HAPLOFREQ - Estimating Haplotype Frequencies Efficiently

Eran Halperin[*] Elad Hazan[†]

Abstract

We present a new method HAPLOFREQ to estimate haplotype frequencies over a short genomic region given the genotypes or haplotypes with missing data. 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 data.

In contrast to previous approaches, our algorithms are guaranteed to converge in polynomial time to a global maximum of the different likelihood functions. Preliminary experiments on biological data show that our estimates are about 10% more accurate than the popular program PHASE and about three to ten times faster.

Our techniques involve new algorithms in convex optimization. These algorithms may be of independent interest. Furthermore, the hardness proof involves a generalization of Turan's theorem, which may also be of independent interest.

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. 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 ([19]).

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 vs. 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 [21, 18] but these are often costly and are not suitable for large scale polymorphism screening obtains *genotype* information at each SNP.

As an alternative to those technologies, many computational methods have been developed for phasing the genotypes (e.g. [3, 10, 11, 16, 23, 20, 9, 12]). In many of the applications, it is crucial to estimate correctly the haplotype frequencies in the population and not necessarily to phase the individual genotypes. There are however a few EM-based (Expectation Maximization) algorithms that aim to estimate the haplotype frequencies ([6, 7, 14, 17]). 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

[*] Computer Science Department, Princeton University. `www.cs.princeton.edu/∼heran`.

[†] Computer Science Department, Princeton University. `www.cs.princeton.edu/∼ehazan`.

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.

These methods try to cope both with the fact that the genotype phase is unknown and with additional noise. In particular, every existing sequencing technique to date introduces some errors and/or missing data. Some of these methods provide the phased haplotypes but with the cost of having a substantial amount of missing data (see e.g. [21]). Therefore, coping with missing data for either phased or unphased genotypes is of great interest.

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 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 in overcoming 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.

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 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 to the haplotypes frequencies in the population.

2 Estimating Haplotype Frequencies

We first consider the case where we have a set of partial haplotypes sampled from a set of unrelated individuals. We assume that the haplotypes have a given frequency distribution in the population, and that we are given a set of haplotypes which are independently sampled from this distribution. We further assume that the set of sampled haplotypes also contain missing data.

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

We say that a partial haplotype $h_1 \in \{0,1,*\}^m$ is **consistent** with a complete haplotype $h_2 \in \{0,1\}^m$ 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.

Let $\mathcal{P}$ be a distribution over the set of all possible complete haplotypes of length m. We denote by $p(h)$ the probability assigned to the haplotype h by $\mathcal{P}$. Given the set of partial haplotypes $\mathcal{H}$, the likelihood of $\mathcal{P}$ is given by

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

Thus, finding the distribution of maximum likelihood can be done by solving the following mathematical programming problem:

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

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

Definition 1 *Given a partial haplotype $h \in \{0,1,*\}^m$ and a set of haplotypes $S = \{h_1, ..., h_n\} \subseteq \{0,1\}^m$, define the* **compatibility vector** *of h with respect to S as a vector $A_h \in \{0,1\}^n$ such that $A_h(i) = 1$ if $h_i \in \mathcal{C}(h)$ and $A_h(i) = 0$ otherwise. For our purposes, the set of haplotypes S is usually all possible haplotypes of length m. In this case we shall drop the extra notation.*

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

Definition 2 (Frequency Estimation for Phased Genotypes) *.*
Input: *A matrix $A \in \{0,1\}^{n \times m}$ consisting of n row vectors $\{A_1, ..., A_n\} \in \{0,1\}^m$*
Goal: *Find a vector $\vec{p} \in \Re^n_+$, such that:*

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

2.1 Algorithms for Frequency Estimation for Phased Genotypes

We first prove that the problem is solvable in polynomial time, and describe a polynomial time algorithm. This is quite surprising given that the problem Frequency Estimation for Phased Genotypes is essentially finding a maximum point of a polynomial of potentially high degree. In general, finding an extremum of a polynomial is an intractable problem, see section 4 for further detail. However, in this special case we prove the following:

Theorem 1 Frequency Estimation for 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 [15] to provide a solution.

We now provide a separation oracle. Given some vector $\vec{p} \in \Re^m_+$, let $\vec{q} = A \cdot \vec{p} \in \Re^m_+$, define the following function:

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

And the corresponding hyper-plane:

$$H_p \overset{\text{def}}{=} \{\vec{x} \in \Re^n_+ | g_p(\vec{x}) \geq n \cdot b\}$$

Claim 1 *Given a point $\vec{x} \in \Re^m_+$ 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:

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

Now consider any point $\vec{z} \in \Re^m_+$ 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:

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

Denote $c_i = \frac{(A\vec{z})_i}{q_i}$. Then we have $g_p(\vec{z}) \geq b \cdot \left[\prod_{i=2}^n \frac{1}{c_i} + \sum_{i=2}^n c_i\right]$. From symmetry, this function is minimized when $\forall i \ \ c_i = c$ for some $c > 0$. So we get: $g_p(\vec{z}) \geq b \cdot \left[\frac{1}{c^{n-1}} + (n-1)c\right]$. 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 then $\vec{p}$ 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 for Phased Genotypes without assumption of Hardy-Weinberg equilibrium.

