A Survey of Computational Methods for Determining Haplotypes

Bjarni V. Halldórsson, Vineet Bafna*, Nathan Edwards, Ross Lippert, Shibu Yooseph**, and Sorin Istrail

Informatics Research, Celera Genomics/Applied Biosystems,
45 W. Gude Drive, Rockville MD, 20850
{Bjarni.Halldorsson, Nathan.Edwards, Ross.Lippert,
Sorin.Istrail}@celera.com

Abstract. It is widely anticipated that the study of variation in the human genome will provide a means of predicting risk of a variety of complex diseases. Single nucleotide polymorphisms (SNPs) are the most common form of genomic variation. Haplotypes have been suggested as one means for reducing the complexity of studying SNPs. In this paper we review some of the computational approaches that have been taking for determining haplotypes and suggest new approaches.

1 Introduction

Genomes can be considered to be a collection of long strings, or sequences, from the alphabet {A,C,G,T}. Each element of the alphabet encodes one of four possible *nucleotides*. With the completion of the sequencing of the human genome, efforts are underway to catalogue genomic variations across human populations. *Single Nucleotide Polymorphisms* or SNPs constitute a large class of these variations. A SNP is a single base pair position in genomic DNA at which different nucleotide variants exist in some populations; each variant is called an *allele*. In human, SNPs are almost always biallelic; that is, there are two variants at the SNP site, with the most common variant referred to as the *major allele*, and the less common variant as the *minor allele*. Each variant must be represented in a significant portion of the population to be useful.

Diploid organisms, such as humans, possess two nearly identical copies of each chromosome. In this paper, we will refer to a collection of SNP variants on a single chromosome copy as a *haplotype*. Thus, for a given set of SNPs, an individual possesses two haplotypes, one from each chromosome copy. A SNP site where both haplotypes have the same variant (nucleotide) is called a *homozygous* site; a SNP site where the haplotypes have different variants is called a *heterozygous* site. The conflated (mixed) data from the two haplotypes is called

* Current address: University of California, San Diego, Computer Science & Engineering, La Jolla, CA, 92093 *vbafna@cs.ucsd.edu*

** Current address: Institute for Biological Energy Alternatives, 1901 Research Blvd., 6th floor, Rockville, MD 20850, *shibu.yooseph@bioenergyalts.org*

S. Istrail et al. (Eds.): SNPs and Haplotype Inference, LNBI 2983, pp. 26–47, 2004.

a *genotype*. Thus, in genotype data, while the nucleotide variants at homozygous and heterozygous sites are known, the information regarding which heterozygous site SNP variants came from the same chromosome copy, is unknown. See Figure 1 for an example of these concepts. Haplotypes play a very important role in several areas of genetics, including mapping complex disease genes, genome wide association studies, and also in the study of population histories. Unfortunately, current experimental techniques to infer the haplotype of an individual are both expensive and time consuming. However, it is possible to determine the genotype of an individual quickly and inexpensively. Computational techniques offer a way of inferring the haplotypes from the genotype data.

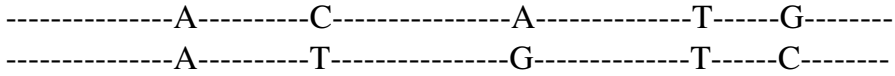

Fig. 1. Two sequences from the same region on two nearly identical copies of a chromosome of an individual. Only the SNPs have been shown with the non-SNP positions labeled with a "-". In this example there are five SNPs. The first and the fourth SNP sites are homozygous, and the remaining three SNP sites are heterozygous. The individual has the two haplotypes $ACATG$ and $ATGTC$; the genotype is $A\{C,T\}\{A,G\}T\{G,C\}$.

Out of the two nearly identical copies of each chromosome in an individual, one copy is inherited from the paternal genome and the other copy from the maternal genome. This simple picture of inheritance is complicated by a process known as *recombination*, which takes place during meiosis - a process involved in the formation of reproductive cells (or *gametes*) in the parents. During recombination, portions of the paternal and maternal chromosomes are exchanged (Figure 2). Recombination can result in haplotypes in offspring that are different from those in the parents. The site on the chromosome where a recombination occurs is called a recombination site. On average, one or two recombinations occur per chromosome per generation [33].

In population studies, it has been shown that the likelihood that a site will act as a recombination site is not uniform across a chromosome[33], recombination sites occur much more frequently than expected in certain chromosomal regions and much less frequently in other chromosomal regions. Regions of high recombination site frequency are called *recombination hotspots*. Several recent studies [9,29,44] have suggested that human genetic variation consists largely of regions of low recombination site frequency, delineated by regions of high recombination site frequency, resulting in *blocks* of SNPs organized in mini-haplotypes.

An assumption that underlies much of population genetics, called the *infinite sites model*, requires that the mutation that results in a SNP occur only once in the history of a population, and therefore, that all individuals with a variant allele must be descendants of a single ancestor. While the infinite sites model is clearly a simplification of the true mechanism of genetic mutation, models of

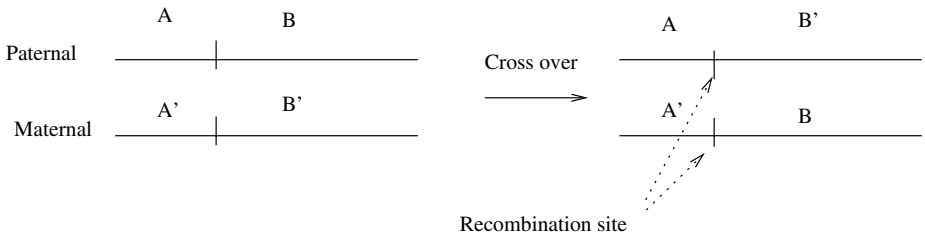

Fig. 2. An illustration of the recombination process that occurs during meiosis. Recombination is characterized by a cross-over event in which a portion of a paternal chromosome is exchanged with a portion of a maternal chromosome. This can result in the offspring having different haplotypes from those in the parents.

genetic variation built under this assumption compare favorably with empirical population genetics studies. Some of the models and algorithms in the text to follow will assume an *infinite sites model*.

In this paper we present an overview of some of the computational approaches that have been taken for determining haplotypes. This survey is split into two parts, first approaches for haplotype phasing are presented and then approaches for haplotype assembly.

2 The Haplotype Phasing Problem

In this section, we consider the problem of haplotype phasing: Given a set of genotypes, find a *good* set of haplotypes that resolve the set.

Generically the haplotype phasing problem can be posed as:

Haplotype Phasing (generic)
Input: A set G of genotypes.
Output: A set H of haplotypes, such that for each $g \in G$ there exists $h_1, h_2 \in H$ such that the conflation of h_1 with h_2 is g.

An alternate related problem is haplotype frequency estimation. In this problem we care primarily about estimating the frequency of each potential haplotype in the population, and less so about the phasings of particular individuals.

By typing genetically related individuals one can get a better estimate of haplotypes present since the haplotype pair of a child is constrained by its inheritance from his parents [35,36]. This version of the problem is considered in various software packages [1]. In this paper, we assume that such pedigree data is not available to us, however recasting the problems presented here in the presence of pedigree data is a worthwhile avenue of research.

