Haplotypes versus Genotypes on Pedigrees

Bonnie Kirkpatrick[1]

Electrical Engineering and Computer Sciences, University of California Berkeley and
International Computer Science Institute, email: bbkirk@eecs.berkeley.edu

Abstract. Genome sequencing will soon produce haplotype data for individuals. For pedigrees of related individuals, sequencing appears to be an attractive alternative to genotyping. However, methods for pedigree analysis with haplotype data have not yet been developed, and the computational complexity of such problems has been an open question. Furthermore, it is not clear in which scenarios haplotype data would provide better estimates than genotype data for quantities such as recombination rates.

To answer these questions, a reduction is given from genotype problem instances to haplotype problem instances, and it is shown that solving the haplotype problem yields the solution to the genotype problem, up to constant factors or coefficients. The pedigree analysis problems we will consider are the likelihood, maximum probability haplotype, and minimum recombination haplotype problems.

Two algorithms are introduced: an exponential-time hidden Markov model (HMM) for haplotype data where some individuals are untyped, and a linear-time algorithm for pedigrees having haplotype data for all individuals. Recombination estimates from the general haplotype HMM algorithm are compared to recombination estimates produced by a genotype HMM. Having haplotype data on all individuals produces better estimates. However, having several untyped individuals can drastically reduce the utility of haplotype data.

Pedigree analysis, both linkage and association studies, has a long history of important contributions to genetics, including disease-gene finding and some of the first genetic maps for humans. Recent contributions include fine-scale recombination maps in humans [4], regions linked to Schizophrenia that might be missed by genome-wide association studies [11], and insights into the relationship between cystic fibrosis and fertility [13]. Algorithms for pedigree problems are of great interest to the computer science community, in part because of connections to machine learning algorithms, optimization methods, and combinatorics [7, 16, 12, 10, 15].

Single-molecule sequencing is an attractive alternative to genotyping and would yield haplotypes for individuals in a pedigree [6]. Such technologies are being developed and may become commercial within five to ten years. Sequencing methods would apparently yield more information from the same set of sampled individuals, which is critical due to the limited availability of individuals for sampling in multi-generational pedigrees (i.e. individuals usually must be living at the time of sampling). There is substantial evidence that haplotypes can be more useful than genotypes for both population and family based studies when using methods such as association studies [1, 3] and pedigree analysis [2, 8]. While it is intuitive that haplotypes provide more information than genotypes, there are instances with family data in which there are few enough typed individuals that there is little practical difference between haplotype and genotype data. Additionally, in order to exploit the information contained in haplotype data, we need to understand the instances where diploid inheritance is computationally tractable given haplotype data.

Pedigree analysis with genotype data is well studied in terms of complexity [12, 10] and algorithms [5, 9, 14]. Less is known about haplotype data on pedigrees. This paper shows that, given haplotype data on a pedigree, finding both minimum recombination and maximum probability haplotypes is as tractable as computing the same quantities for pedigrees with genotype data (i.e., these problems are NP- and #P-hard, respectively). To obtain a reduction that applies equally well to several types of pedigree calculations, we will consider a modular polynomial-time mapping from the genotype problem to the haplotype problem. The reduction preserves the solutions to the analyses, meaning that the solution to the haplotype problem is the solution to the genotype problem after adjusting by constant factors or coefficients.

Since the reduction uses a biologically unlikely recombination scenario, we will investigate the accuracy and information of realistic examples with haplotypes and genotype data on the same pedigree. Pedigree data was simulated having a known number of recombinations. The recombination distributions were computed at a particular locus of interest and compared to the ground-truth. Since both the haplotypes and genotypes of a specific person contain the same alleles, the differences between the haplotype and genotype recombination distributions were determined by the extra information in the haplotype data. As expected, the haplotype data reliably yields greater accuracy when all the pedigree individuals are typed. However, as fewer pedigree individuals are typed, there is less practical difference between the utility of haplotype versus genotype data. The number of untyped generations that separate typed individuals influences whether haplotype data are actually more accurate than genotype data. For instance with two half-siblings, having two untyped parents results in estimates from genotype data that are nearly as accurate as the estimates computed from haplotype data.

