$ returned maximizes $\sum_{i=1}^n \frac{A_i \vec{\delta}}{q_i}$ under the first two conditions. This follows from the fact $\sum_{i=1}^n \frac{A_i \vec{\delta}}{q_i} = \vec{\alpha}^T \cdot \vec{\delta}$ and the definition of $\vec{\delta}$. ■

In the rest of this section, we prove the following theorem:

Theorem 2 (Main). *For any constant $\varepsilon > 0$, the algorithm HAPLOFREQ(ε) finds a $(1 + \varepsilon)$ -approximate solution in polynomial time.*

To prove this theorem, we first prove that we can always find an ε -good vector if our current solution is not an e^ε -approximate solution. We then show that using an ε -good vector we can improve our current solution and that polynomially many improvements suffice to obtain a $(1 + \varepsilon)$ -approximate solution.

Lemma 2. *If $\frac{OPT}{ALG} = \frac{f(\vec{o})}{f(\vec{p})} \geq e^\varepsilon$, then there exists an ε -good vector $\vec{\delta}$.*

Proof. The optimal solution gives rise to a natural vector $\delta := \vec{o} - \vec{p}$. It obviously satisfies the first two conditions above, and as for the last,

$$\sum_{i=1}^n \frac{A_i \vec{\delta}}{q_i} = \sum_{i=1}^n \frac{w_i - q_i}{q_i} \geq n \cdot \sqrt[n]{\prod_{i=1}^n \frac{w_i}{q_i}} - n \geq n \cdot \sqrt[n]{e^\varepsilon} - n = n \cdot (e^{\varepsilon/n} - 1) \geq \varepsilon$$

where the first inequality follows from the arithmetic and geometric mean inequality and the last inequality is true since $e^x > x + 1$. ■

Lemma 2 shows that if we are still far from the optimal solution, then there is at least one ε -good vector. Since one such vector exists, FINDDELTA is guaranteed to provide such a vector. We now show that the resulting improvement step brings us closer to the optimum.

Lemma 3. *Let $\varepsilon > 0$, and let $\vec{\delta}$ be an ε -good vector with respect to \vec{p} . As before, let $\tau = \min_{p_i > 0} p_i$. Let $p' := p + \sigma \delta$ where $\sigma = \frac{\tau^2 \varepsilon}{8n}$. Then $\frac{f(\vec{p}')}{f(\vec{p})} \geq e^{\varepsilon \sigma / 2}$.*

Proof. Denote $c_i := \frac{A_i \vec{\delta}}{q_i}$. By Lemma 1, we know that $|A_i \vec{\delta}| \leq 2$, and therefore $|c_i| \leq \frac{2}{\tau}$ (since $q_i \geq \tau$). Hence,

$$\begin{aligned} \log \left(\frac{f(\vec{p}')}{f(\vec{p})} \right) &= \log \left(\prod_{i=1}^n \frac{A_i(\vec{p} + \sigma \vec{\delta})}{q_i} \right) = \sum_{i=1}^n \log \frac{A_i(\vec{p} + \sigma \vec{\delta})}{q_i} = \sum_{i=1}^n \log(1 + \sigma c_i) \\ &\geq \sum_{i=1}^n [(\sigma c_i) - (\sigma c_i)^2] \geq \sigma \varepsilon - \sigma^2 \sum_{i=1}^n c_i^2 \geq \sigma \varepsilon - \frac{4n\sigma^2}{\tau^2} \geq \frac{\varepsilon \sigma}{2}, \end{aligned}$$

where the approximation to the logarithm holds since $|\sigma c_i| \leq \frac{1}{2}$. ■

Now we can prove Theorem 2:

Proof (Theorem 2). Let \vec{p}_i be the vector obtained by the algorithm after i improvement steps, and let $a_i = \log(f(\vec{p}_i))$. Let \vec{o} be an optimal solution, and let $opt = \log(f(\vec{o}))$. By Lemma 3, $a_{i+1} \geq a_i + \frac{\sigma}{2}(opt - a_i)$, where $\sigma = \frac{\tau^2 \epsilon}{8n}$, as long as $opt - a_i \geq \epsilon$. Therefore, as long as $opt - a_i \geq \epsilon$, we get that

$$opt - a_{i+1} \leq \left(1 - \frac{\tau^2 \epsilon}{8n}\right) (opt - a_i).$$

Since we can start from a solution of value at least n^{-n} and the optimal solution is bounded by 1, we have that $opt - a_0 \leq n \log n$. Therefore, after $r = \tilde{O}\left(\frac{n}{\tau^2 \epsilon}\right)$ iterations, we find a solution such that $a_r \geq opt - \epsilon$. ■

Note that our analysis of the running time of HAPLOFREQ so far had a polynomial dependence on the precision parameter τ . The following lemma shows that the running time is indeed polynomial in the size of the problem (or rather that we may assume that τ is larger than a polynomial).

Lemma 4. For each solution \vec{p} throughout the algorithm, it holds that $\min_i q_i \geq \frac{1}{m^2}$.

Proof. The initial \vec{p} satisfies $\min_i q_i \geq p_1 \geq \frac{1}{m^2}$. Suppose that after some local improvement there exists some coordinate with $q_i < \frac{1}{m^2}$. This implies the existence of a coordinate $0 < p_i < \frac{1}{m^2}$, denoted p_1 . In addition, since \vec{p} is a distribution, there is always one coordinate $p_j > \frac{1}{m}$, which we denote by p_2 .

Let $c = p_1 + p_2 > \frac{1}{m}$. Given values \vec{p} , we optimize over the value of p_1 , while keeping c and the rest of \vec{p} constant. We denote $x = p_1$. Maximizing the objective as a function of x is equivalent to maximizing the following expression:

$$f(x) = \prod_{i \in S_1} (x + y_i) \cdot \prod_{i \in S_2} (c - x + y_i),$$

where $S_1 = \{k \in [m] \mid A_{k1} = 1, A_{k2} = 0\}$ is the set of partial haplotypes where p_1 appears and p_2 does not appear, and $S_2 = \{k \in [m] \mid A_{k2} = 1, A_{k1} = 0\}$ is the set of all partial haplotypes where p_2 appears and p_1 does not. Furthermore, the terms y_i are independent of x (they depend on the other p_i).