Haplotype phasing has a variety of applications, each of which warrant different methodologies. Coarsely, one can partition haplotype phasing problems into three classes, based on their tractability:

Small The number of sites is small enough that solutions requiring exponential space or time in it would be practical. It is sufficient for analyzing the SNPS in the vicinity of a single gene.

Medium The number of sites is small enough that methods which are polynomial in the number of sites and individuals are practical. Number of individuals and number of sites may be on the order of $100's$. This size roughly corresponds to the number of SNPs across a region spanning several genes.

Large Chromosome size, where algorithms which are linear in the number of SNPs are the only ones practical. The number of sites could be in the tens of thousands while the number of individuals sampled is small.

Additionally, many of the population genetics assumptions that hold for the small problems will not extend easily to the medium and large problems where the effects of recombination become significant. Different measures of success are appropriate depending on the problem size. Given a set of genotypes with a priori phasing information, a natural questions to ask is whether the algorithm retrieves the correct phasing. For small and medium problems, appropriate measures include the number of haplotypes that are predicted correctly or the difference in population frequency of the haplotypes in the known and the predicted set. For very large problems it is likely that these measures will be blunt and all methods will not perform well. An alternate measure suggested in [37] is the number of crossovers to explain the correct haplotypes from the predicted haplotypes.

When presenting the problems, we will assume that the genotype information we have is accurate. However, in practice, this is not the case, current genotyping technologies will fairly frequently not call genotypes (missing data) and less frequently miscall a genotype (wrong data). A practical algorithm needs to deal with these problems, in particular the missing data problem. The discussion in this paper is in terms of SNP's, most of the results and methods also will apply, perhaps with some modification, to studies of alternate genetic variations (markers) such as microsatellites.

Notation We will follow notation by Gusfield [19] for haplotypes and genotypes. We will arbitrarily label the two alleles of any SNP 0 and 1. A genotype, representing a pair of haplotypes, can take three values for each SNP, corresponding to the observation of $\{0\}, \{1\}, \{0,1\}$. To simplify notation we will use 0 for $\{0\}$, 1 for $\{1\}$ and 2 for $\{0,1\}$. We will say that a SNP is *ambiguous* in a genotype if it has value 2. A genotype is *ambiguous* if it contains more than one *ambiguous* SNP.

We will generally use subscripts for objects associated with haplotypes and superscripts for objects associated with genotype. For example, the probability of observing the genotype g in a given population might be given as ϕ^g and the haplotype probabilities as ϕ_h. Since superscripts are possibly confused with exponentiation, explicit parentheses will be placed around exponentiated quantities to disambiguate this.

We will use $+$ to denote conflation and write $h + \bar{h} = g$ if the conflation of h and $\bar{h}$ is g. To capture the notion that two haplotypes combine to make a

genotype, we will, when convenient to do so, use the Kronecker delta, $\delta^g_{h+\bar{h}} = 1$ if $h + \bar{h} = g$ and 0 else.

We will denote the number of genotypes with n and the number of SNP sites with m.

2.1 Clark's Rule

In a seminal paper[7], Clark proposed a common sense approach to phasing, that has become known as *Clark's rule*. Clark's rule is an inference method that resolves genotypes to their haplotype pairs. First, all homozygous and single ambiguous site genotypes are identified. The haplotypes that phase these genotypes are completely determined, forming an initial set of haplotypes supported by the data. Given a set of haplotypes H representing the resolved genotypes, Clark's rule finds $g \in G$ and $h \in H$ such that $g = h + \bar{h}$ for some $\bar{h}$. The haplotype $\bar{h}$ is added to H. The process continues until either all genotypes are resolved, or no suitable pair of unresolved genotype and resolving haplotype (g, h) exists.

Note that it may not even be possible to get this algorithm started if there are no homozygous or single ambiguous site genotypes. Further, there is no guarantee that a particular sequence of applications of Clark's rule will resolve all genotypes. Genotypes that remains unresolved after a maximal sequence of applications of Clark's rule are called *orphans*.

It should be clear from the description of Clark's rule that it describes a *class* of algorithms, each of which uses a different protocol for selecting a genotype-haplotype pair from which to infer a (typically) new haplotype. Clark's paper applies a greedy approach, in which the known haplotypes are tested against the unresolved genotypes in turn. The first genotype that Clark's rule can be applied to is resolved, potentially adding a new haplotype to the set of known haplotypes for the next iteration.

It is natural to ask for a Clark's rule application sequence that results in the fewest number of orphans. Clark's experiments [7] on real and simulated data suggest that the sequence of applications of Clark's rule that resolves the most genotypes generates fewest incorrect haplotype assignments.

Problem 1 (Minimizing Orphans). Find a sequence of Clark's rule applications that results in the fewest orphans.

Biological intuition about the nature of haplotypes present in human populations prompt us to think about versions of problem 1 that produce solutions that respect this intuition.

Problem 2 (Maximizing Unique Resolutions). Find a sequence of Clark's rule applications that maximizes the number of resolutions subject to the constraint that the final set of haplotypes must provide a single unique resolution to each genotype.

Problem 3 (Minimizing Inference Distance). Find a sequence of Clark's rule applications that minimizes the number of Clark's rule applications necessary to generate the genotypes' haplotypes.

Gusfield [19,20] studied a slightly restricted version of this problem, in which each genotype can participate in at most one Clark's rule application. Gusfield showed that finding an optimal Clark's rule application sequence is NP-hard, but that in practice, on medium-sized instances, this version of the problem can be solved by a combination of careful enumeration and linear programming. Gusfield also evaluated the effectiveness of an algorithm incorporating a greedy application of Clark's rule with mixed results.

2.2 Maximum Likelihood

Hardy-Weinberg equilibrium (HWE) is the condition that the probability of observing a genotype is equal to the product of the probabilities of observing its constituent haplotypes (see [23]). Under this hypothesis, the probability of genotype g in the population is related to the haplotype probabilities by the compact expression

$$\phi^g = \sum_{h+\bar{h}=g} \phi_h \phi_{\bar{h}}$$

where ϕ_h is the probability of haplotype h in the population.

The *maximum likelihood method* of [15,24,39,52,16,1] estimates the haplotype probabilities $\phi_H = (\phi_h, \phi_{\bar{h}}, \ldots, \phi_{h'})$ from observed genotype frequencies $\hat{\phi}^G$ in n individuals. The approach assumes HWE and a uniform prior on the ϕ_h's. The likelihood function of the observed is then

$$L(\phi_H) = \prod_{g \in G} (\phi^g)^{n\hat{\phi}^g} \tag{1}$$

where $\phi^g = \sum_{h+\bar{h}=g} \phi_h \phi_{\bar{h}}$. The estimated ϕ_H is a maximum of L subject to the constraints that $\sum_{h \in H} \phi_h = 1$ and $\phi_h \geq 0$, $\forall h \in H$.

There is a great deal of literature on the maximization of this polynomial, for example the method of *Expectation Maximization* is a linearly convergent method guaranteed to locate a local maximum of L from almost every (feasible) starting point.

However, a naïve implementation of the EM method requires exponential space, since there are 2^m unknown haplotype probabilities which must be stored for m variant sites. One notes note that, for n sampled individuals, $\Omega(n)$ haplotypes are expected to have significant probability. An efficient way to discover those haplotypes which contribute significantly to the maximizer of L would make this approach much more efficient.