Finally, there is an important instance where haplotype data is more computationally tractable than genotype data. When all individuals in the pedigree are typed, although unlikely from a practical perspective of obtaining genetic samples, the computational problem decomposes into conditionally independent sub-problems, and has a linear-time algorithm. This can be contrasted with the known hardness of the genotype problem even when all individuals are genotyped. The existence of this linear-time algorithm for haplotype data could facilitate useful approaches that combine population genetic and pedigree methods. For instance, if the haplotypes of the founders are drawn from a coalescent and the pedigree individuals are all haplotyped, the probability of a combined model could easily be computed for certain coalescent models.

1 Introduction to Pedigree Analysis

A *pedigree* is a directed acyclic graph where the set of nodes, I, are individuals, and directed edges indicate genetic inheritance between parent and child. A diploid pedigree (i.e. for humans) necessarily has either zero or two incoming edges for each person. The set, F, of individuals without incoming edges are referred to as pedigree *founders*. An individual, i, with two parents is a *non-founder*, and we will refer to their two parents as $m(i)$ and $p(i)$.

As is commonly done to accommodate inheritance of genetic information, we will extend this model to include a representation of the alleles of each individual and of the inheritance origin of each allele. More formally, we represent a single chromosome as an ordered sequence of variables, x_j, where each variable takes on an *allele* value in $\{1, ..., k_j\}$. Each variable represents a *polymorphic site*, j, in the genome, where there are k_j possible sequence variants. Since diploid individuals have two copies of each chromosome, one copy inherited from each parent, we will use a superscript m and p to indicate the maternal and paternal chromosomes respectively. For a particular individual i, the information on both copies of a particular chromosome at site j is represented as $x^m_{i,j}$ and $x^p_{i,j}$.

Furthermore, we assume that inheritance in the pedigree proceeds with recombination and without mutation (i.e. Mendelian inheritance at each site). This imposes consistency rules on parents and children: the allele $x^m_{i,j}$ must appear in the mother $m(i)$'s genome as either the grand-maternal or grand-paternal allele, $x^m_{m(i),j}$ or $x^p_{m(i),j}$, and similarly for the paternal allele and the father $p(i)$'s genome.

Let x be a vector containing all the haplotypes x^m_i, x^p_i for all individuals $i \in I$, then we are interested in the probability

$$\mathbb{P}[x] = \prod_{f \in F} \mathbb{P}[x^p_f]\mathbb{P}[x^m_f] \prod_{i \in I \setminus F} \mathbb{P}[x^p_i | x^p_{p(i)}, x^m_{p(i)}]\mathbb{P}[x^m_i | x^p_{m(i)}, x^m_{m(i)}], \tag{1}$$

where the superscript m and p indicate maternal and paternal alleles, while the functions $m(i)$ and $p(i)$ indicate parents of i. The first product is over the independent founder individuals whose haplotypes are drawn from a uniform prior distribution, while the second product, over the non-founders, contains the probabilities for the children to inherit their haplotypes from their parents. The unobserved vector x is not immediately derived from observed haplotype data, since vector x contains haplotype alleles labeled with their parental origins for all the individuals. To compute this quantity, we need notation to represent the parental origins of each allele where differing origins for neighboring haplotype alleles will indicate recombination events.

For each non-founder, let us indicate the source of each maternal allele using the binary variable $s^m_{i,j} \in \{m, p\}$, where the value m indicates that $x^m_{i,j}$ allele has grand-maternal origin and p indicates grand-paternal origin. Similarly, we define

$s^p_{i,j}$ for the origin of i's paternal allele. For a particular site, these indicators for all the individuals, s_j, is commonly referred to as the identity-by-descent (IBD) inheritance path. A recombination is observed at consecutive sites as a change in the binary value of a source vector, for instance, $s^m_{i,j} = p$ and $s^m_{i,j+1} = m$. To compute the inheritance portion of Equation 1, we will sum over the inheritance options $\mathbb{P}[x] = \sum_s \mathbb{P}[x|s]\mathbb{P}[s]$ where $\mathbb{P}[s] = 1/2^{2|I \setminus F|}$