One can verify that the expression is minimized when $|S_1| = 1$ and $|S_2| = m - 1$. In this case, the expression reduces to

$$f(x) = (x + y_1) \cdot \prod_{i=2}^{m-1} (c - x + y_i).$$

This function is maximized when

$$1 = x \cdot \sum_{j=2}^{m-1} \frac{1}{c - x + y_j} \leq (m - 1) \cdot \frac{x}{c - x}.$$

From this equation, we have that $x \geq \frac{c}{m} \geq \frac{1}{m^2}$. Hence, the original objective function is maximized when all $p_i \geq \frac{1}{m^2}$.

We therefore change the algorithm HAPLOFREQ in the following way. After every iteration, if the minimum q_i is smaller than $\frac{1}{m^2}$, we maximize the objective as a function of the smallest p_i , when all other values are kept constant, except for the largest p_i . By the above, we see that the optimum of this function will be obtained when the smallest p_i is larger than $\frac{1}{m^2}$. Therefore, in the end of each iteration, one can assume that $\tau \geq \frac{1}{m^2}$. ■

In practical instances, the size of τ is a constant, and therefore the algorithm perform $\tilde{O}(n)$ iterations. In the worst case, the algorithm performs $\tilde{O}(nm^4)$ iterations.

4. ESTIMATING HAPLOTYPE FREQUENCIES FROM UNPHASED GENOTYPES

We now turn to the case where we have a set of genotypes and our goal is to find the frequencies of the underlying haplotypes. Recall that under the Hardy–Weinberg equilibrium, the likelihood function of a set of genotypes \mathcal{G} and a distribution \mathcal{P} is given by Equation (1). Thus, finding the haplotype distribution with the maximum likelihood can be done by solving the following mathematical programming problem:

$$\begin{aligned} & \text{Maximize} && \prod_{g \in \mathcal{G}} \sum_{(h_1, h_2) \in \mathcal{C}(g)} p(h_1)p(h_2) \\ & \text{s.t.} && \sum_{h \in \{0,1\}^k} p(h) = 1 \\ & && p(h) \geq 0, \quad h \in \{0,1\}^k \end{aligned}$$

We follow the approach used for phased data and try to solve this mathematical program by first abstracting it out. We first need the following definition, which is analogous to Definition 1.

Definition 4. Given a genotype $g \in \{0, 1, 2, *\}^k$ and a set of haplotypes $S = \{h_1, \dots, h_m\} \subseteq \{0, 1\}^k$, define the (symmetric) compatibility matrix of g with respect to S as a matrix $A^g \in \{0, 1\}^{m \times m}$ such that $A_{ij}^g = 1$ if $(h_i, h_j) \in \mathcal{C}(g)$ and $A_{ij}^g = 0$ otherwise.

It is easy to verify that the maximum likelihood formulation given above can be solved if the following problem can be solved:

Definition 5. FREQUENCY ESTIMATION OF UNPHASED GENOTYPES.

Input: A set of matrices $\{A_1, \dots, A_n\} \in \{0, 1\}^{m \times m}$.

Goal: Let $\mathcal{P} \subseteq [0, 1]^m$ be the polytope of all probability distribution vectors over m elements $\vec{p} \in \mathfrak{R}^m$ (that is, the set of all vectors \vec{p} such that $\forall i \ p_i \geq 0$ and $\sum_i p_i = 1$). Find the vector in \mathcal{P} that maximizes the product $\prod_i p_i^T A_i \vec{p}$. Formally:

$$\max_{\vec{p} \in \mathcal{P}} f(\vec{p}) = \max_{\vec{p} \in \mathcal{P}} \prod_{i=1}^n p_i^T A_i \vec{p}.$$

Unfortunately, the above mathematical program is NP-hard (see Section 5). We therefore suggest to use a different likelihood function \mathcal{L}_2 which can be thought of as a relaxation of \mathcal{L} . Instead of having a distribution \mathcal{P} over the haplotypes, we assign to each haplotype h_i a k -dimensional vector \vec{v}_i , such that $\sum_{i=1}^k \vec{v}_i = v_0$ where $\|v_0\| = 1$. The likelihood \mathcal{L}_2 is now defined as a function of \mathcal{G} and of $\mathcal{V} = \{\vec{v}_1, \dots, \vec{v}_m\}$:

$$\mathcal{L}_2(\mathcal{G}, \mathcal{V}) = \prod_{g \in \mathcal{G}} \sum_{(h_i, h_j) \in \mathcal{C}(g)} \vec{v}_i \cdot \vec{v}_j.$$

We call \mathcal{V} a vector distribution of the haplotypes. Note that if we restrict the vectors of \mathcal{V} to be in one dimensional space, then \mathcal{V} is a probability distribution and $\mathcal{L}_2(\mathcal{G}, \mathcal{V}) = \mathcal{L}(\mathcal{G}, \mathcal{V})$. The vectors \mathcal{V} can be represented by a matrix \mathcal{P} such that $P_{ij} = \vec{v}_i \cdot \vec{v}_j$; i.e., P_{ij} is the scalar product of \vec{v}_i and \vec{v}_j . Such a matrix \mathcal{P} is called a positive semidefinite (PSD) matrix (see Lovász [1995]). Therefore, an analogous problem to

FREQUENCY ESTIMATION OF UNPHASED GENOTYPES is the following:

Definition 6. MAXIMUM RELAXED UNPHASED LIKELIHOOD.

Input: A set of matrices $\{A_1, \dots, A_n\} \in \{0, 1\}^{m \times m}$.