Problem 4 (Haplotype Support Problem). Given observed genotype frequencies ϕ^g, and $\epsilon > 0$, find $H' \subset H$, such that one can guarantee that there exists a ϕ_H that is a global maximizer of L and that $h \notin H'$ implies $\phi_h < \epsilon$.

The Phasing Polynomial We will now give a combinatorial interpretation of L. We assume that ϕ^G comes from counts of individual observed genotypes, and thus $n\phi^g$ is integral for each genotype g. We may then formulate L in terms of a product over n observed individual genotypes g_i $(1 \leq i \leq n)$, i.e.

$$L = \prod_{i=1}^{n} \phi^{g_i} = \prod_{i=1}^{n} \Big(\sum_{h+\bar{h}=g_i} \phi_h \phi_{\bar{h}} \Big)$$

Interchanging product and summation this becomes

$$L = \sum_{h_1,h_2,\cdots h_{2n}} \delta^{g_1}_{h_1+h_2} \delta^{g_2}_{h_3+h_4} \cdots \delta^{g_n}_{h_{2n-1}+h_{2n}} \phi_{h_1} \phi_{h_2} \cdots \phi_{h_{2n}}$$

Let an *explanation* of the genotypes $\boldsymbol{g} = (g_1, \ldots, g_n)$ be a sequence of $2n$ haplotypes $\boldsymbol{h} = (h_1, h_2, \ldots, h_{2n})$ such that $h_{2i-1} + h_{2i} = g_i$. Then the polynomial above can be more compactly expressed as

$$L = \sum_{\boldsymbol{h} \text{ explains } \boldsymbol{g}} \phi_{h_1} \phi_{h_2} \cdots \phi_{h_{2n}}$$

with the sum ranging over all explanations of $\boldsymbol{g}$. The likelihood function is a polynomial with a term of coefficient 1 for each possible explanation of the observed genotypes. Thus, a solution to the genotype phasing problem corresponds to a particular term in this polynomial.

The maximum likelihood approach seeks frequencies ϕ_H which maximize L. This problem is known to be NP-hard [26]. Also note that the problem does not directly address the problem of computing the individual phasings for each genotype. However, approximations can be made which recover the combinatorial nature of the phasing problem.

A Discrete Approximation Let us collect the terms of L, and use a multi-index $\boldsymbol{P}$ (a vector of non-negative integers indexed by H) to keep track of the exponents, then

$$L = \sum_{\boldsymbol{P}} K(\boldsymbol{P}, \boldsymbol{g}) \prod_{h \in H} (\phi_h)^{P_h},$$

where $K(\boldsymbol{P}, \boldsymbol{g})$ denotes the number of explanations of the observed genotype counts $\boldsymbol{g}$ which have $\boldsymbol{P}$ haplotype counts.

Since the ϕ_h are constrained to lie between 0 and 1, most of the terms in L are expected to be small. We may approximate L with its largest term:

$$L \sim L_{MAX} = \max_{\boldsymbol{P}} \left\{ K(\boldsymbol{P}, \boldsymbol{g}) \prod_{h \in H} (\phi_h)^{P_h} \right\}.$$

The maximization of L_{MAX} with respect to the ϕ_h is trivial, since any monomial $\prod_{h\in H}(\phi_h)^{P_h}$ in probabilities, ϕ_h, is maximized by $\phi_h = P_h/(2n)$. Thus

$$\max_{\phi_H} L \sim L_{MAX} = \max_{\boldsymbol{P}} \left\{ (2n)^{-2n} K(\boldsymbol{P}, \boldsymbol{g}) \prod_h (P_h)^{P_h} \right\}.$$

Thus, we see that the maximization of the maximal term of the likelihood polynomial reduces to a discrete problem. The solution of this problem does not give a phasing, but a collection of possible phasings with identical counts. The solution may also be a good initial point for an iterative maximum likelihood method, such as expectation maximization.

The objective function in this optimization problem is

$$F(\boldsymbol{P}, N\hat{g}^k) = K(\boldsymbol{P}, G) \prod_h (P_h)^{P_h},$$

where $\sum_i P_i = 2N$, which counts the number of ways to select $2N$ haplotypes from a bag with counts $\boldsymbol{P}$ *with replacement* to form an explanation of the genotypes G.

We are not aware of any results about the complexity of evaluating F or its maximum. In fact, there is a feasibility problem to which we have found no easy answer as well.

Problem 5 (Haplotype Count Feasibility). Given genotypes $\boldsymbol{g} = (g_1, \ldots, g_n)$ and a vector of counts $\boldsymbol{P}$ over H, decide whether there exists an explanation of $\boldsymbol{g}$ with counts $\boldsymbol{P}$.

Problem 6 (Counting Arrangements $K(\boldsymbol{P}, \boldsymbol{g})$). Given genotypes $\boldsymbol{g} = (g_1, \ldots, g_n)$ and a vector of counts $\boldsymbol{P}$, count how many distinct explanations, $\boldsymbol{h} = (h_1, h_2, \ldots, h_{2n-1}, h_{2n})$, exist for $\boldsymbol{g}$ with counts $\boldsymbol{P}$.

Problem 7 (Maximizing Arrangements). Given $\boldsymbol{g} = (g_1, \ldots, g_n)$, find counts $\boldsymbol{P}$, such that $K(\boldsymbol{P}, \boldsymbol{g}) \prod (P_h)^{P_h}$ is maximized.

Links to Clark's Rule One method for breaking ties in the application Clark's rule is to allow the haplotype frequencies to serve as probabilities, and randomly select g's and h's to which to apply it. In such a scheme, one would still resolve the homozygotes and the single-site heterozygotes, since they are unambiguous, but, when faced with a choice between multiple phasings, one randomly selects the phasing $h, \bar{h}$ with probability $\phi_h \phi_{\bar{h}} / \phi^{h+\bar{h}}$. Since this procedure is still strongly dependent on the order of consideration for the ambiguous genotypes, one draws them, and re-draws them, uniformly at random, from the sampled individuals.

This process can be viewed as a means to generate a sample from a stationary point of the maximum likelihood function. To see this, we view the individual samples as all having some phase, which we rephase through random applications

of Clark's rule with random tie-breaking as above. In the continuum limit, new instances of haplotype h are introduced at a rate

$$\Delta\phi_h = \sum_{g,\bar{h}:h+\bar{h}=g} \frac{\hat{\phi}^g}{\phi^g}\phi_h\phi_{\bar{h}} \tag{2}$$

where $\phi^g = \sum_{h+\bar{h}=g} \phi_h\phi_{\bar{h}}$, while instances of haplotype h are being removed (by individuals with haplotype h being re-drawn) at a rate

$$\Delta\phi_h = -\phi_h.$$

A steady state occurs when the two processes balance, i.e.

$$\phi_h = \sum_{g,\bar{h}:h+\bar{h}=g} \frac{\hat{\phi}^g}{\phi^g}\phi_h\phi_{\bar{h}}$$

which is a sufficient condition for ϕ_H to be a local maximum of the likelihood function of equation 1. Thus, the haplotypes sampled in this random application of Clark's rules are distributed according to some stationary distribution of L.