We can observe two kinds of data for pedigree individuals whose genetic material is available. The first, and most common, is genotype data, a tuple of alleles $(g^0_{i,j}, g^1_{i,j})$ that must appear in the variables $x^m_{i,j}$ and $x^p_{i,j}$ for each site j. Since these alleles are unlabeled for origin, we do not know which allele was inherited from which parent. The second type of data is haplotypes, where we observe two sequences of alleles h^0_i and h^1_i and each sequence represents alleles that were inherited together from the same parent. However, we do not know which sequence is maternal and which is paternal. For either type of data define a function $C_{i,j}$ for locus j which indicates compatibility of the assigned haplotype alleles with the data and requires inheritance consistency between generations. Specifically, for genotype data $C_{i,j} = 1$ if $x^m_{i,j} = x^{s^m_{i,j}}_{m(i),j}$, $x^p_{i,j} = x^{s^p_{i,j}}_{f(i),j}$, and $\{x^m_{i,j}, x^p_{i,j}\} = \{g^0_{i,j}, g^1_{i,j}\}$. Under haplotype data, the $C_{i,j} = 1$ when the first two equalities, above, hold and $\{x^m_{i,j}, x^p_{i,j}\} = \{h^0_{i,j}, h^1_{i,j}\}$, which are the haplotype alleles at locus j.

Now, we write Equation 1 as a function of the per-site recombination probability $\theta \leq 0.5$. For particular values of all the haplotype alleles $x^m_{i,j}$ and $x^p_{i,j}$, the haplotype probability conditional on the inheritance options and the observed data through $C_{i,j}$ is

$$\mathbb{P}[x|s] = \prod_{f \in F} \prod_{j=1}^{l} C_{f,j}\mathbb{P}[x^p_{f,j}]\mathbb{P}[x^m_{f,j}] \prod_{i \in I \setminus F} C_{i,1} \prod_{j=2}^{l} C_{i,j} \cdot \theta^{(R^m_{i,j} + R^p_{i,j})} \cdot (1-\theta)^{(2 - R^m_{i,j} - R^p_{i,j})}$$

where $R^m_{i,j} = \mathbb{I}[s^m_{i,j-1} \neq s^m_{i,j}]$ and $R^p_{i,j} = \mathbb{I}[s^p_{i,j-1} \neq s^p_{i,j}]$.

1.1 Pedigree Problem Formulations

Given a pedigree and some observed genotype or haplotype data, there are three problem formulations that we might be interested in. The first is to compute the probability of some observed data, while the last two problems find values for the unobserved haplotypes of individuals in the pedigree.

Likelihood. Find the probability of the observed data by summing over all the possible unobserved haplotypes, i.e. $\sum_x \sum_s \mathbb{P}[x|s]\mathbb{P}[s]$.

Maximum Probability. Find the values of $x^m_{i,j}$ and $x^p_{i,j}$ that maximize the probability of the data, i.e. $\max_x \sum_s \mathbb{P}[x|s]\mathbb{P}[s]$.

Minimum Recombination. Find the values of $x^m_{i,j}$ and $x^p_{i,j}$ that minimize the number of required recombinations, i.e.
$\min_{x,s} \sum_i \sum_{j \geq 2} \mathbb{I}[s^p_{i,j-1} \neq s^p_{i,j}] + \mathbb{I}[s^m_{i,j-1} \neq s^m_{i,j}]$.

The likelihood is commonly used for estimating site-specific recombination rates, relationship testing, computing p-values for association tests, and performing linkage analysis. Haplotype and/or IBD inferences, obtained by maximizing the probability or minimizing the recombinations, are useful for non-parametric association tests, tests on haplotypes, and tests where there is disease information for unobserved genomes.