Goal: Let \mathcal{Q} be the cone of all positive-semi-definite matrices $P \in \mathfrak{R}^{m \times m}$ that satisfy $\sum_{i,j} P_{ij} = 1$, $\forall i, j$. $P_{ij} \geq 0$. Find the PSD matrix in $P \in \mathcal{Q}$ that maximizes the product $\prod_i A_i \bullet P$ (where \bullet stands for the Frobenius inner product⁴). Formally:

$$\max_{\tilde{p} \in \mathcal{Q}} f(P) = \max_{P \in \mathcal{Q}} \prod_{i=1}^n A_i \bullet P.$$

Clearly, if we can solve MAXIMUM RELAXED UNPHASED LIKELIHOOD we could find the vector distribution \vec{V} which maximizes $\mathcal{L}_2(G, V)$. In Section 4.2, we introduce an efficient algorithm which solves MAXIMUM RELAXED UNPHASED LIKELIHOOD in polynomial time.

4.1. Asymptotic behavior of the likelihood function

Finding the maximum likelihood of \mathcal{L}_2 does not ensure us that we will converge to the correct haplotype distribution when the number of samples is sufficiently large. We now turn to show that under Hardy–Weinberg equilibrium, and under the assumption that there is no missing data, if the sample size is large enough, the maximum of $\mathcal{L}_2(\mathcal{G}, \mathcal{V})$ is attained in a point which converges to the correct distribution.

Lemma 5. Under Hardy–Weinberg equilibrium, the solution to MAXIMUM RELAXED UNPHASED LIKELIHOOD converges to the haplotype frequencies in the population.

Proof. Let the set of sampled genotypes be \mathcal{G} . Denote by $p(g)$ the frequency of genotype g in the population. Therefore, $p(g)$ is the probability to sample a genotype $g \in \mathcal{G}$. When the number of samples goes to infinity, the ratio of sampled genotypes g approaches $p(g)$. If the ratio is exactly $p(g)$, then maximizing $\mathcal{L}_2(\mathcal{G}, \mathcal{V})$ is equivalent to maximizing the function

$$\prod_{g \in \mathcal{G}} \left(\sum_{h_i, h_j \in \mathcal{C}(g)} \vec{v}_i \cdot \vec{v}_j \right)^{p_g}.$$

It is easy to see that this objective is maximized when

$$\forall g \in \mathcal{G} \quad \sum_{h_i, h_j \in \mathcal{C}(g)} \vec{v}_i \cdot \vec{v}_j = p_g.$$

As $|\mathcal{G}| \mapsto \infty$, we know that $p_g \mapsto \sum_{i,j \in \mathcal{C}(g)} p_i p_j$. Therefore, one optimal solution to this equation system is the solution $\vec{v}_i = p_i$. Observe that equations above imply that the homozygous genotype g_{ii} with haplotype h_i satisfies that $p_i^2 = \|\vec{v}_i\|^2$. These restrictions, together with the rest of the constraints, determine \mathcal{V} uniquely. We can now use the fact that the function $\max_{\mathcal{V}} \mathcal{L}_2(\mathcal{G}, \mathcal{V})$ is a continuous function G in order to complete the proof. ■

Now that we know that the solution of MAXIMUM RELAXED UNPHASED LIKELIHOOD converges to the correct solution, in particular we know that for large enough sample the vectors \vec{v}_i should be closed to one dimensional. We therefore define $p_i = \vec{v}_i \cdot \vec{1}$ as the suggested probability distribution. By Lemma 5, as the number of samples grow, the probabilities p_i get closer to to the true frequencies in the population.

⁴The Frobenius inner product of matrices X and Y is $\sum_{i,j} X_{ij} Y_{ij}$.

4.2. A polynomial time approximation algorithm for MAXIMUM RELAXED UNPHASED LIKELIHOOD

For all problems defined hereby, our notion of an approximate solution uses the logarithm of the objective function in order to avoid numerical instabilities in practice. A formal definition is as follows:

Definition 7. An ε -approximate solution to one of the probability estimation problems defined above is a probability vector $\vec{p} \in \mathfrak{R}^m$ (or PSD matrix P) such that

$$\log(OPT) - \log f(\vec{p}) \leq \varepsilon.$$

We proceed to provide a polynomial time algorithm for MAXIMUM RELAXED UNPHASED LIKELIHOOD.

An initial solution of value $f(\vec{p}) \geq (\frac{1}{m^2})^n$ can be easily obtained by assigning all probabilities to be $\frac{1}{m}$ (that is, a PSD matrix P where $p_{ij} = \frac{1}{m^2}$).

The same as for the linear case, the running time of our algorithm will depend on the minimal value of $A_i \bullet P$. We set $\tau = \frac{1}{m^4}$. In Lemma 8, we prove that $\min_i p^T A_i p \geq \tau$ for the optimal solution \vec{p} of FREQUENCY ESTIMATION OF UNPHASED GENOTYPES. We therefore add the constraint $\forall i . A_i \bullet P \geq \tau$ to the formulation of MAXIMUM RELAXED UNPHASED LIKELIHOOD, which only tightens the relaxation.

The general framework for our algorithm is identical to the algorithm for the linear case. Starting from the trivial solution above, the algorithm makes a series of local improvements until required performance guaranty is reached. However, for each ‘‘improvement step,’’ we amend the current PSD matrix into another PSD matrix such as to improve the overall value of the solution. We make sure that the additional constraints added to the mathematical program are maintained at all times (i.e., that the matrix is positive semidefinite and that $A_i \bullet P \geq \tau$) The algorithm, called HAPLOFREQ2, is as follows:

Procedure HAPLOFREQ2(ε)

$P \leftarrow J \cdot m^{-2}$

set $q_i \leftarrow A_i \bullet P$

$\Delta \leftarrow \text{FINDPSDDelta}(P, q, \{A_i\})$

while $\sum_i \frac{A_i \bullet \Delta}{q_i} \geq \ln(\varepsilon)$ **do**

Update P to be: $P \leftarrow P + \frac{\tau^2 \varepsilon}{2m} \Delta$

$\Delta \leftarrow \text{FINDPSDDelta}(P, q, \{A_i\})$

return P

The procedure FINDPSDDelta is similar to the procedure FINDDelta used in the linear variant. The description of FINDPSDDelta can be found implicitly in Claim 2.