2.3 Clark's Rule and Population Models

The observation that the maximum likelihood method could be modeled by a certain probabilistic application of Clark's rule was known to researchers, Stephens, Smith, and Donnelly [50], who proposed a modification of the ML sampling procedure of the previous section. Their modification introduces an approximate population genetics model [48] as a prior for observing the set of haplotypes.

Instead of phasing randomly selected individuals with probabilities weighted by $\phi_h\phi_{\bar{h}}$, they proposed a more complicated probability rule, where the weight of phasing $h, \bar{h}$ for g is given by

$$\delta^g_{h+\bar{h}}\pi_h(\boldsymbol{h}\backslash h)\cdot\pi_{\bar{h}}(\boldsymbol{h}\backslash h\backslash\bar{h}) \tag{3}$$

where $\boldsymbol{h}\backslash h$ is the sequence of haplotypes $\boldsymbol{h}$ with one occurrence of h removed. The function $\pi_h(\boldsymbol{h})$ approximates the probability that haplotype h might be generated either by direct descent or mutation from a population with haplotype counts of $\boldsymbol{h}$.

It should be noted that equation 2 applies only when N is much larger than the number of haplotype variants in the population. Thus it is not strictly applicable for small populations where a substantial portion of variants occur only once. It is not an issue for this Markov Chain Monte Carlo (MCMC) approach.

The algorithm they propose is to iteratively modify the explanation of the given genotypes, selecting the explaining haplotypes $h, \bar{h}$ for a random individual with genotype g, and replacing that pair with a pair generated randomly with weights from equation 3, updating the current frequencies, ϕ_H, of the variants in the sample. Statistics of the sampled set of phasings are then used to select the phasings of the individuals.

It remains to define the approximation of $\pi_h(\boldsymbol{h})$, for which they propose

$$\pi_h(\boldsymbol{h}) = \sum_{h'}(h \cdot \boldsymbol{h})(I - \frac{\theta}{2N+\theta}M)^{-1}_{hh'} \tag{4}$$

where $h \cdot \boldsymbol{h}$ counts the number of occurrences of h in $\boldsymbol{h}$, θ is an estimate for the per site mutation rate, I is the identity matrix, and M is the single site mutation matrix, $M_{hh'} = 1$ iff h and h' have exactly one mismatch and 0 otherwise. This is an approximation to the probability that h can come from some h' after a geometrically distributed number of single site mutations. This approximation arose from considering a random population model in [28]. It should be noted that while the matrix M appears to be of exponential size, an arbitrary element of $(I - \frac{\theta}{2N+\theta}M)^{-1}$ can be computed in $O(m)$ time.

An implementation of this algorithm by Stephens, Smith, and Donnelly is *PHASE* [50,49]. An alternative implementation, which more closely follows the maximum likelihood method was produced by Niu et al [42]. *PHASE* works well on medium problems with a small population.

2.4 Parsimony Formulations

Extending Clark's basic intuition that unresolved haplotypes are to look like known ones, a variety of parsimony objectives can be considered.

In the context of haplotype phasing, the most *parsimonious* phasing refers to the solution that uses the fewest haplotypes. Hubbell [27] showed that this version of the problem is NP-hard, in general, by a reduction from minimum clique cover. Gusfield [22] solved the problem via an (exponentially large) integer programming formulation that is solvable in many cases, even for medium-sized problems.

Phasing via Integer Programming Integer programming, as employed by Gusfield [22] to find the most parsimonious phasing, is a very effective solution technique for optimization problems with combinatorial structure, even when they are NP-hard. As Gusfield demonstrated, with a suitable integer programming formulation and a powerful solver package, many realistic instances can be solved. The tractability of such integer programs for a given instance depends critically on the nature of the integer programming formulation for the problem.

We present two integer programming formulations for the haplotype phasing problem.

Formulation 1 (Gusfield [22]) *Given genotypes G, we define the set of possible haplotypes $\hat{H} = \{h | \exists g, \bar{h} \text{ s.t. } g = h + \bar{h}\}$. Further, for each genotype $g \in G$ we define the set of valid phasings, $P_g = \{\{h, \bar{h}\} | g = h + \bar{h}; h, \bar{h} \in \hat{H}\}$.*

In this formulation, we use an indicator variable x_h for each haplotype $h \in \hat{H}$ and an indicator variable y_{gp} for each genotype $g \in G$ and $p \in P_g$.

$$
\begin{aligned}
&\min \sum_h x_h \\
s.t. \quad &\sum_{p \in P_g} y_{gp} = 1, \quad \textit{for all } g \in G, \\
&y_{gp} \leq x_h, \quad \textit{and} \\
&y_{gp} \leq x_{\bar{h}}, \quad \textit{for all } g \in G \textit{ and } p = \{h, \bar{h}\} \in P_g \\
&x_h \in \{0,1\} \quad \textit{for all } h \in \hat{H} \\
&y_{gp} \in \{0,1\} \quad \textit{for all } g \in G \textit{ and } p \in P_g.
\end{aligned}
$$

The first constraint of formulation 1 ensures that all genotypes are phased by some phasing, while the second and third constraints ensure that if a genotype is phased by some haplotype pair, then those haplotypes are paid for in the objective function.

In practice, this formulation can be prohibitively large. To solve this integer program Gusfield shows that for particular problem instances many of the variables and constraints can be eliminated without changing the nature of the problem. This brings the size of the integer program down to manageable size for many instances.

A more explicit formulation of the phasing problem models the resolution of the ambiguous genotype sites explicitly.

Formulation 2 *Given genotypes $G = \{g_1, \ldots, g_n\}$, we denote the value at site i of genotype $g \in G$ by $g(i)$. Let A_g be the ambiguous sites of $g \in G$, so that for all $i \in A_g$, $g(i) = 2$. Further, let $\mathcal{H}(g)$ and $\bar{\mathcal{H}}(g)$ represent the first and second phasing haplotype of g, respectively, so that $g = \mathcal{H}(g) + \bar{\mathcal{H}}(g)$. Similarly, we denote the genotype g that haplotype h phases by $\mathcal{G}(h)$, so that $\mathcal{G}(\mathcal{H}(g)) = \mathcal{G}(\bar{\mathcal{H}}(g))$. The phasing haplotypes of G are indexed so that $h_{2i-1} = \mathcal{H}(g_i)$ and $h_{2i} = \bar{\mathcal{H}}(g_i)$.*

In this formulation, we use an indicator variable x_{ga} for each ambiguous site $a \in A_g$ and genotype $g \in G$. $x_{ga} = 0$ indicates that the haplotypes $\mathcal{H}(g)$ and $\bar{\mathcal{H}}(g)$ have a 0 and 1 at position a of g, respectively; while $x_{ga} = 1$ indicates that haplotypes $\mathcal{H}(g)$ and $\bar{\mathcal{H}}(g)$ have a 1 and 0 at position a of g, respectively. The indicator variable $c_{jj'}$ is 1 if haplotypes h_j and $h_{j'}$ are different, and 0 if they are the same. Finally, we use the indicator variable l_j to indicate whether or not haplotype h_j is the last use of the haplotype represented by h_j.