2 Hardness Results

With genotype data, the likelihood and minimum recombination problems are NP-hard, while the maximum probability problem is #P-hard. Piccolboni and Gusfield [12] proved the hardness of the likelihood and maximum probability computations by relying on a single locus sub-pedigree with half-siblings. Although their paper discussed a more elaborate setting involving a phenotype, their proof, however, applies to this setting. Li and Jiang proved the minimum recombination problem to be hard by using a two-locus sub-pedigree with half-siblings [10]. In all these proofs, half-siblings were pivotal to establishing reductions from well known NP and #P problems.

In this paper, we introduce a simple and powerful reduction that converts any genotype problem on a pedigree of n individuals into a haplotype problem on a pedigree of at most $6n$ individuals. This reduction is simple, because it merely introduces four full-siblings and an extra parent for each genotyped individual. We do not need complicated structures involving inbreeding or half-siblings. The reduction works equally well for all three problem formulations.

Mapping. Given a pedigree with genotype data, for any of the three pedigree problems, we define a polynomial mapping to a corresponding haplotype problem with exactly $5|G|$ individuals haplotyped. First we create the pedigree graph for the new haplotype instance, and later we construct the required haplotype observations from the genotype data.

Let $G \subset I$ represent the set of genotyped individuals in a pedigree having individuals I and edges E. We will create a haplotype instance of the problem, with individuals $H \cup I$ and edges $R \cup E$. To obtain the set H, we add five individuals, i_0, i_1, i_2, i_3, i_4, to H for every individual $i \in G$. The set of new relationship edges, R, will connect individuals in sets H and G. Specifically, the edges stipulate that i and i_0 are the parents of full-siblings i_1, i_2, i_3, and i_4 by including the edges: $i_0 \rightarrow i_1$, $i_0 \rightarrow i_2$, $i_0 \rightarrow i_3$, $i_0 \rightarrow i_4$, $i \rightarrow i_1$, $i \rightarrow i_2$, $i \rightarrow i_3$, and $i \rightarrow i_4$. We will refer to these five individuals, i_0, i_1, i_2, i_3, and i_4, and their relationships with i as the *proxy family* for individual i. For example in Figure 1, the 6-individual genotype pedigree in becomes a 21-individual haplotype pedigree. This produces a pedigree graph with exactly $5|G|+|I|$ individuals and $8|G|+|E|$ edges.

To obtain the new haplotype data from the genotype data, we type only individuals in $|H|$ such that the corresponding genotyped individual in G is required, by the rules of inheritance, to have the observed genotypes. Without loss of generality, assume that the genotype alleles are sorted such that $g^0_{i,j} < g^1_{i,j}$. Now we can easily constrain the parental genotype for individual $i \in G$ by giving the spouse, i_0, homozygous haplotypes of all ones while giving child i_1 the haplotypes $\{\mathbf{1}, g^0_i\}$, child i_2 haplotypes $\{\mathbf{1}, g^1_i\}$. This guarantees the correct genotype, but does not ensure that the haplotypes of that genotype have the same probability or number of recombinations.

Since there is an arbitrary assorting of genotype alleles at neighboring loci into the parent haplotypes x^p_i and x^m_i, we will use the remaining two children to represent possible re-assortments of the genotyped parent's T_i heterozygous loci, indexed by t_j where $1 \leq j \leq T_i$. In addition to the haplotype $\mathbf{1}$, child i_3, will have haplotype consisting of $h_{i_3,t_j} := g^{1-j \mod 2}_{i,t_j}$ while child i_4 has the genotyped parent's complementary alleles $h_{i_4,t_j} := g^{j \mod 2}_{i,t_j}$. This results in child i_3 and i_4 alternating in having the smaller allele at every other heterozygous locus.