Theorem 3. For any constant $\varepsilon > 0$, the algorithm HAPLOFREQ2(ε) finds an ε -approximate solution in polynomial time.

The proof is similar in nature to the linear variant proof, with several technical points that need attention. The amendment matrix Δ , which is iteratively added to the current solution, is defined as follows.

Definition 8. Define a (ε, σ) -good matrix with respect to a current solution, P as a matrix Δ , that satisfies the following

1. $W := P + \Delta \succeq 0$
2. $\sum_{i,j} W_{ij} = 1$; $W_{ij} \geq 0$
3. $\forall i \left| \frac{A_i \Delta}{q_i} \right| \leq \sigma$
4. $\sum_{i=1}^n \frac{A_i \Delta}{q_i} \geq \varepsilon$
5. $\forall i . A_i W \geq \tau$

The following claim follows from the fact that the definition of an (ε, σ) -good matrix can be translated to a semi-definite program (that is solvable in polynomial time; see Lovász [1995]) whose objective is to maximize ε . The procedure FINDPSDELTA essentially produces an (ε, σ) -good matrix by solving this semi-definite formulation.

Claim 2. *For any $\sigma \leq 1$, the (ε, σ) -good matrix for the maximal $\varepsilon > 0$ can be found in polynomial time.*

We proceed to show that in case we're far from the optimum, there exists an improvement matrix.

Lemma 6. *If $\frac{OPT}{ALG} = \frac{f(O)}{f(P)} \geq e^\varepsilon$, then there exists an $(\varepsilon\sigma, \frac{2\sigma}{\tau})$ -good matrix for every $0 \leq \sigma \leq 1$.*

Proof. We assume that the current solution P is a PSD matrix and satisfies $\sum_{ij} p_{ij} = 1$. The optimal solution vectors give rise to a natural scaled improvement matrix Δ_σ . Define an intermediate PSD matrix to be a convex combination of P and O as $W := (1 - \sigma)P + \sigma O$. Then

$$\Delta_\sigma := W - P = \sigma(O - P).$$

Notice that Δ_σ is not necessarily PSD and that Δ_σ satisfies the first two conditions of being $(\varepsilon\sigma, \frac{\sigma}{\tau})$ -good, since W is a convex combination of two matrices that satisfy these constraints. In addition,

$$\begin{aligned} \sum_{i=1}^n \frac{A_i \Delta_\sigma}{q_i} &= \sigma \cdot \sum_{i=1}^n \frac{A_i O - A_i P}{q_i} \\ &= \sigma \cdot \sum_{i=1}^n \frac{w_i - q_i}{q_i} \\ &= \sigma \cdot \left(\sum_{i=1}^n \frac{w_i}{q_i} - n \right) \\ &\geq \sigma n \cdot \left(\sqrt[n]{\prod_{i=1}^n \frac{w_i}{q_i}} - 1 \right) \quad \text{by the AMGM inequality} \\ &\geq \sigma n (\sqrt[n]{e^\varepsilon} - 1) \\ &= \sigma n \cdot (e^{\varepsilon/n} - 1) \geq \sigma \cdot \varepsilon \quad \text{by Taylor series of } e^x. \end{aligned}$$

Since $\forall_i |A_i(O - P)| \leq 2$, we have $|\frac{A_i \Delta_i}{q_i}| \leq \frac{2\sigma}{\tau}$.

In addition, since for the optimal solution of MAXIMUM RELAXED UNPHASED LIKELIHOOD we have $\forall_i \cdot A_i O \geq \tau$ (recall previous discussion on τ and see Lemma 8) and invariantly throughout the running of the algorithm we have $A_i P \geq \tau$, it holds that $A_i(P + \Delta_\sigma) = (1 - \sigma)A_i P + \sigma A_i O \geq \tau$. ■

Lemma 7. *Let Δ be aN $(\varepsilon\sigma, \frac{2\sigma}{\tau})$ -good PSD matrix with respect to \vec{p} . Define $P' := P + \Delta_\sigma$ (for $\sigma = \frac{\tau^2 \varepsilon}{8n}$). Then the solution obtained by P' is larger than the one obtained by P by at least*

$$\frac{f(P')}{f(P)} \geq e^{\frac{\tau^2 \varepsilon^2}{16n}}.$$

Proof. Denote

$$c_i := \frac{A_i \cdot \Delta}{q_i}.$$

The new solution $P' = P + \Delta$ satisfies the properties needed of a valid solution, according to the definition of a good matrix. In addition, according to the definition of an $(\varepsilon\sigma, \frac{2\sigma}{\tau})$ -good matrix, we have $|c_i| \leq \frac{2\sigma}{\tau}$. Therefore,

$$\begin{aligned}
\log\left(\frac{f(\vec{p}')}{f(\vec{p})}\right) &= \log\left(\prod_{i=1}^n \frac{A_i(P + \Delta)}{q_i}\right) \\
&= \sum_{i=1}^n \log \frac{q_i + A_i \Delta}{q_i} \\
&= \sum_{i=1}^n \log(1 + c_i) \\
&\geq \sum_{i=1}^n [c_i - (c_i)^2] \quad \text{holds when } |c_i| < \frac{1}{2} \\
&= \sum_{i=1}^n c_i - \sum_{i=1}^n (c_i)^2 \\
&\geq \sigma\varepsilon - n \frac{\sigma^2}{\tau^2} \geq \frac{4\tau^2\varepsilon^2}{n} \quad \text{pick } \sigma = \frac{\tau^2\varepsilon}{16n}. \quad \blacksquare
\end{aligned}$$

The preceding lemmas conclude the proof of Theorem 3 in the same manner as the proof of Theorem 2.