$$
\begin{aligned}
&\min \sum_{j=1}^{2n} l_j \\
s.t.\ &L(h_j, i) - L(h_{j'}, i) \leq c_{jj'} \quad \textit{for all } j, j' = 1, \ldots, 2n, j < j', i = 1, \ldots, m \\
&L(h_{j'}, i) - L(h_j, i) \leq c_{jj'} \quad \textit{for all } j, j' = 1, \ldots, 2n, j < j', i = 1, \ldots, m \\
&L(h, i) = 0 \quad \textit{if } g(i) = 0, g = \mathcal{G}(h) \\
&L(h, i) = 1 \quad \textit{if } g(i) = 1, g = \mathcal{G}(h) \\
&L(h, i) = x_{gi} \quad \textit{if } g(i) = 2, g = \mathcal{G}(h), h = \mathcal{H}(g)
\end{aligned}
$$

$$
\begin{aligned}
L(h,i) = (1 - x_{gi}) \quad & \textit{if } g(i) = 2, g = \mathcal{G}(h), h = \bar{\mathcal{H}}(g) \\
1 - \sum_{j<j'} (1 - c_{jj'}) \leq l_j \quad & \textit{for all } j = 1, \ldots, 2n, \\
x_{ga} = 0 \quad & \textit{for all } g \in G, a = \min A_g \\
x_{ga} \in \{0,1\} \quad & \textit{for all } g \in G, a \neq \min A_g \\
c_{jj'} \in \{0,1\} \quad & \textit{for all } j, j' = 1, \ldots, 2n, j < j', \\
l_j \in \{0,1\} \quad & \textit{for all } j = 1, \ldots, 2n,
\end{aligned}
$$

The first and second constraints of formulation 2 enforce the definition of the $c_{jj'}$ in terms of the unambiguous sites of each genotype and the settings of the x_{ga} variables, while the last constraint enforces the definition of the l_j ensuring that l_j is 1 if all later haplotypes are different.

As with formulation 1, formulation 2 contains many unnecessary variables and constraints, depending on the particular problem instance that is being solved. This formulation isn't particularly suitable for solving with traditional integer programming solvers, since even when particular c variables are known to be 1, the relevant x variables can take many possible values.

Problem 8 (Implicit Phasing Integer Program). Reformulate formulation 2 to eliminate the x variables by instead deriving a sufficient set of constraints on the c variables from the set of genotypes G.

An intriguing open problem is to determine whether there are practical instances when this problem can be solved efficiently (for example if the perfect phylogeny condition holds, see section 2.5).

Problem 9 (Restricted Parsimony). Find a restriction on the input to the haplotype phasing problem that most real world instances satisfy, for which the most parsimonious haplotype assignment can be found in polynomial time.

Diversity is another commonly used parsimony objective in population genetics. Haplotype diversity is defined as the probability that two haplotypes drawn uniformly at random from the population are not the same.

Problem 10 (Haplotype Diversity Minimization). Devise an algorithm for the haplotype phasing under the objective of minimizing haplotype diversity.

We note that the integer programming formulation 2 above naturally solves the diversity objective under a suitable cost function. Graph theoretically, this problem can be posed as constructing a graph with a node for every haplotype in the observed population (two nodes for each observed genotype), an edge between every pair of haplotypes that are not equal and then minimizing the number of edges in the graph.

We observe that Clark's rule is not effective for parsimony.

Lemma 1. *Clark's rule does not yield an effective approximation algorithm for parsimony.*

Let n_d be the number of distinct genotypes in the G. The trivial algorithm of arbitrarily phasing each distinct genotype will return a phasing with at most $2n_d$ haplotypes. $\Omega(\sqrt{n_d})$ is a lower bound on the number of haplotypes as each genotype is made of at most two distinct haplotypes. A worst case approximation guarantee is thus $O(\sqrt{n_d})$, we will give such an example.

$$\left\{\begin{array}{l}(1111)+(1111)\\(1111)+(0011)\\(1111)+(1001)\\(1111)+(1100)\\(1111)+(1010)\\(1111)+(0101)\\(1111)+(0110)\end{array}\right\} \overset{\mathcal{P}_C}{\Longleftarrow} \left\{\begin{array}{l}(1111)\\(2211)\\(1221)\\(1122)\\(1212)\\(2121)\\(2112)\end{array}\right\} \overset{\mathcal{P}_P}{\Longrightarrow} \left\{\begin{array}{l}(1111)+(1111)\\(0111)+(1011)\\(1011)+(1101)\\(1101)+(1110)\\(1011)+(1110)\\(0111)+(1101)\\(0111)+(1110)\end{array}\right\}$$

Fig. 3. Set of 7 genotypes with 7 haplotype Clark's rule resolution $\mathcal{P}_C$, and 4 haplotype parsimony resolution $\mathcal{P}_P$.

Let m be the number of SNPs and let G be comprised of genotype that has all ones and all $\binom{m}{2}$ possible genotypes that have exactly two 2s and all other SNPs as 1s. Clark's inference rule will initially infer the haplotype of all ones and then infer the $\binom{m}{2}$ haplotypes that have all but 2 SNPs as 1s. The resolution with the minimum number of haplotypes however has the m haplotypes with all but 1 SNP as 1. An example when $m = 4$ is given in Figure 3. □

The Hamming distance between a pair of haplotypes is, under the infinite sites model, the number of mutations that occurred in the evolutionary history between the pair of haplotypes. If we consider an evolutionary history to be a tree whose nodes are the unknown haplotype sequences of the observed genotype sequences, then a likelihood function which approximates it [43] in terms of Hamming distance is given by:

$$L(\boldsymbol{h}) \propto \sum_{T} \prod_{e \in \text{Edges}(T)} f(D(e)) \tag{5}$$

where T ranges over all trees on the $2n$ nodes with unknown haplotypes $h_i \in H$, $1 \leq i \leq 2n$, e ranges over all $2n-1$ edges in T, $D(e)$ is the Hamming distance between the h_i and h_j which are joined by e, and f is a monotonic function. One reasonable choice might by $f(x) = e^{-\beta x}$ where β plays the role of the mutation rate, or one might take f from equation 4.

This sum over all trees of products of edge weights can be evaluated in polynomial time (using Kirchoff's matrix-tree theorem [5,32]). Methods for sampling from this and related distributions can be found in [4,8].

If we take $f(x) = e^{-\beta x}$, then we can interpret equation 5 as a partition function from statistical mechanics,

$$Z(\boldsymbol{h};\beta) = \sum_T e^{-\beta E(T,\boldsymbol{h})}$$

where $E(T,\boldsymbol{h})$ is the sum of the Hamming distances on all the edges in T.

Problem 11 (Partition Function Maximization). Devise an algorithm which maximizes

$$Z(\boldsymbol{h};\beta) = \sum_T e^{-\beta E(T,\boldsymbol{h})} \tag{6}$$

over all $\boldsymbol{h}$ explaining $\boldsymbol{g}$.

This problem has two asymptotic regimes.

The first is the *low temperature* regime $\beta \to \infty$, where, one can approximate the summation with maximization,

$$\begin{aligned} Z(\boldsymbol{h};\beta \sim \infty) &\sim \max_T e^{-\beta E(T,\boldsymbol{h})} \\ &= \exp\{-\beta \min_T E(T,\boldsymbol{h})\} \end{aligned}$$

and approximate the partition function with the minimum weight tree.