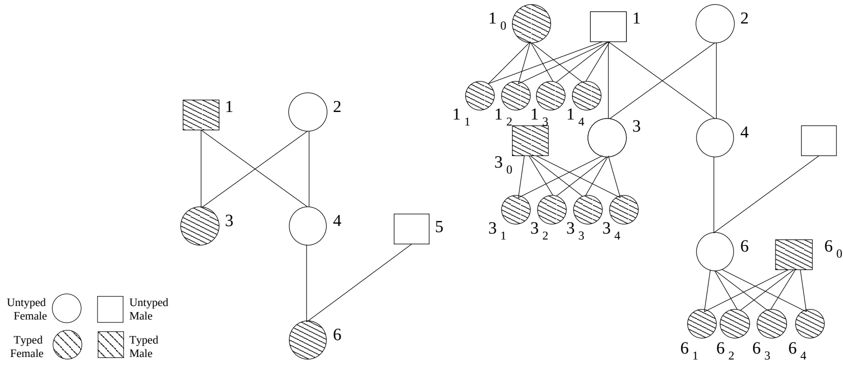

Fig. 1. Genotype and Haplotype Pedigrees. Values are averages computed from 500 simulations. (Left) Genotyped individuals are shaded, and all the individuals are labeled. Individuals 1, 2, and 5 are the founders, and individual 6 is the grandchild of 1 and 2. (Right) Haplotyped individuals are shaded, and individuals have the same labels. For each of the genotyped individuals, i, from the previous figure, the mapping adds a nuclear family containing five new individuals labeled i_0, i_1, i_2, i_3, i_4.

This reduction preserves the solutions to the three problems up to constant factors or constant coefficients. Specifically, the solution to the haplotype version of the problem is the solution to the genotype version with the values of the functions being related by constant factors or coefficients, depending on whether the function is a recombination count or a probability.

Lemma 1 *Let r_g be the minimum number of recombinations in the genotype problem instance. The mapping yields a haplotype problem instance having $r_h = r_g + \sum_{i \in G} 2(T_i - 1)$ for the minimum number of recombinations, where T_i is the number of heterozygous sites in genotype i.*

To prove this result, we exploit the alternating pattern of alleles assigned to the four children. This pattern forces there to be two recombinations, among the four children, between consecutive heterozygous loci.

After applying the mapping, the haplotype probability turns out to have a coefficient that is independent of the haplotype assignment to the non-founding parent of the proxy family. This coefficient can be computed in linear time from the genotype data using a Markov chain. The Markov chain has 16 states and has a transition step between each pair of neighboring loci. This small Markov model can be thought of in the sum-product algorithm as an elimination of the typed individuals in the proxy family; alternatively, it is also equivalent to peeling-off the typed proxy individuals in the Elston-Stewart algorithm [5]. Once we have this coefficient, independent of the haplotype assignment, it is clear that the likelihood and maximum probability haplotype problems also have haplotype solutions related proportionally to the genotype solution.

Lemma 2 *The mapping yields a haplotype problem instance having haplotype probabilities proportional to the haplotype probabilities of the genotype instance. Specifically, for all x,*

$$\mathbb{P}_h[x] = \left(\mathbb{P}_g\left[\{x_i | i \in I\}\right]\right) \prod_{i \in G} p_t(i) \prod_j \mathbb{P}[x^p_{i_0,j} = 1]\mathbb{P}[x^m_{i_0,j} = 1]$$

where the proxy family transmission probability is a function of genotype g_i, the recombination rate $\theta \leq 0.5$, and of the transition matrices P, Q_{0110}, and Q_{1001},

$$p_t(i) = \left(\frac{1}{16}\right) \mathbf{1} \cdot P^{h_0} \prod_{j=0}^{T_i} \left(O_j Q_{0110} + (1 - O_j) Q_{1001}\right) \cdot P^{h_j} \cdot \mathbf{1}^T$$

and O_j indicates whether index j is odd, h_0 is the number of homozygous loci that begin proxy parent's genotype, and h_j is the number of consecutive homozygous loci after the j'th heterozygous locus where there are T_i heterozygous loci for proxy parent i. The transition probabilities are given by $P_{ij} = \theta^{H(i,j)}(1-\theta)^{4-H(i,j)}$ where $H(i,j)$ is the Hamming distance between inheritance states i and j. Let Q_{0110} be a transition matrix having non-zero recombination probabilities only in column 0110 (i.e. $Q_{0110,i,j} = P_{ij}$ when $j = 0110$). Similarly, let Q_{1001} be a transition matrix with non-zero recombination probabilities only in column 1001.