The following lemma proves that for the optimal solution of FREQUENCY ESTIMATION OF UNPHASED GENOTYPES, denoted \vec{p} , we have $\forall i . p^T A_i p \geq \frac{1}{m^4}$. By the previous discussion, this implies that the precision parameter for MAXIMUM RELAXED UNPHASED LIKELIHOOD is bounded by $\tau \leq \frac{1}{m^4}$.

Lemma 8. *Let \vec{p} be the optimal solution for an instance of FREQUENCY ESTIMATION OF UNPHASED GENOTYPES. Then $\forall i . p^T A_i p \geq \frac{1}{m^4}$*

Proof. Suppose that there exists some coordinate with $q_i < \frac{1}{m^4}$. This implies the existence of a coordinate $0 < p_i < \frac{1}{m^2}$. In addition, since \vec{p} is a distribution, there is always one coordinate $p_j > \frac{1}{m}$. From this point on, one can prove that the solution can be further improved by increasing p_i at the expense of p_j , much in the same way as Lemma 4, thereby contradicting optimality. \blacksquare

5. LOWER BOUNDS

In this section, we show the following hardness result for the specific case of FREQUENCY ESTIMATION OF UNPHASED GENOTYPES. In general, strong hardness results for optimizing over polynomials are known (see Bellare and Rogaway [1992]). In some cases, as presented in this paper, the specific structure of a specific polynomial may be used to get a polynomial time algorithm. We show that for the special case of the polynomial introduced in FREQUENCY ESTIMATION OF UNPHASED GENOTYPES, there is no such polynomial time algorithm if $P \neq NP$.

We denote the size of a certain instance for the problem by N (that is, $N = n * m^2$ where n is the number of matrices and m their dimension). We now prove the following.

Theorem 4. *The FREQUENCY ESTIMATION OF UNPHASED GENOTYPES problem is NP-hard to approximate to within $2^{N^{1-\varepsilon}}$ for every constant $\varepsilon > 0$.*

To prove the theorem, we first show that FREQUENCY ESTIMATION OF UNPHASED GENOTYPES is NP-hard, via a reduction from the MAXIMUM CLIQUE problem. The reduction is done using the following simple rule. An instance for the MAXIMUM CLIQUE problem is a graph $G = (V, E)$. We use the adjacency matrix A_G as an instance for FREQUENCY ESTIMATION OF UNPHASED GENOTYPES. In particular, in this new instance, the set of matrices $\{A_1, \dots, A_n\}$ consists of a single matrix, which is the adjacency matrix of G . We denote this new instance as I_G . Rewriting the objective function, we get that a probability p_i is assigned to every vertex in the graph, and we seek to maximize $\sum_{(i,j) \in E} p_i p_j$ where $\sum_{i \in V} p_i = 1$ and $p_i \geq 0$.

Claim 3. *For a given graph G , the objective of the FREQUENCY ESTIMATION OF UNPHASED GENOTYPES instance I_G has value $\frac{1}{2}(1 - \frac{1}{r})$ if and only if the maximal clique size of G is r .*

Proof. If the graph G contains a clique of size r , then by assigning $p_i = \frac{1}{r}$ for all vertices of this clique and 0 for all other vertices, we obtain an objective value of $\frac{1}{2}(1 - \frac{1}{r})$ for I_G .

The other direction is a generalization of Turán’s theorem (see Alon and Spencer [1992]). Turán’s theorem states that for a graph with n vertices that does not contain a clique of size $r + 1$ the maximum number of edges is $\frac{n^2}{2}(1 - \frac{1}{r})$. Furthermore, the maximum is attained for the complement of the graph composed of r disjoint cliques of size $\frac{n}{r}$ each (these graphs are called Turán’s graphs). When every vertex has a weight p_i and the weight of an edge is $p_i p_j$, then Turán’s theorem can be viewed as a special case of our claim, in which all p_i must be set to $\frac{1}{n}$, and thus all edges have the same weight. We now show that even when the vertices may have different weights, as long as the weights are nonnegative, the objective function value for a graph without K^{r+1} is maximized for the Turán graph $T^r(n)$, in which it is precisely $\frac{1}{2}(1 - \frac{1}{r})$.

We first consider the properties of an optimal solution to the instance I_G . Consider an optimal solution p_1, \dots, p_n , and consider any two vertices v_i, v_j , such that $1 > p_i, p_j > 0$. One can verify that by setting $p_i \mapsto p_i + \varepsilon$ and $p_j \mapsto p_j - \varepsilon$, the objective function will be changed in the following way:

$$\Delta(i, j, \varepsilon) = \pm \Theta(\varepsilon^2) + \varepsilon \sum_{t \in \Gamma_i \Delta \Gamma_j} x_t,$$

where Γ_i denotes set of neighbors of v_i , and $\Gamma_i \Delta \Gamma_j$ is the symmetric difference between the two sets. We thus get that for all p_i, p_j that are nonzero, the optimal solution satisfies $P_i = P_j$, where $P_i \triangleq \sum_{j \in \Gamma(i)} p_j$ is the sum of all variables corresponding to the vertices in $\Gamma(i)$. We denote $\forall i . P = P_i$.