As the ellipsoid algorithm is slow in practice for many applications, we proceed to provide an efficient combinatorial algorithm that approximates the solution to Frequency Estimation for Phased Genotypes to within any required (constant) precision parameter.

2.2 Combinatorial Approximation Algorithm for Frequency Estimation for Phased Genotypes

Let the input be $A \in \{0,1\}^{n \times m}$. Denote the solution vector (optimal probabilities assigned to the haplotypes) by $\{o_1, ..., o_n\}$. Denote $\vec{q} = A \cdot \vec{p}$. Also define $\vec{w} = A \cdot \vec{o}$ as the "weights" vector of the optimum solution. Let f be the objective function, that is $f(\vec{x}) = \prod_i (A_i \vec{x})$.

A trivial observation, is that we can always obtain the value of: $f(\vec{p}) \geq \left(\frac{1}{n}\right)^n$ by picking from each row one p_i, and then assigning equal weights to all of those picked. Alternatively, we can obtain an initial value of $f(\vec{p}) \geq \left(\frac{1}{m}\right)^n$ by assigning all probabilities to be $\frac{1}{m}$.

Let τ be a precision parameter to our algorithm. That is, all probabilities will be rounded to the nearest value within τ distance. In particular, we assume that every non-zero p_i or q_i value is at least τ. We further discuss precision in subsection 2.4.

Starting from the trivial solution above, our algorithm makes a series of improvements up to the required performance guaranty is reached. In each "improvement step" we amend the current vector of probabilities $\vec{p}$ to $\vec{p'} = \vec{p} + \vec{\delta}$, such that to improve the overall value. The algorithm, called HaploFreq (when not assuming Hardy-Weinberg equilibrium, further on we describe a version which does assume Hardy-Weinberg equilibrium), which takes as an input a precision parameter ε, works as follows:

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

We proceed to prove correctness. A vector δ as used by the algorithm must satisfy various conditions to be a valid amendment. We define the properties of such an "amendment vector":

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

1. $\sum_{i=1}^{m} \delta_i = 0$
2. $0 \leq \delta_i + p_i \leq 1$
3. $\sum_{i=1}^{n} \frac{A_i\vec{\delta}}{q_i} \geq \varepsilon$

In subsequent discussion, we describe the procedure FindDelta used by HaploFreq to find a ε-good vector if one exists. 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 e^{ε}-approximate solution in polynomial time.*

To prove this theorem, we first prove that we can always find a ε-good vector if our current solution is not a e^{ε}-approximate solution. We then show that using a ε-good vector we can improve our current solution, and that polynomially many improvements suffice to obtain a ε-approximate solution. Finally, we show how to efficiently implement the procedure FindDelta.

Lemma 1 *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 three conditions above, and as for the last:

$$\begin{aligned}
\sum_{i=1}^{n} \frac{A_i\vec{\delta}}{q_i} \quad & = \textstyle\sum_{i=1}^{n} \frac{w_i - q_i}{q_i} \\
& = \textstyle\sum_{i=1}^{n} \frac{w_i}{q_i} - n \\
& \geq n \cdot \sqrt[n]{\textstyle\prod_{i=1}^{n} \frac{w_i}{q_i}} - n \qquad && \text{by the AMGM inequality} \\
& \geq n \cdot \sqrt[n]{e^{\varepsilon}} - n \\
& = n \cdot \sqrt[n]{e^{\varepsilon}} - n \ = \ n \cdot (e^{\varepsilon/n} - 1) \\
& \geq n \cdot (1 + (\varepsilon/n) - 1) \ \geq \ \varepsilon && \text{by Taylor series of } e^x
\end{aligned}$$

□

And indeed, suppose that we can find such a vector δ, we could get closer to the optimum, as shown by the following lemma:

Lemma 2 *Let $\vec{\delta}$ be a ε-good vector with respect to $\vec{p}$. Let τ be the smallest q_i for this $\vec{p}$. Define $p' := p + \sigma\delta$ (for $\sigma = \frac{\tau^2\varepsilon}{2n}$). Then the solution obtained by p' is larger then the one obtained by p by at least:*

$$\frac{f(\vec{p'})}{f(\vec{p})} \geq e^{\frac{\tau^2\varepsilon^2}{4n}}$$

Proof: Denote $c_i := \frac{A_i\vec{\delta}}{q_i}$. We assume that $q_i \geq \tau$. In addition, from the definition of $\vec{\delta}$ it follows that

$|A_i\vec{\delta}| \leq 1$, and therefore: $|c_i| \leq \frac{1}{\tau}$. Hence:

$$\begin{aligned}
\log\left(\frac{f(\vec{p'})}{f(\vec{p})}\right) &= \log\left(\prod_{i=1}^{n} \frac{A_i(\vec{p}+\sigma\vec{\delta})}{q_i}\right) \\
&= \sum_{i=1}^{n} \log \frac{A_i(\vec{p}+\sigma\vec{\delta})}{q_i} \\
&= \sum_{i=1}^{n} \log(1+\sigma c_i) \\
&= \sum_{i=1}^{n} \left[\sum_{j=1}^{\infty} \frac{1}{j}(\sigma c_i)^j(-1)^{j+1}\right] && \text{by Taylor series expansion} \\
&\geq \sum_{i=1}^{n} \left[(\sigma c_i) - (\sigma c_i)^2\right] && \text{assuming } |\sigma c_i| < \tfrac{1}{2}\text{, see claim 6} \\
&\geq \sigma\varepsilon - \sigma^2 \sum_{i=1}^{n} c_i^2 \\
&\geq \sigma\varepsilon - \frac{n\sigma^2}{\tau^2} \geq \frac{\tau^2\varepsilon^2}{4n} && \text{for } \sigma = \frac{\tau^2\varepsilon}{2n}
\end{aligned}$$

□

This shows that an improvement has been made towards the optimal solution. The following lemma will be proved in subsection 2.3.