Problem 12 (Tree Minimization). Devise an algorithm which finds

$$\min_{T,\boldsymbol{h}} E(T,\boldsymbol{h}) \tag{7}$$

over all $\boldsymbol{h}$ explaining $\boldsymbol{g}$ and all trees T.

The second is the *high temperature* regime $\beta \sim 0$

$$Z(\boldsymbol{h};\beta) \sim \sum_T (1 - \beta E(T,\boldsymbol{h})) = (2n)^{2n-2}(1 - \frac{1}{2n} \sum_{h_1,h_2 \in \boldsymbol{h}} D(h_1,h_2))$$

where $D(h_1,h_2)$ is the Hamming distance between h_1 and h_2. In this extreme, the approximate problem is the minimization of the sum of all pairwise Hamming distances.

Problem 13 (Sum of Pairs Hamming Distance Minimization). Devise an algorithm which finds

$$\min_{\boldsymbol{h}} \sum_{h_1,h_2 \in \boldsymbol{h}} D(h_1,h_2) \tag{8}$$

over all $\boldsymbol{h}$ explaining $\boldsymbol{g}$ and all trees T.

Figure 4 gives an example where the sum of pairs Hamming distance minimization does not yield the same phasing as parsimony.

At the time of this writing, we are not familiar with any progress on these problems.

$$\left\{\begin{array}{l}(11111111)+(11111100)\\(11111111)+(11111001)\\(11111111)+(11110011)\\(11111111)+(11001111)\\(11111111)+(10011111)\\(11111111)+(00111111)\end{array}\right\} \underset{\mathcal{P}_P}{\Longleftarrow} \left\{\begin{array}{l}(11111122)\\(11111221)\\(11112211)\\(11221111)\\(12211111)\\(22111111)\end{array}\right\} \underset{\mathcal{P}_H}{\Longrightarrow} \left\{\begin{array}{l}(11111101)+(11111110)\\(11111011)+(11111101)\\(11110111)+(11111011)\\(11011111)+(11101111)\\(10111111)+(11011111)\\(01111111)+(10111111)\end{array}\right\}$$

Fig. 4. Set of 6 genotypes with 7 haplotype parsimony phasing $\mathcal{P}_P$, and 8 haplotype minimum sum of paired Hamming distances phasing $\mathcal{P}_H$.

2.5 Perfect Phylogeny

The concept of a *perfect phylogeny* [11,46,3] has also been used to formulate constraints on haplotype phasings. A (binary) perfect phylogeny is defined as follows: Let S be a set of n sequences (haplotypes) each drawn from Σ^m, where the alphabet $\Sigma = \{0, 1\}$. We say that S admits a *perfect phylogeny* if there exists a tree T with n leaves that has the following properties: (1) Each leaf of T is uniquely labeled with a sequence from S, (2) Every internal node v in T is labeled with a sequence from Σ^m, and (3) For each sequence position i (where $1 \leq i \leq m$) and for each $a \in \Sigma$, the set of nodes whose sequence labels each have the symbol a at position i, forms a subtree of T. The tree T is said to be a perfect phylogeny for S.

Gusfield [21] introduced a haplotype phasing problem that was motivated by studies on the haplotype structure of the human genome that reveal the genome to be *blocky* in nature ([9,25,47,17]), i.e., these studies show that human genomic DNA can be partitioned into long blocks where genetic recombination has been rare, leading to strikingly fewer distinct haplotypes in the population than previously expected. This *no-recombination in long blocks* observation together with the standard population genetic assumption of infinite sites, motivates a model of haplotype evolution where the haplotypes in a population are assumed to evolve along a coalescent, which as a rooted tree is a *perfect phylogeny*. Informally, this means that each SNP changed from a 0 to a 1 at most once in this rooted tree (here we are assuming that 0 is the ancestral state for a SNP). This motivates the following algorithmic problem called Perfect Phylogeny Haplotyping problem (PPH) - given n genotypes of length m each, does there exist a set S of at most $2n$ haplotypes such that each genotype is explained by a pair of haplotypes from S, and such that S admits a perfect phylogeny?

In [21], it was shown that the PPH problem can be solved in polynomial time by reducing it to a graph realization problem. The algorithm runs in $O(nm\alpha(nm))$, where α is the inverse Ackerman function, and hence this time bound is almost linear in the input size nm. The algorithm also builds a linear-space data structure that represents all the solutions, so that each solution can be generated in linear time. Although the reduction described in [21] is simple and the total running time is nearly optimal, the algorithm taken as a whole is very

difficult to implement, primarily due to the complexity of the graph realization component.

Following the work in [21], additional algorithms [2,14] have been proposed to solve the PPH problem that are simpler, easy to program and yet still efficient. These algorithms also produce linear-space data structures to represent all solutions for the given instance. Though they use quite different approaches, the algorithms in [2] and [14] take $O(nm^2)$ time. In [2], a non-trivial upper bound on the number of PPH solutions is also proved, showing that the number is vastly smaller than the number of haplotype solutions when the perfect phylogeny requirement is not imposed; furthermore, a biologically appealing representation is proposed that aids in visualizing the set of all solutions. In [14], an approach is also provided to deal with parent-child genotype data.

There are several interesting questions posed as a result of the works of [21,2,14]. We list three of them here.

Problem 14 (Optimal PPH). Can the PPH problem be solved in $O(nm)$? If so, is a practical algorithm possible?

Problem 15 (PPH with missing data). Devise solutions to deal with missing data and errors in the input.

The above problem is important as real data, very often, contains both missing data and errors. There are several directions that one could pursue here. For example, as in [14], one could study the complexity of the problem of removing a minimum number of genotypes so that the phasing of the remaining genotypes admits a perfect phylogeny. Alternatively, one could ask the question, can each missing value be set to one of $0, 1$, or 2 so that the resulting instance has a perfect phylogeny? Halperin and Karp [12] study this problem in a framework where the data are assumed to be generated by probabilistic models. Their results include a quadratic-time algorithm for inferring a perfect phylogeny from genotype data (with missing values) with high probability, under certain distributional assumptions.

The above problem is important as real data, very often, contains both missing data and errors. There are several directions that one could pursue here. For example, one could ask the question, can each missing value be set to one of $0, 1$, or 2 so that the resulting instance has a perfect phylogeny? Alternatively, as in [14], one could study the complexity of the problem of removing a minimum number of genotypes so that the phasing of the remaining genotypes admits a perfect phylogeny.

Problem 16 (PPH with recombination). What is the complexity of the problem when small deviations from the *no-recombination* model are allowed?

For instance, allowing for a small number of recombination events, can we still phase the genotypes efficiently in this framework? Allowing recombination events means that the solution is no longer a tree but a network (i.e. a graph with cycles) [51].

Recently, several approaches have been presented that explicitly model recombination [18,13,31].

Further work on haplotype phasing under the perfect phylogeny assumption has been Damaschke [10], by making the assumption that each haplotype has frequency at least $\frac{1}{m}$ a probabilistic algorithm is given that determines a phasing in time $O(mn\text{polylog}(m))$.

3 Haplotype Assembly

The need to infer haplotypes directly from genotypes is based on the assumption that biotechnologies for haplotype determination are unlikely to be available in the short term. This may not be the case. Various approaches to single molecule sequencing have been described recently [40,41,6] and some of these may mature to the point that phasing based solely on genotype analysis becomes unnecessary.