Assume that for the optimal solution, $\sum_{(i,j) \in E} p_i p_j > \frac{1}{2}(1 - \frac{1}{r})$. It suffices to show that the maximal clique size in G is $\omega(G) > r$. By the definition of P , we get

$$P = \sum_{i \in V} p_i P = \sum_{i \in V} p_i \sum_{j \in \Gamma_i} p_j = 2 \sum_{(i,j) \in E} p_i p_j > 1 - \frac{1}{r}.$$

Consider the subgraph of all vertices with $p_i > 0$, and consider the largest clique of this subgraph, say, of size k . For simplicity of notations, assume that the variables of such a clique are p_1, \dots, p_k . The set of neighbors of these vertices satisfy

$$\sum_{i=1}^k P_i = \sum_{i=1}^k \sum_{j \in \Gamma(i)} p_j \leq (k - 1) \sum_{j \in V} p_j \leq k - 1,$$

where the first inequality holds since each vertex in the graph can be a neighbor of at most $k - 1$ vertices of the clique. On the other hand, we have that $\sum_{i=1}^k P_i = k \cdot P > k(1 - \frac{1}{r})$. Using both inequalities, we get that $k(1 - \frac{1}{r}) < k - 1$ and thus $k > r$. ■

In order to prove Theorem 4, we use the following theorem by Hastad (see Hastad [1996]),

Theorem 5 (Hastad). *It is NP-hard to decide whether a given graph on n vertices contains a clique of size $n^{99/100}$ or whether the maximal clique in the graph is of size at most $n^{1/100}$.*

Using this theorem we can now prove the following.

Proof (Theorem 4). According to Claim 3 and Hastad’s theorem, FREQUENCY ESTIMATION OF UNPHASED GENOTYPES is NP-hard to approximate to within

$$\frac{(1 - N^{-99/100})}{(1 - N^{-1/100})} = 1 + \frac{N^{-1/100} - N^{-99/100}}{1 - N^{-1/100}} \geq 1 + \frac{1}{\sqrt{N}}.$$

Now, amplify this construction as follows: Given a graph G , create an instance of FREQUENCY ESTIMATION OF UNPHASED GENOTYPES as follows. In this instance, the set of matrices $\{A_1, \dots, A_n\}$ consists of M copies of the adjacency matrix of G . We denote this new instance as I'_G .

The size of the instance built is MN , where N is the size of the instance I_G described in Claim 3 (with a single copy of the adjacency matrix of G). Since I_G is hard to approximate within $1 + \frac{1}{\sqrt{N}}$, it follows that I'_G is hard to approximate within

$$\left(1 + \frac{1}{\sqrt{N}}\right)^M \geq e^{-M/\sqrt{N}}.$$

Taking M to be a large polynomial in N , say $M = N^k$, we get an instance of size $T \triangleq N^{k+1}$ that is hard to approximate within $e^{-N^{k-1/2}} \leq e^{-T^{1-\frac{2}{k}}}$. Taking $k = \frac{1}{2\epsilon}$ completes the proof. ■

6. EXPERIMENTAL RESULTS

We implemented the algorithms HAPLOFREQ and HAPLOFREQ2 (described in Sections 3.1 and 4.2, respectively) and compared them to the widely used software PHASE (Stephans *et al.*, 2001).

Implementation details. Both HAPLOFREQ and HAPLOFREQ2 assume that the number of possible haplotypes is limited—and usually small. In many scenarios, this is actually the case, but in order to avoid extensive running time in the other cases, we use a preprocessing mechanism which filters out haplotypes that seem to have an extremely small frequency. The preprocessing mechanism is based on a greedy procedure, similar to the one given by Halperin and Karp (2003). After the preprocessing we are typically left with about 50 possible haplotypes on which we run our algorithms.

In each iteration of HAPLOFREQ2, we have to solve a semidefinite program. Even though semidefinite programs can be solved in polynomial time, existing software packages are relatively slow. We therefore implemented a semidefinite programming solver which is specifically tailored for our needs. This implementation takes advantage of some properties of our algorithm in order to speed up the semidefinite solver.

The resulting implementation of both algorithms is very efficient. In particular, HAPLOFREQ (on haplotypes) typically runs 15 to 25 times faster than PHASE, and HAPLOFREQ2 (on genotypes) typically runs 3 to 10 times faster. Figure 2 gives a concise comparison of measured running times of PHASE, HAPLOFREQ, and HAPLOFREQ2.

The datasets. We applied our algorithms to the dataset of Daly *et al.* (2001) and Rioux *et al.* (2001) and to population D of Gabriel *et al.* (2002). The first dataset is a 500 kilobase region of chromosome 5q31, spanning 103 SNPs collected from 129 mother, father, child trios from a European-derived population in an attempt to identify a genetic risk factor for Crohn’s disease. A significant portion of the genotype data

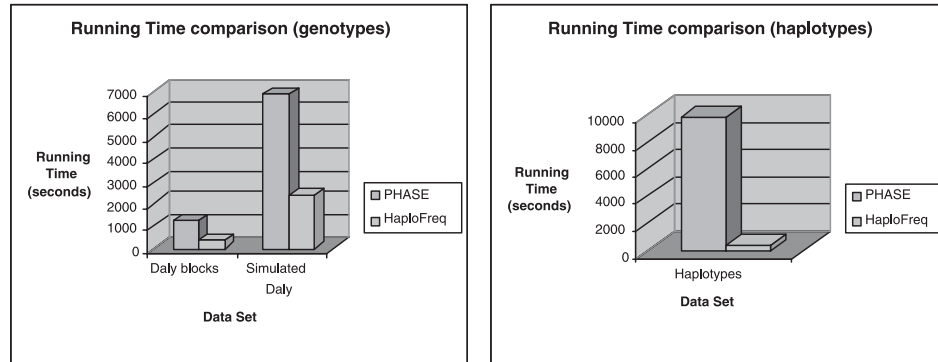


FIG. 2. Running times comparison.

(about 10%) is missing with an average of 10 SNPs per individual's genotype missing. This data set was partitioned by Daly *et al.* (2001) and Rioux *et al.* (2001) into 11 blocks of high correlation. Since this set consists of trios, we can infer each individual's haplotypes in all positions except for the positions where all three individuals are heterozygous or missing. We use populations D from the data of Gabriel *et al.* (2002) which has pedigree information. The data consists of genotypes of SNPs from 62 regions. Population D consists of 90 individuals from 30 trios from Yoruba.