Lemma 3 *The procedure* FindDelta*, that finds a ε-good vector if one exists, can be implemented to run in time $O(nm + m\log m)$.*

We now have all the ingredients needed to prove theorem 2:

Proof:[Theorem 2]

We can obtain an initial solution with value at least m^{-n} (see above). According to lemmas 1,2, as long as we are e^{ε}-far from the optimum, we can find a ε-good vector.

Suppose we make r iterations, then the final value will be at least:

$$m^{-n} \cdot e^{\frac{\tau^2\varepsilon^2}{4n}r} \geq e^{-n\log m} e^{\frac{\tau^2\varepsilon^2}{4n}r}$$

As the optimum is bounded by 1, there can be at most $r = \Omega(\frac{n^2 \log m}{\varepsilon^2\tau^2})$ iterations. □

2.3 Implementing FindDelta

An ε-good vector can be found in polynomial time by solving the LP derived from definition 3. In this subsection we describe how to find an ε-good vector combinatorially in $O(nm + m\log m)$ time.

```
Procedure FindDelta(p,q,A)
Let $\vec{\alpha}$ such that $\forall i \,.\, \vec{\alpha}_i = (\vec{1} \cdot A)_i / q_i$
Suppose w.l.o.g that $\alpha_1 \leq \alpha_2 \leq ... \leq \alpha_m$ (o/w sort $\vec{\alpha}$)
Set $\delta_m = 1 - p_m$
Set $\forall i < m \,.\, \delta_i = -p_i$
return $\vec{\delta}$
```

Claim 2 *The procedure* FindDelta *above find an ε-good vector if one exists.*

Proof: The vector returned by FindDelta obviously satisfies the second of the conditions of a 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 $\vec{\delta}$. □

2.4 A note on precision

Notice that our analysis of the running time of HaploFreq so far had a polynomial dependence on the precision parameter τ. In this subsection we prove that the parameter is indeed polynomial in the size of the problem.

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

Proof: This is obviously true for the first solution chosen. Suppose that after some local improvement there exists some $q_1 < \frac{1}{m^2}$. This implies the existence of a $p_i < \frac{1}{m^2}$, denote it p_1. In addition, there is always one $p_j > \frac{1}{m}$, denote it p_2.

Suppose that p_1 appears only in one q_i, and p_2 appears in all the rest of the q_j's (otherwise our claim only strengthens). Let $p_1 + p_2 = c > \frac{1}{m}$. Let us optimize over the value of p_1 with respect to p_2. The expression obtained is:

$$f(p_1) = p_1 \cdot \prod_{i=2}^{m-1} (c - p_1 + \delta_i)$$

Taking the derivative (and denoting $x = p_1$):

$$f'(x) = \prod_{i=2}^{m-1} (c - x + \delta_i) - x \cdot \sum_{j=2}^{m-1} \prod_{i \neq j} (c - x + \delta_i)$$

Finding where the derivative equals zero, and dividing by $\prod_{i=2}^{m-1}(c - x + \delta_i)$, we get:

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

Which implies:

$$x \geq cm \geq \frac{1}{m^2}$$

Therefore, assuming that we find the optimal ε-good vector at each iteration, the minimum q_i must be larger then this quantity. □

3 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. We will first introduce some notations.

We denote a genotype by a string over $\{0, 1, 2, *\}^m$, 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 $g \in \{0, 1, 2, *\}^m$, and a pair of complete haplotypes $h^1, h^2 \in \{0, 1\}^m$ 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, we assume that the two haplotypes of each individual are independently picked from the distribution of haplotypes in the population. Under Hardy Weinberg equilibrium, the likelihood function of a set of genotypes $\mathcal{G}$ and a distribution $\mathcal{P}$ is given by

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

Thus, finding the haplotype distribution with the maximum likelihood can be done by solving the following mathematical programming problem:

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

This problem can be formalized in a more general way. We first need to introduce another definition.

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

Thus, 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 for Unphased Genotypes) .
Input: *A set of matrices $\{A_1, ..., A_n\} \in \{0,1\}^{m \times m}$*
Goal: *Let $\mathcal{P} \subseteq [0,1]^m$ be the polytope of all probability distribution vectors over m elements $\vec{p} \in \Re^m$ (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 $\prod_i \vec{p^T} A_i \vec{p}$. Formally:*

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

Unfortunately, the above mathematical program is NP-hard (as we explain later in section 4). However, consider the following relaxation of the problem:

Definition 6 (Relaxed Frequency Estimation) .
Input: *A set of matrices $\{A_1, ..., A_n\} \in \{0,1\}^{m \times m}$*
Goal: *Let $\mathcal{Q}$ be the cone of all positive-semi-definite matrices $P \in \Re^{m \times m}$ 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 product). Formally:*

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

3.1 Asymptotic Behavior of the Likelihood Function.

Given this relaxation, it is natural to ask what is the relation between the optimal value of the relaxation and the true frequencies of the haplotypes. We now show that under Hardy-Weinberg equilibrium, and under the assumption that there is no missing data, if the sample size is large enough, the optimal relaxed likelihood is attained for a distribution which is very close to the actual frequencies in the population.