An increase in the current read length (~ 500) in a sequencing reaction to a few thousand basepairs, make it possible to phase large regions of a chromosome. Assuming that a SNP occurs every 1000 basepairs, many fragments will contain multiple SNPs. Consider a sequence assembly containing fragments f_1, f_2, f_3 from a single individual. If f_1 and f_2 differ in a SNP, they must come from different chromosomes. Likewise if f_2, and f_3 also differ in (some other) SNP, they come from different chromosomes. However, for a diploid organism, this must imply that f_1 and f_3 come from the same chromosome, and we have therefore *phased* the individual in this region (see Figure 5).

If reads have a length L and the distance between SNPs is exponentially distributed with a density ρ, then, with high coverage, the probability of being able to resolve the phase of two adjacent SNPs is $(1 - e^{-\rho L})$, which is close to 0.99 even for $\rho L = 5$. Thus, an order of magnitude increase in the current average read length could result in a haplotype resolution by haplotype assembly of groups on the order of 100 SNPs. Even current technology can produce reads of over 1000 basepairs, by creating a gapped read, where only the ends of the fragment are actually read, leaving a large gap in the middle.

Formally, define a SNP matrix M with rows representing fragments, and columns representing SNPs. Thus $M[f, s] \in \{0, 1, -\}$ is the value of SNP s in fragments f. Gapped reads are modeled as single fragments with gaps $(-)$ in SNPs that in the gap. Two fragments f and g conflict if there exists SNP s such that $M[f, s] = 0$, and $M[g, s] = 1$ or vice-versa. Based on this, a SNP matrix M can be used to define a fragment conflict graph $G_{\mathcal{F}}$. Each fragment is a node in $G_{\mathcal{F}}$. Two nodes are connected by an edge if they have different values at a SNP. It is easy to see that $G_{\mathcal{F}}$ is bipartite in the absence of errors. In the presence of errors, we can formulate combinatorial problems that involve deleting a minimum number of nodes (poor quality fragments), or edges (bad SNP calls), so that the resulting graph is bipartite (can be phased trivially).

Fig. 5. An illustration of the construction of long-range haplotypes from assembly data. A) the fragment assembly of a diploid organism. B) the identification of SNPs. C) the partitioning of the fragments into two consistent groups, introducing a long-range phasing.

In [45,34], the following is shown:

1. The minimum fragment removal and minimum SNP removal problems are tractable if the underlying matrix M has the consecutive ones property, i.e., there is no gap within a fragment.
2. They are NP-hard in the case of gaps, even when limited to at most one gap per fragment. The problems are shown to be tractable under a fixed parameter. That parameter being the total length of gaps in a fragment.

The algorithms are thus not tractable for dealing with the case of fragments with gaps, and it is an interesting open problem to design heuristics/approximations that give good results in practice. Some branch and bound heuristics were reported to work very well on real and simulated assembly data in [38]. Li *et al.* [30] give a statistical reformulation of this problem.

A fairly immediate extension to this problem, is the problem of simultaneous assembly multiple haplotypes. This will occur when studying multiploidal organisms or when simultaneously assembling related organisms, or a pooled set of individuals. For practical consideration it may be easier to sequence multiple related organisms simultaneously, for example to assemble different strains of a bacteria simultaneously.

Problem 17 (Multiploidal Haplotype Assembly). Devise an algorithm for assembling multiple haplotypes simultaneously.

4 Discussion

While the subject of haplotype phasing, or frequency inference may be of interest on purely statistical and mathematical grounds, the desired end result generally is an understanding of the implications of genetic variation in individual pathology, development, etc. As such, these variances are one part of a larger set of interactions which include individual environment, history, and chance.

Although the media often carries stories of "genes" (actually alleles) being found for some popular diseases, biologists agree that such stories are rather the exception than the rule when it comes to disease causation. It is suspected to be more the case that an individual's genetic variations interact with each other as well as other factors in excruciatingly complex and sensitive ways.

The future open problems of SNP analysis are those regarding the interactions of genetic variation with external factors. Substantial progress in multifactorial testing, or a tailoring of current multifactorial testing to this setting, is required if we are to see an impact of haplotype analysis on human health care.

5 Acknowledgements

We would like to thank Mark Adams, Sam Broder, Michelle Cargill, Andy Clark, Francis Kalush, Giuseppe Lancia, Kit Lau, Itsik Pe'er, Russell Schwartz, Roded Sharan, Hagit Shatkay, Francisco de la Vega and Mike Waterman for many valuable discussions on this subject.

References

1. G.R. Abecasis, R. Martin, and S. Lewitzky. Estimation of haplotype frequencies from diploid data. *American Journal of Human Genetics*, 69(4 Suppl. 1):114, 2001.
2. V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as a perfect phylogeny. A direct approach. *Journal of Computational Biology*, 10(3):323–340, 2003.
3. H. Bodlaender, M. Fellows, and T. Warnow. Two strikes against perfect phylogeny. In *Proceedings of the 19th International Colloquium on Automata, Languages, and Programming (ICALP)*, Lecture Notes in Computer Science, pages 273–283. Springer Verlag, 1992.
4. A. Broder. Generating random spanning trees. In *Proceedings of the IEEE 30th Annual Symposium on Foundations of Computer Science*, pages 442–447, 1989.
5. S. Chaiken. A combinatorial proof of the all-minors matrix tree theorem. *SIAM Journal on Algebraic and Discrete Methods*, 3:319–329, 1982.
6. E. Y. Chen. Methods and products for analyzing polymers. U.S. Patent 6,355,420.
7. A. G. Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7(2):111–122, 1990.
8. H. Cohn, R. Pemantle, and J. Propp. Generating a random sink-free orientation in quadratic time. *Electronic Journal of Combinatorics*, 9(1), 2002.
9. M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson, and E. S. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29:229–232, 2001.