Simulating distributions. In order to evaluate the performance of HAPLOFREQ, we need to know what the underlying distribution in the population is. We therefore partitioned the data into regions containing 5, 12, and 19 SNPs. For each of those regions, we used the trios to infer the haplotypes and used the resulting haplotype distribution to generate more datasets by picking haplotypes randomly and independently from that distribution. We then added randomly scattered missing data and randomly scattered sequencing errors. Note that these simulations implicitly assume that the underlying genotype distribution in the population has no departures from the Hardy–Weinberg equilibrium. On the other hand, when we sample from that distribution, the sampling deviations result in departures from Hardy–Weinberg equilibrium.

Distance measures. We use two measures for the distance between two distributions. The first measure is the l_1 norm of the difference between the two distributions. Given two distributions, $\{p_1, \dots, p_k\}$ and $\{q_1, \dots, q_k\}$, the l_1 norm of their difference is defined as $\sum_{i=1}^k |p_i - q_i|$. We also used the chi-square difference, that is, $\sum_{i=1}^k \frac{(p_i - q_i)^2}{q_i}$. The chi-squared distance is particularly interesting since when an association study is performed, one uses the chi-squared test in order to test the hypothesis that the two underlying distributions are the same. In both cases, we take the sum only over the probabilities q_i that are greater than 0.05.

Accuracy of estimations. We compared the accuracy of the frequency estimations of our HAPLOFREQ to PHASE (Stephans *et al.*, 2001). We considered all possible regions spanning 5, 12, and 19 SNPs. For each of those regions, we used the trios' information to deduce the haplotypes whenever possible and used the distribution of the deduced parents' haplotypes as the underlying distribution. We then ran both PHASE and HAPLOFREQ over the data containing the parents' deduced haplotypes (with missing data whenever there was an ambiguity). We find that HAPLOFREQ is typically 10–50% more accurate than PHASE on both datasets.

Additionally, we compared our algorithms over the simulated datasets described above. In this case, the underlying distribution is known, and therefore we can compare the methods both to the sampled distribution and to the distribution of haplotypes in the population. We compared HAPLOFREQ to PHASE over these datasets, and we found again that HAPLOFREQ is typically 10 – 50% more accurate than PHASE.

We note that the deviations caused by sampling have an enormous affect on all three algorithms. In particular, the distributions found by all three algorithms (PHASE, HAPLOFREQ, and HAPLOFREQ2) are much closer to the sampled distribution than to the underlying population distribution. A complete summary of the comparison can be found in Figs. 3, 4, 5, and Tables 1 and 2.

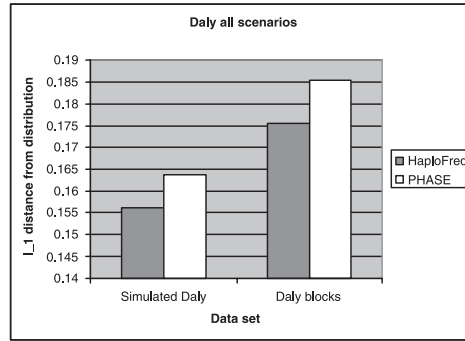


FIG. 3. Average l_1 distance from the actual distribution on the Daly datasets (both blocks and simulated).

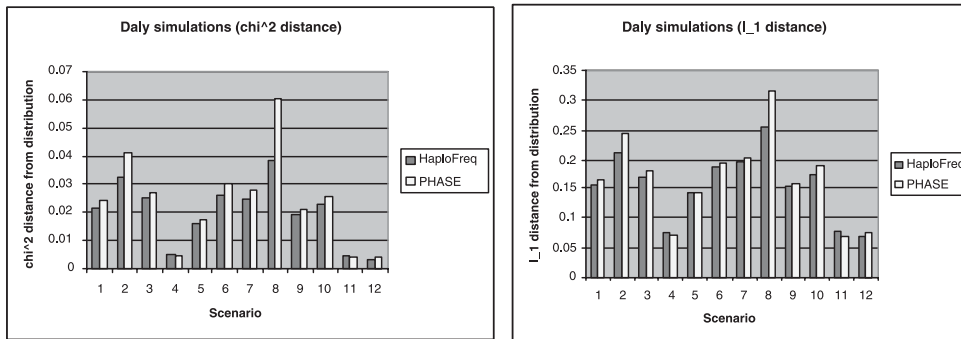


FIG. 4. Average χ^2 and l_1 distances from the actual distribution on the simulated Daly data, with various simulation parameters, as detailed in Table 1.

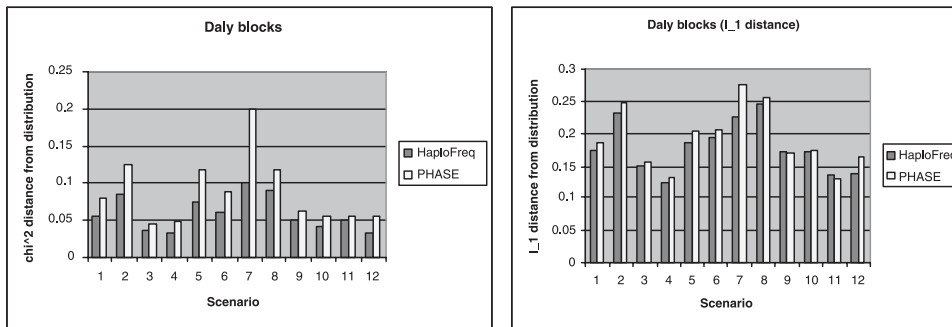


FIG. 5. Average χ^2 and l_1 distances from the actual distribution on the Daly blocks data, with various simulation parameters, as detailed in Table 2.