Although this reduction establishes the hardness of these haplotype pedigree problems, it does so by constructing children whose haplotypes require many recombinations and would be extremely unlikely to occur naturally. Accordingly, we suspect that realistic instances of these haplotyping problems may provide more information about the locations of recombinations than genotype instances.

3 Algorithms and Accuracy of Estimates

One indication that the haplotype problem might be practically more tractable is the amount of information in the haplotype data relative to the genotype data. To understand this, we can consider a pedigree with a fixed set of sampled individuals. Assume that there are two input data sets available, either the haplotype or the genotype data, for all the sampled individuals. Note that the alleles observed will be identical in both the haplotype and genotype data, so we are interested in the distribution that these data impose on the inheritance probabilities. By comparing the accuracy of the recombination estimates under these two data sets, we can get an idea for how useful the respective probability distributions are.

Let R_j be a random variable representing the number of recombinations in the whole pedigree that occur between loci $j-1$ and j. Similar to our notation before, $R_j = \sum_{i \in I} R^p_{i,j} + R^m_{i,j}$. We want to compute the distribution of R_j under both the genotype and haplotype inheritance probability distributions. These two inheritance distributions are different precisely because there are haplotypes and inheritance paths that are consistent with the genotype constraints but disallowed by the haplotype constraints.

These distributions are obtained by constructing a hidden Markov model for the linkage dependencies along the genome. At each locus, the HMM considers the constraints given by either the haplotype or genotype data (i.e. the haplotype data HMM is a variation on the Lander-Green algorithm [9]). We first use the forward-backward algorithm to compute the marginal inheritance probabilities for each locus using a hidden Markov model. Once we have the marginal probabilities, we can easily obtain the distribution for R_j.

3.1 General Haplotype and Genotype HMMs

The likelihood can be modeled using a hidden Markov model along the genome with inheritance paths as hidden states. An *inheritance path* is a graph with nodes being the alleles of individuals and directed edges between alleles that are inherited from parent to child. The transition probabilities are functions of θ and the number of recombinations between a given pair of inheritance graphs.

Given the data, we compute the marginal inheritance path probabilities at each site by using the forward-backward algorithm for HMMs. Sobel and Lange described a method for enumerating the inheritance paths compatible with the allele data observed at each locus [14]. There are at most $k = 2^{2|I \setminus F|}$ inheritance paths when $I \setminus F$ is the set of non-founder individuals, and both the forward and backward recursions do an $O(k^2)$ calculation at each site.

To compute the analogous probability for haplotype data, we use a similar HMM. For haplotypes, the hidden states must consider the *haplotype orientations*, which specify the parental origins of all the observed haplotypes. Notice that these orientations are not equivalent to inheritance paths, since they only specify inheritance edges between haplotyped individuals and their parents. For each of the $2^{2|H|}$ haplotype orientations, where H is the set of haplotyped individuals, we enumerate the inheritance paths compatible with the haplotype alleles, their orientations, and the pedigree relationships. Alternatively, each of the inheritance paths enumerated for the genotype algorithm induces a particular orientation on the haplotypes heterozygous for that locus (i.e. parental origin of the entire haplotype). Thus, the hidden states for the haplotype HMM are the cross-product of the orientations and the inheritance paths.

The haplotype HMM has transition probabilities that are nearly identical to the genotype HMM with the exception that transitions between inheritance paths with different haplotype orientations have probability zero. Recombinations are only allowed when they do not occur between typed haplotypes.

The forward-backward algorithm is also used on the haplotype HMM. However, there are $2^{2(|I|+|H|-|F|)}$ hidden states, yielding a slightly slower calculation. Fortunately, the haplotype recursions can be run simultaneous with the genotype recursions, meaning that the inheritance paths need only be enumerated once.

3.2 Haplotype Likelihoods in Linear Time

There is one obvious instance of the haplotyping problems where there are polynomial-time algorithms. Even though it is impractical to assume that we can sample genetic material from deceased individuals in a multi-generational pedigree, for a moment, let us consider the case where all the individuals in the pedigree are haplotyped.