Formally, we prove that the solution of Relaxed Frequency Estimation converges to the underlying frequencies as the number of samples increases, $n \mapsto \infty$ (note that this is obvious for Frequency Estimation for Unphased Genotypes).

Lemma 5 *Denote by n the number of genotypes sampled as input to* Relaxed Frequency Estimation. *Under the Hardy-Weinberg generation model, the solution to* Relaxed Frequency Estimation *converges to the underlying haplotype frequencies.*

Proof: Let the genotype set sampled be $\mathcal{G}$. Under the SDP formulation, the maximization function is (for a PSD matrix $Q \succeq 0; Q_{ij} \geq 0; \sum Q_{ij} = 1$):

$$\prod_{g \in \mathcal{G}} \sum_{i,j \in \mathcal{C}(g)} Q_{ij}$$

Denote by $p(g)$ the probability to sample a genotype $g \in \mathcal{G}$. Then disregarding normalization, the maximization function is:

$$\prod_{g \in \mathcal{G}} \left(\sum_{i,j \in \mathcal{C}(g)} Q_{ij} \right)^{p_g}$$

It is easy to see that this objective is maximized when:

$$\forall_{g \in \mathcal{G}} \sum_{i,j \in \mathcal{C}(g)} Q_{ij} = p_g$$

As $n \mapsto \infty$, we know that $p_g = \sum_{i,j \in \mathcal{C}(g)} p_i p_j$. Therefore, one optimal solution to this equation system is the rank 1 PSD matrix $Q_{ij} = p_i p_j$. Observe that equations above contain, in particular, the equation $p_{g_{ii}} = p_i^2 = Q_{ii}$. These restrictions, together with the rest of the constraints, determine Q uniquely. □

3.2 A Polynomial Time Approximation Algorithm for the SDP Genotype Estimation Problem

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 $\vec{p} \in \Re^m$ (or PSD matrix P) such that:*

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

We proceed to provide a polynomial time algorithm for the Positive-Semi-Definite Probability Estimation Problem.

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

Same as for the linear case, we denote by τ the precision parameter to our algorithm. Precision issues are handled similarly.

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 up to the required performance guaranty is reached. However, for each "improvement step" we amend the current PSD matrix into another PSD matrix such that to improve the overall value of the solution. The algorithm, called HaploFreq2 is as follows:

```
Procedure HaploFreq2($\varepsilon$)
$P \leftarrow J \cdot m^{-2}$
set $q_i \leftarrow A_i \bullet P$
$\Delta \leftarrow$ FindPsdDelta($P, q, \{A_i\}$)
while $\sum_i \frac{A_i \bullet \Delta}{q_i} \geq \ln(\varepsilon)$ do
    Update $P$ to be: $P \leftarrow P + \frac{\tau^2 \varepsilon}{2m} \Delta$
    $\Delta \leftarrow$ FindPsdDelta($P, q, \{A_i\}$)
return $P$
```

The procedure FindPsdDelta is similar to the procedure FindDelta used in the linear variant.

Theorem 3 *For any constant $\varepsilon > 0$, the algorithm* HaploFreq2*(ε) finds a ε-approximate solution in polynomial time.*

Proof: The proof is similar in nature to the linear variant proof, with several technical points that need attention.

One technically concerns the amendment matrix Δ. Unfortunately, this matrix is not necessarily a PSD matrix, as the PSD cone is not closed under substraction.

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

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. $\sum_{i=1}^{n} \frac{A_i \Delta}{q_i} \geq \varepsilon$

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

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. Also notice that Δ_σ satisfies the easy conditions of being $(\varepsilon\sigma, \frac{\sigma}{\tau})$-good, that is the first few conditions above, from the fact that Y is a matrix that is a convex combination of two matrices that satisfy these constraints. In addition:

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

Since $\forall_i A(O-P) \leq 1$, we have $\left| \frac{A_i \Delta_l}{q_i} \right| \leq \frac{\sigma}{\tau}$.
□

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

$$\frac{f(P')}{f(P)} \geq e^{\frac{\tau^2 \varepsilon^2}{8n}}$$

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 a $(\varepsilon\sigma, \frac{\sigma}{\tau})$-good matrix, we have

$|c_i| \leq \frac{\sigma}{\tau}$. Therefore:

$$\begin{aligned}
\log\left(\frac{f(\vec{p'})}{f(\vec{p})}\right) \qquad & = \log\left(\prod_{i=1}^{n} \frac{A_i(P+\Delta)}{q_i}\right) & \\
& = \sum_{i=1}^{n} \log \frac{q_i + A_i\Delta}{q_i} & \\
& = \sum_{i=1}^{n} \log(1+c_i) & \\
& = \sum_{i=1}^{n} \left[\sum_{j=1}^{\infty} \frac{1}{j}(c_i)^j(-1)^{j+1}\right] & \text{by Taylor series expansion} \\
& \geq \sum_{i=1}^{n} \left[c_i - (c_i)^2\right] & \text{as } |c_i| < \tfrac{1}{2}\text{, see claim 6} \\
& \geq \sum_{i=1}^{n} c_i - \sum_{i=1}^{n}(c_i)^2 & \\
& \geq \sigma\varepsilon - n\frac{\sigma^2}{\tau^2} & \\
& \geq \frac{\tau^2\varepsilon^2}{4n} & \text{pick } \sigma = \frac{\tau^2\varepsilon}{8n}
\end{aligned}$$

□

It remains to show how to find a $(\varepsilon\sigma, \frac{\sigma}{\tau})$-good matrix efficiently.