10. Peter Damaschke. Fast perfect phylogeny haplotype inference. In *14th Symposium on Fundamentals of Computation Theory FCT'2003, Malm, LNCS 2751*, pages 183–194, 2003.
11. W. H. E. Day and D. Sankoff. Computational complexity of inferring phylogenies by compatibility. *Systematic Zoology*, 35(2):224–229, 1986.
12. Richard M. Karp Eran Halperin. Perfect phylogeny and haplotype assignement. In *Proceedings of the Eigth Annual International Conference on Computational Molecular Biology (RECOMB)*, 2004. To appear.
13. Lauri Eronen, Floris Geerts, and Hannu Toivonen. A markov chain approach to reconstruction of long haplotypes. In *Pacific Symposium on Biocomputing (PSB 2004)*, 2004. To appear.
14. E. Eskin, E. Halperin, and R. M. Karp. Large scale reconstruction of haplotypes from genotype data. In *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 104–113, 2003.
15. L. Excoffier and M. Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):921–927, 1995.
16. D. Fallin and N.J. Schork. Accuracy of haplotype frequency estimation for biallelic loci, via the expectation-maximization algorithm for unphased diploid genotype data. *American Journal of Human Genetics*, 67(4):947–59, 2000.
17. L. Frisse, R. Hudson, A. Bartoszewicz, J. Wall, T. Donfalk, and A. Di Rienzo. Gene conversion and different population histories may explain the contrast between polymorphism and linkage disequilibrium levels. *American Journal of Human Genetics*, 69:831–843, 2001.
18. Gideon Greenspan and Dan Geiger. Model-based inference of haplotype block variation. In *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 131–137, 2003.
19. D. Gusfield. A practical algorithm for optimal inference of haplotypes from diploid populations. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 183–189, 2000.
20. D. Gusfield. Inference of haplotypes from samples of diploid populations: Complexity and algorithms. *Journal of Computational Biology*, 8(3):305–324, 2001.
21. D. Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions (Extended abstract). In *Proceedings of the Sixth Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 166–175, 2002.
22. D. Gusfield. Haplotyping by pure parsimony. In *Proceedings of the 2003 Combinatorial Pattern Matching Conference*, pages 144–155, 2003.
23. D. L. Hartl and A. G. Clark. *Principles of Population Genetics*. Sinauer Associates, 1997.
24. M. E. Hawley and K. K. Kidd. HAPLO: A program using the EM algorithm to estimate the frequencies of multi-site haplotypes. *Journal of Heredity*, 86:409–411, 1995.
25. L. Helmuth. Genome research: Map of the human genome 3.0. *Science*, 293(5530):583–585, 2001.
26. E. Hubbell. Finding a maximum likelihood solution to haplotype phases is difficult. Personal communication.
27. E. Hubbell. Finding a parsimony solution to haplotype phase is NP-hard. Personal communication.
28. R. R. Hudson. Gene genealogies and the coalescent process. In D. Futuyma and J. Antonovics, editors, *Oxford surveys in evolutionary biology*, volume 7, pages 1–44. Oxford University Press, 1990.

29. A. J. Jeffreys, L. Kauppi, and R. Neumann. Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex. *Nature Genetics*, 29(2):217–222, 2001.
30. L. Kim, J.H. Kim, and M.S. Waterman. Haplotype reconstruction from SNP alignment. In *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 207–216, 2003.
31. Gad Kimmel and Ron Shamir. Maximum likelihood resolution of multi-block genotypes. In *Proceedings of the Eigth Annual International Conference on Computational Molecular Biology (RECOMB)*, 2004. To appear.
32. G. Kirchhoff. Über die auflösung der gleichungen, auf welche man bei der untersuchung der linearen verteilung galvanischer ströme geführt wird. *Annalen für der Physik und der Chemie*, 72:497–508, 1847.
33. A. Kong, D. F. Gudbjartsson, J. Sainz, G. M. Jonsdottir, S. A. Gudjonsson, B. Richardsson, S. Sigurdardottir, J. Barnard, B. Hallbeck, G. Masson, A. Shlien, S. T. Palsson, M. L. Frigge, T. E. Thorgeirsson, J. R. Gulcher, and K. Stefansson. A high-resolution recombination map of the human genome. *Nature Genetics*, 31(3):241–247, 2002.
34. G. Lancia, V. Bafna, S. Istrail, R. Lippert, and R. Schwartz. SNPs problems, complexity and algorithms. In *Proceedings of the Ninth Annual European Symposium on Algorithms (ESA)*, pages 182–193, 2001.
35. J. Li and T. Jiang. Efficient rule based haplotyping algorithms for pedigree data. In *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 197–206, 2003.
36. J. Li and T. Jiang. An exact solution for finding minimum recombinant haplotype configurations on pedigrees with missing data by integer linear programming. In *Proceedings of the Eigth Annual International Conference on Computational Molecular Biology (RECOMB)*, 2004. To Appear.
37. S. Lin, D. J. Cutler, M. E. Zwick, and A. Chakravarti. Haplotype inference in random population samples. *American Journal of Human Genetics*, 71:1129–1137, 2002.
38. R. Lippert, R. Schwartz, G. Lancia, and S. Istrail. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Briefings in Bioinformatics*, 3(1):23–31, 2002.
39. J. C. Long, R. C. Williams, and M. Urbanek. An E-M algorithm and testing strategy for multiple-locus haplotypes. *American Journal of Human Genetics*, 56(2):799–810, 1995.
40. R. Mitra, V. Butty, J. Shendure, B. R. Williams, D. E. Housman, and G. M. Church. Digital genotyping and haplotyping with polymerase colonies. *Proceedings of the National Academy of Sciences*, 100(10):5926–31, 2003.
41. R. Mitra and G. M. Church. In situ localized amplification and contact replication of many individual DNA molecules. *Nucleic Acids Research*, 27(e34):1–6, 1999.
42. T. Niu, Z. S. Qin, X. Xu, and J. S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *American Journal of Human Genetics*, 70:157–169, 2002.
43. M. Nordborg. *Handbook of Statistical Genetics*, chapter Coalescent Theory. John Wiley & Sons, Ltd, 2001.
44. N. Patil, A. J. Berno, D. A. Hinds, W. A. Barrett, J. M. Doshi, C. R. Hacker, C. R. Kautzer, D. H. Lee, C. Marjoribanks, D. P. McDonough, B. T. N. Nguyen, M. C. Norris, J. B. Sheehan, N. Shen, D. Stern, R. P. Stokowski, D. J. Thomas, M. O. Trulson, K. R. Vyas, K. A. Frazer, S. P. A. Fodor, and D. R. Cox. Blocks of limited

haplotype diversity revealed by high resolution scanning of human chromosome 21. *Science*, 294:1719–1723, 2001.

45. R. Rizzi, V. Bafna, S. Istrail, and G. Lancia. Practical algorithms and fixed-parameter tractability for the single individual SNP haplotyping problem. In *Proceedings of the Second International Workshop on Algorithms in Bioinformatics (WABI)*, pages 29–43, 2002.
46. M. A. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9:91–116, 1992.
47. J. C. Stephens, J. A. Schneider, D. A. Tanguay, J. Choi, T. Acharya, S. E. Stanley, R. Jiang, C. J. Messer, A. Chew, J.-H. Han, J. Duan, J. L. Carr, M. S. Lee, B. Koshy, A. M. Kumar, G. Zhang, W. R. Newell, A. Windemuth, C. Xu, T. S. Kalbfleisch, S. L. Shaner, K. Arnold, V. Schulz, C. M. Drysdale, K. Nandabalan, R. S. Judson, G. Ruano, and G. F. Vovis. Haplotype variation and linkage disequilibrium in 313 human genes. *Science*, 293(5529):489–493, 2001.
48. M. Stephens and P. Donnelly. Inference in molecular population genetics. *Journal of the Royal Statistical Society, Series B*, 62(4):605–635, 2000.
49. M. Stephens and P. Donnelly. A comparison of bayesian methods for haplotype reconstruction from population genotype data. *American Journal of Human Genetics*, 73:1162–1169, 2003.
50. M. Stephens, N. J. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics*, 68:978–989, 2001.
51. L. Wang, K. Zhang, and L. Zhang. Perfect phylogenetic networks with recombination. *Journal of Computational Biology*, 8(1):69–78, 2001.
52. P. Zhang, H. Sheng, A. Morabia, and T. C. Gilliam. Optimal step length EM algorithm (OSLEM) for the estimation of haplotype frequency and its application in lipoprotein lipase genotyping. *BMC Bioinformatics*, 4(3), 2003.