TABLE 1. SIMULATION PARAMETERS FOR THE SIMULATED DALY DATASET

Scenario	Simulation parameters
1	All parameters
2, 3, 4	Sets of 25, 50, 75 genotypes, respectively
5, 6	10%, 20% missing data, respectively
7, 8	Sets of 25 genotypes with 10%, 20% missing data, respectively
9, 10	Sets of 50 genotypes with 10%, 20% missing data, respectively
11, 12	Sets of 75 genotypes with 10%, 20% missing data, respectively

TABLE 2. SIMULATION PARAMETERS FOR THE DALY BLOCKS DATASET

Scenario	Simulation parameters
1	All parameters
2, 3, 4	Sets of 20, 40, 60 genotypes, respectively
5, 6	10%, 20% missing data, respectively
7, 8	Sets of 20 genotypes with 10%, 20% missing data, respectively
9, 10	Sets of 40 genotypes with 10%, 20% missing data, respectively
11, 12	Sets of 60 genotypes with 10%, 20% missing data, respectively

ACKNOWLEDGMENTS

E.H. supported by Sanjeev Arora's David and Lucile Packard Fellowship and NSF grant CCR-0205594.

REFERENCES

- Alon, N., and Spencer, J.H. 1992. *The Probabilistic Method*, Wiley, New York.
- Bellare, M., and Rogaway, P. 1992. The complexity of approximating a quadratic program. *Journal of Mathematical Programming B*, 69, 429–441.
- Clark, A.G. 1990. Inference of haplotypes from pcr-amplified samples of diploid populations. *J. Mol. Biol. Evol.* 7(2), 111–122.
- Daly, M.J., Rioux, J.D., Schaffner, S.F., Hudson, T.J., and Lander, E.S. 2001. High-resolution haplotype structure in the human genome. *Nature Genet.* 29(2), 229–232.
- Excoffier, L., and Slatkin, M. 1995. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *J. Mol. Biol. Evol.* 12(5), 921–927.
- Fallin, D., and Schork, N.J. 2000. Accuracy of haplotype frequency estimation for biallelic loci, via the expectation-maximization algorithm for unphased diploid genotype data. *Am. J. Human Genet.* 67, 947–959.
- Gabriel, G.B., Schaffner, S.F., Nguyen, H., Moore, J.M., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M., Liu-Cordero, S.N., Rotimi, C., Adeyemo, A., Cooper, R., Ward, R., Lander, E.S., Daly, M.J., and Altshuler, D. 2002. The structure of haplotype blocks in the human genome. *Science* 296, 2225–2229.
- Grötschel, M., Lovász, L., and Schrijver, A. 1988. *Geometric Algorithms and Combinatorial Optimization*, Springer Verlag, New York.
- Gusfield, D. 2000. A practical algorithm for optimal inference of haplotypes from diploid populations. *Proc. 8th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*.
- Gusfield, D. 2001. Inference of haplotypes from samples of diploid populations: Complexity and algorithms. *J. Comp. Biol.* 8(3), 305–323.
- Gusfield, D. 2002. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. *Proc. 6th Ann. Int. Conf. on Research in Computational Molecular Biology*.
- Halperin, E., and Eskin, E. 2004. Haplotype reconstruction from genotype data using imperfect phylogeny. *Bioinformatics*.
- Halperin, E., and Karp, R. 2003. The minimum-entropy set cover problem. Manuscript.
- Hastad, J. 1996. Clique is hard to approximate within $n^{1-\epsilon}$. *Proc. 37th Ann. Symp. on Foundations of Computer Science*, 627.
- Hawley, M.E., and Kidd, K.K. 1995. Haplo: A program using the em algorithm to estimate the frequencies of multi-site haplotypes. *J. Heredity* 86(5), 409–411.
- Khachiyan, L.G. 1980. Polynomial algorithms in linear programming. *USSR Comp. Math. and Math. Phys.* 20, 53–72.
- Kimmel, G., and Shamir, R. 2004. Maximum likelihood resolution of multi-block genotypes. *Proc. 8th Ann. Int. Conf. Computational Molecular Biology*, 2–9.
- Lancia, G., Bafna, V., Istrail, S., Lippert, R., and Schwartz, R. 2001. Snps problems, algorithms and complexity. *Proc. Eur. Symp. on Algorithms (ESA-2001)*, 182–193.
- Long, J.C., Williams, R.C., and Urbanek, M. 1995. An e-m algorithm and testing strategy for multiple-locus haplotypes. *Am. J. Human Genet.* 56(3), 799–810.
- Lovász, L. 1995–2001. Semidefinite optimization. Lecture notes.
- Michalatos-Beloin, S., Tishkoff, S.A., Bently, K.L., Kidd, K.K., and Ruano, G. 1996. Molecular haplotyping of genetic markers 10 kb apart by allele-specific long-range pcr. *Nucl. Acids Res.* 24, 4841–4843.

- Nesterov, Y., and Nemirovskii, A. 1994. *Interior Point Polynomial Methods in Convex Programming*, SIAM, Philadelphia, PA.
- NIH. 2002. Large-scale genotyping for the haplotype map of the human genome. RFA: HG-02-005.
- Niu, Qin, Xu, and Liu. 2001. In silico haplotype determination of a vast set of single nucleotide polymorphisms. Technical report, Department of Statistics, Harvard University.
- Patil, N., Berno, A.J., Hinds, D.A., *et al.* 2001. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science* 294(5547), 1719–1723.
- Rioux, J.D., Daly, M.J., Silverberg, M.S., *et al.* 2001. Genetic variation in the 5q31 cytokine gene cluster confers susceptibility to crohn disease. *Nature Genet.* 29(2), 223–228.
- Stephens, M., Smith, N., and Donnelly, P. 2001. A new statistical method for haplo-type reconstruction from population data. *Am. J. Human Genet.* 68, 978–989.

Address correspondence to:

Eran Halperin
International Computer Science Institute
1947 Center St., Suite 600
Berkeley, CA

E-mail: heran@icsi.berkeley.edu