Claim 3 *Suppose that there exists a $(\varepsilon\sigma, \frac{\sigma}{\tau})$-good matrix for every $\sigma \geq 1$. Then we can find a $(\varepsilon\sigma, \frac{\sigma}{\tau})$-good matrix for every $\sigma \geq 1$ in polynomial time.*

Proof: Let $W = P + \Delta$. We need Δ to satisfy the following semi-definite program:

1. $W := P + \Delta \succeq 0$
2. $\sum_{ij} W_{ij} = 1$ and $|W_{ii}| \leq 1$
3. $|\Delta_{ij}| \leq \frac{\sigma}{n^2}$ (this implies: $\forall i \left|\frac{A_i\Delta}{q_i}\right| \leq \frac{\sigma}{\tau}$)
4. maximize $\sum_{i=1}^{n} \frac{A_i\Delta}{q_i}$

And we are guarantied that the objective value is at least $\varepsilon\sigma$ if we are e^{ε} from the optimum. Notice that we can change the objective function to the simpler term:

5. maximize $\sum_{i=1}^{n} \frac{A_i\Delta}{q_i} = B \bullet \Delta$

And the semi-definite program above can be solved in polynomial time. In fact, it can be solved for $\sigma = 1$, and obtaining W we can create a $(\varepsilon\sigma, \frac{\sigma}{\tau})$-good matrix for every $\sigma \leq 1$ by defining:

$$W_\sigma = \sigma W + (1-\sigma)P$$

□

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

□

4 Lower Bounds

Strong hardness results for optimizing over polynomials are know, see [2]. We prove hardness for a more closely-related polynomial optimization problem.

Claim 4 *For every constant k, it is NP hard to approximate the maximum of a polynomial in n variables with $\{0,1\}$ coefficients of total degree k up to a factor $\Omega(k)$, under the restriction $0 \leq x_i \leq 1$.*

Proof: We reduce from Hyper-graph Vertex Cover. As shown in [5], it is NP-hard to decide if a k-uniform hyper graph has a vertex cover of size $(1+\varepsilon)T$ or if it's minimal VC is of size at least $(k-1-\varepsilon)T$.

Let $H=(V,E)$, $V=\{v_1,...,v_n\}$ be an instance of k-HGVC. We construct a corresponding polynomial $p_H(x_1,...,x_n)$, as:

$$p_H(x_1,...,x_n)=\sum_{i=1}^{n} x_i+\sum_{e\in E}\prod_{i\in e}(1-x_i)$$

The degree of $p_H(x_1,...,x_n)$ is obviously k. In addition, if H has a VC of size T, then the assignment:

$$x_i=\begin{cases} 1 & i\in VC \\ \\ 0 & o/w \end{cases}$$

Assigns the polynomial a value of exactly T. On the other hand, suppose that there exists a vector $\vec{x}$ such that $p_H(\vec{x})=l$. Then we claim that there exists a VC of size at most $20l$.

To see this, define a set $S\subseteq V$ according to the vector $\vec{x}$, such that:

$$\Pr[v_i\in S]=x_i$$

Then the expected size of S is $E[S]=\sum_i x_i\le p(\vec{x})=l$. Therefore, according to Markov we have that: $\Pr[S>10l]\le\frac{1}{l}$.
In addition, for a certain edge $e\in E$, notice that the probability that it is not covered by S is $\Pr[e\cap S=\phi]=\prod_{i\in e}(1-x_i)$. Therefore, the expected number of edges NOT covered by S (which is denoted by $NC=|\{e|e\cap S=\phi\}|$) is

$$E[NC]=\sum_e\prod_{i\in e}(1-x_i)\le p(\vec{x})=l$$

Again, according to Markov, we get that $\Pr[NC>10l]\le\frac{1}{10}$.
Therefore, the probability that both events happen is at least:

$$\Pr[(S<10l)\wedge(NS<10l)]>\frac{4}{5}$$

This gives rise to a VC of size at most $20l$ (by picking a vertex from each uncovered edge). □

We also show directly that the Quadratic Probability Estimation Problem is NP-hard:

Theorem 4 *The* Frequency Estimation for Unphased Genotypes *Problem is NP-hard.*

To prove this theorem we describe a reduction from the Clique problem. Given a graph $G=(V,E)$, we create an input for the Quadratic Probability Estimation Problem as follows:

We assign a variable x_i for each node v_i in G. Then the set of matrices will consist only of a single matrix, which is the adjacency matrix of G.

The following claim implies the theorem:

Claim 5 *The solution to the instance of QPE created has value $\frac{1}{2}(1-\frac{1}{r})$ if and only if the maximal clique size of G was r.*

Proof: If the graph contains a clique of size r, then by assigning $\frac{1}{r}$ to all vertices corresponding to this clique, we obtain a value of $\frac{1}{2}(1-\frac{1}{r})$.

We proceed to prove even a stronger statement, which is a generalization of Turán's theorem. The statement is that 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 $\frac{1}{2}(1-\frac{1}{r})$.

As a first step, we explore the properties of an optimal solution to the reduced instance. Consider any two vertices v_i,v_j and their corresponding variables x_i,x_j. If x_i,x_j are non-zero, then changing them by ε (say $x_i\mapsto x_i+\varepsilon$ and $x_j\mapsto x_j-\varepsilon$) will have the following effect on the objective function (where Γ_i denotes set of neighbors of v_i):

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

This implies that for all x_i, x_j that are non-zero, the optimal solution satisfies $N_i = N_j$, where N_i is the sum of all weights of the vertices in $\Gamma(i)$.

We now prove the claim using this observation. Let G be a graph with objective function value $> \frac{1}{2}(1-\frac{1}{r})$. It suffices to show that the maximal clique size in G is $\omega(G) > r$. Notice that we can assume that $N_i > 1-\frac{1}{r}$. This is because the optimum is bounded by $\frac{1}{2}\sum_i x_i N = \frac{1}{2}N$, and if $N \leq 1-\frac{1}{r}$, we get that the optimum is bounded by $\frac{1}{2}(1-\frac{1}{r})$ in contradiction to our initial assumption.

Observe the subgraph of all non-zero vertices, and the largest clique amongst them. Suppose its size is k, and let the participating vertices have variables $x_1, ..., x_k$. Then the set of neighbors of these vertices satisfy:

$$\sum_{i=1}^{k} N_i = \sum_{i=1}^{k} \sum_{j \in N_i} x_j \leq (k-1)\sum_j x_j \leq k-1$$

Where the first inequality follows since each variable can be a neighbor of only $k-1$ vertices of the clique (as otherwise we would have a clique of larger size). In addition, we know that $\sum_{i=1}^{k} N_i = k \cdot N > k(1-\frac{1}{r})$. Taking both facts into account we have:

$$k(1-\frac{1}{r}) < k-1 \Rightarrow k > r$$

Hence we conclude that the largest clique is of size strictly larger then r. □

Corollary 1 *It is NP-hard to approximate* Frequency Estimation for Unphased Genotypes *to within* $2^{n^{1-\varepsilon}}$ *for every constant* $\varepsilon > 0$.

Proof: The above theorem holds for a very degenerate instance of Frequency Estimation for Unphased Genotypes, namely with only one input matrix. Looking closely, and using previous hardness of approximation results for the Clique, it follows that Frequency Estimation for Unphased Genotypes is hard to approximate to within:

$$\frac{(1-n^{-1+\varepsilon})}{(1-n^{-\delta})} = 1 + \frac{n^{-\delta} - n^{-1+\varepsilon}}{1-n^{-\delta}} \geq 1 + \frac{1}{\sqrt{n}}$$

(ε, δ are small constants)

Now amplify this construction by repeating the same matrix of the input graph M times. The size of the instance built is Mn, and the hardness of approximation becomes:

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

Taking M to be a large polynomial, and rephrasing in terms of input size yields the result. □

5 Experimental Results

We implemented the algorithms HaploFreq and HaploFreq2 (described in Sections 2.2 and 3.2 respectively) and compared them to the widely used software PHASE [23].

Implementation details. Both HaploFreq and HaploFreq2 assume that the number of possible haplotypes is limited - and usually small. This is usually the case, but when we consider a region spanning more than twenty SNPs the number of possible haplotypes may affect the running time of the algorithms considerably. We therefore use a preprocessing mechanism which filters out unreasonable haplotypes. The preprocessing mechanism is based on a greedy procedure, similar to the one given in [13]. After the preprocessing we are typically left with about 50 possible haplotypes. We then run our algorithms on those 50 haplotypes.

Another crucial issue which one has to overcome is the use of semidefinite programming in HaploFreq2. Recall that in each iteration of HaploFreq2 we have to solve a semidefinite program. Even though semidefinite programs can be solved in polynomial time, in practice they are very slow. We therefore implemented a semidefinite programming solver which is specifically tailored for our needs. This semidefinite solver runs faster on the instance given by the HaploFreq2 than other semidefinite programming solvers such as SDPPack [1]. The details of the algorithm of the SDP solver can be found in [**?**]. Furthermore, since HaploFreq2 only uses the solution of the semidefinite program in order to find an improved solution, it is sufficient to efficiently find a sub-optimal solution to the semidefinite program, as long as the solution gives an improvement over the current point. such an improvement can be found using our SDP solver very efficiently.

Due to these implementation optimization, our programs are 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 1 gives a concise comparison of measured running times of PHASE, HaploFreq and HaploFreq2.

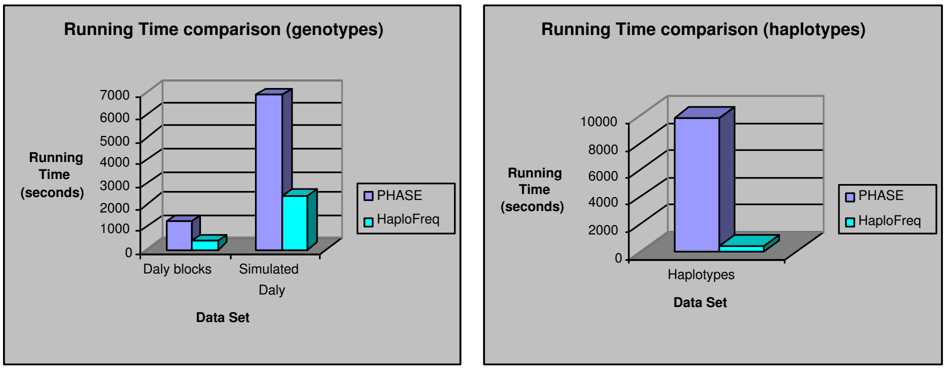

Figure 1: Running times comparison

The data sets. We applied our algorithms to to the data set of Daly et al. [4, 22] and to population D of Gabriel et al. [8]. The first data set 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 (about 10%) is missing with an average of 10 SNPs per individual's genotype missing. This data set was partitioned in [4, 22] into eleven 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 [8] data 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.

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, $\{p_1, \ldots, p_k\}$ and $\{q_1, \ldots, q_k\}$, 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.

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 data sets by picking haplotypes randomly and independently from that distribution. We then added randomly scattered missing data and random 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.

Accuracy of estimations. We compared the accuracy of the frequency estimations of our HaploFreq to PHASE [23]. 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 data sets.

Additionally, we compared our algorithms over the simulated data sets 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 data sets, and we found again that HaploFreq is typically $10 - 50\%$ more accurate than PHASE. We note that both PHASE and HaploFreq are much closer to the sampled distribution than to the underlying population distribution. A complete summary of the comparison can be found in figures 2,3,4.

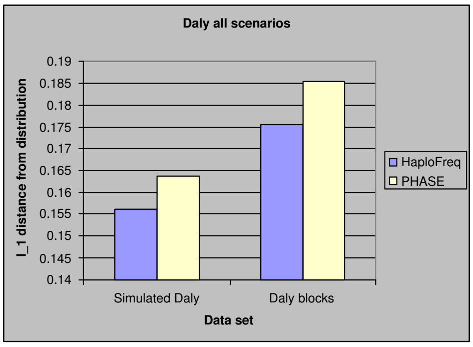

Figure 2: Average l_1 distance from the actual distribution on the Daly data sets (both blocks and simulated)

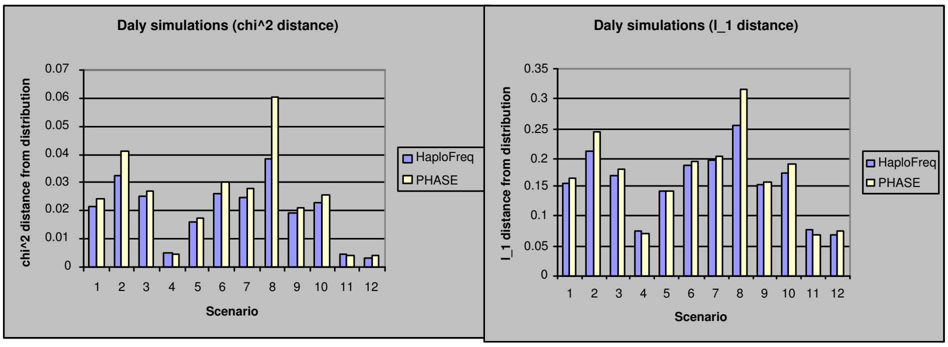

Figure 3: Average chi^2 and l_1 distances from the actual distribution on the simulated Daly data, with various simulation parameters

The scenarios depicted above are:

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 1: simulation parameters for the simulated Daly data set

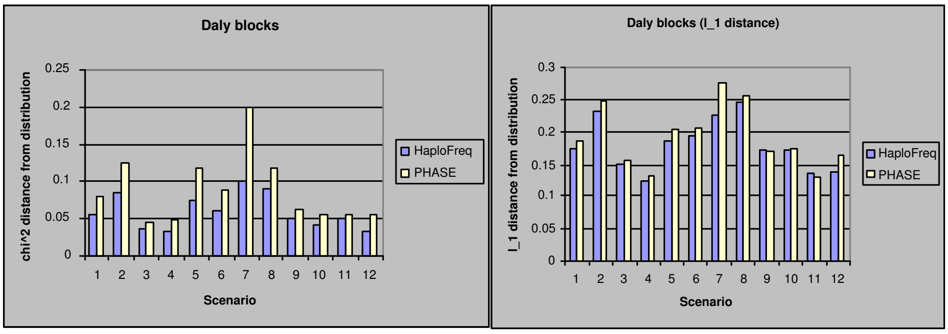

Figure 4: Average chi^2 and l_1 distances from the actual distribution on the Daly blocks data, with various simulation parameters

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

Table 2: simulation parameters for the Daly blocks data set

References

[1] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5:13–51, 1995.

[2] M. Bellare and P. Rogaway. The complexity of approximating a quadratic program. 409, 1992.

[3] AG Clark. Inference of haplotypes from pcr-amplified samples of diploid populations. *Journal of Molecular Biology and Evolution*, 7(2):111–22, Mar 1990.

[4] MJ Daly, JD Rioux, SF Schaffner, TJ Hudson, and ES Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29(2):229–32, Oct 2001.

[5] Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered pcp and the hardness of hypergraph vertex cover. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 595–601. ACM Press, 2003.

[6] L Excoffier and M Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):921–7, Sept 1995.

[7] D. Fallin and NJ. Schork. Accuracy of haplotype frequency estimation for biallelic loci, via the expectation-maximization algorithm for unphased diploid genotype data. *American Journal of Human Genetics*, 67:947–959, 2000.

[8] GB. Gabriel, SF. Schaffner, H. Nguyen, JM. Moore, J. Roy, B. Blumenstiel, J. Higgins, M. DeFelice, A. Lochner, M. Faggart, SN. Liu-Cordero, C. Rotimi, A. Adeyemo, R. Cooper, R. Ward, ES. Lander, MJ. Daly, and D. Altshuler. The structure of haplotype blocks in the human genome. *Science*, 296:2225–2229, 2002.

[9] Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In *Proceedings of the 6th Annual International Conference on (Research in) Computational (Molecular) Biology*, 2002.

[10] D Gusfield. A practical algorithm for optimal inference of haplotypes from diploid populations. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 2000.

[11] D Gusfield. Inference of haplotypes from samples of diploid populations: complexity and algorithms. *Journal of Computational Biology*, 8(3):305–23, 2001.

[12] E. Halperin and E. Eskin. Haplotype reconstruction from genotype data using imperfect phylogeny. *Bioinformatics*, 2004.

[13] E. Halperin and R. Karp. The minimum-entropy set cover problem. Manuscript, 2003.

[14] ME Hawley and KK Kidd. Haplo: a program using the em algorithm to estimate the frequencies of multi-site haplotypes. *Journal of Heredity*, 86(5):409–11, Sep-Oct 1995.

[15] L. G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Math. Phys.*, 20:53–72, 1980.

[16] G. Lancia, V. Bafna, S. Istrail, R. Lippert, and R. Schwartz. Snps problems, algorithms and complexity, european symposium on algorithms. In Springer-Verlag, editor, *Proceedings of the European Symposium on Algorithms (ESA-2001), Lecture Notes in Computer Science*, volume 2161, pages 182–193, 2001.

[17] JC Long, RC Williams, and M Urbanek. An e-m algorithm and testing strategy for multiple-locus haplotypes. *American Journal of Human Genetics*, 56(3):799–810, Mar 1995.

[18] S. Michalatos-Beloin, SA. Tishkoff, KL. Bently, KK. Kidd, and G. Ruano. Molecular haplotyping of genetic markers 10 kb apart by allele-specific long-range pcr. *Nucleic Acids Res*, 24:4841–4843, 1996.

[19] NIH. Large-scale genotyping for the haplotype map of the human genome. RFA: HG-02-005, 2002.

[20] Niu, Qin, Xu, and Liu. In silico haplotype determination of a vast set of single nucleotide polymorphisms. Technical report, Department of Statistics, Harvard University, 2001.

[21] N Patil, AJ Berno, DA Hinds, WA Barrett, JM Doshi, CR Hacker, CR Kautzer, DH Lee, C Marjoribanks, DP McDonough, BT Nguyen, MC Norris, JB Sheehan, N Shen, D Stern, RP Stokowski, DJ Thomas, MO Trulson, KR Vyas, KA Frazer, SP Fodor, and DR Cox. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294(5547):1719–23, Nov 23 2001.

[22] JD Rioux, MJ Daly, MS Silverberg, K Lindblad, H Steinhart, Z Cohen, T Delmonte, K Kocher, K Miller, S Guschwan, EJ Kulbokas, S O'Leary, E Winchester, K Dewar, T Green, V Stone, C Chow, A Cohen, D Langelier, G Lapointe, Gaudet D, J Faith, N Branco, SB Bull, RS McLeod, AM Griffiths, A Bitton, GR Greenberg, ES Lander, KA Siminovitch, and TJ Hudson. Genetic variation in the 5q31 cytokine gene cluster confers susceptibility to crohn disease. *Nature Genetics*, 29(2):223–8, Oct 2001.

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

A Useful facts

Claim 6 *If $|x| \leq \frac{3}{5}$ then:*

$$\sum_{j=1}^{\infty} \frac{1}{j} x^j (-1)^{j+1} \geq x - x^2$$

Proof: For $x > 0$ it suffices to show that:

$$\frac{1}{2}x^2 \geq -\frac{1}{3}x^3 + \frac{1}{4}x^4 - \frac{1}{5}x^5 + ...$$

And this is obviously true since the RHS is negative as long as $|x| < 1$. If $x < 0$, denote $x = -y$ for $y = |x| > 0$, then we need to show that $\frac{1}{2}y^2 - \frac{1}{3}y^3 - \frac{1}{4}y^4 - \frac{1}{5}y^5 - ... \geq 0$, and indeed:

$$\begin{aligned} \frac{1}{3}y^3 + \frac{1}{4}y^4 + \frac{1}{5}y^5 + ... &\leq \tfrac{1}{3}y^3 \left[1 + y + y^2 + y^3 + ...\right] \\ &= \tfrac{1}{3}y^3 \tfrac{1}{1-y} \\ &= \tfrac{1}{2}y^2 \qquad \text{as long as } y \leq \tfrac{3}{5} \end{aligned}$$

□