The Elston-Stewart algorithm [5] for genotype data has a direct analogue for haplotype data. This algorithm calculates the likelihood via the belief propagation algorithm by eliminating individuals recursively from the bottom up. Each individual is "peeled off", after their descendants have been peeled off, by using a forward-backward algorithm on the HMM for the mother-father-child trio.

The haplotype version of this algorithm is linear when all the individuals are haplotyped, since each elimination step is conditionally independent of all the others. Given the parents' haplotypes, regardless of which was inherited from which grand-parent, the probability of the child's haplotype is independent of all other trios. Therefore, we can take a product over the likelihoods for all the trios, and compute each trio likelihood using a 4-state HMM. Then for k non-founding individuals, and l loci, this algorithm has $O(kl)$ running time.

This same intuition carries through to the minimum recombination problem, and each trio can be considered independent of the others. This contrasts with the genotype minimum recombination problem which is known to be hard, even when all the individuals are genotyped [10].

3.3 Results

To simulate realistic pedigree data, SNPs were selected from HapMap that span 100mb on both sides of a loosely-linked pair of sites. There are 40 SNPs total, with 20 tightly linked SNPs on each side of a strong recombination breakpoint having $\theta = 0.25$. The haplotypes for these SNPs were selected randomly from HapMap. Pedigree haplotype and genotype data were simulated for each child by uniformly selecting one of the parental alleles for the first locus, and subsequent loci were selected on the same parental haplotype with probability θ_j for each locus j. Inheritance was simulated for 500 simulation replicates.

The simulation yielded completely typed pedigrees. For each pedigree, we removed the genotype and haplotype information for increasing numbers of untyped individuals. For each instance of a specific number of untyped individuals, two values were computed on the estimated number of recombinations between the central pair of loci: the haplotype and genotype accuracies. Accuracy was computed as a function of the l_1 distance between the deterministic number of recombinations and the calculated distribution. Specifically, accuracy was $2 - \sum_{i \geq 0} |x_i - a_i|$, where x_i was the estimated probability for i recombinations and a_i was the deterministic indicator of whether there were i recombinations in the data simulated on the pedigree.

In all the instances we observed a trend where the best accuracy was obtained with haplotype data where everyone in the pedigree was haplotyped. For example, a five-individual pedigree with two half-siblings is shown in Figure 2, left panel. With the three founders untyped, the haplotype data yielded similar accuracy as the genotype data. Consider a three-generation pedigree having two parents, their two children, an in-law, and a grandchild for a total of six individuals, three of them founders. This pedigree has a similar trend in accuracy as the number of untyped founders increases, Figure 2, right panel. As the number of untyped individuals increases, the accuracies of genotype and haplotype estimates appear to converge.

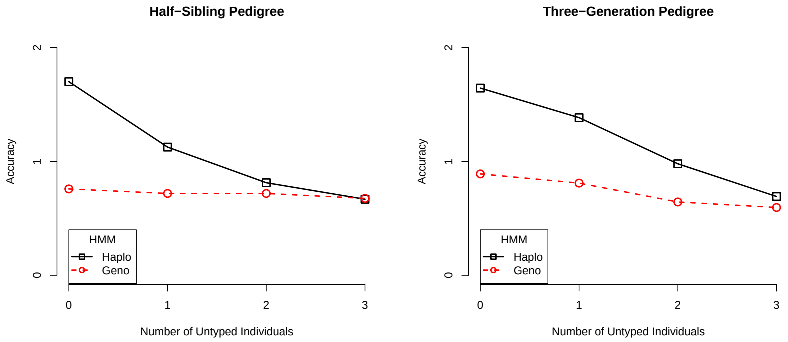

Fig. 2. Predicting Recombinations. The left panel is the average accuracy for predictions from a pedigree with two half-siblings and three parents. The right panel shows results from a six-individual, three-generation pedigree. In both cases, 500 simulation replicates were performed, and the average accuracy of estimates from the haplotype data is superior to those from genotype data. However, as the number of untyped founders increases, in both cases, the accuracy of estimates from haplotype data drop relative to the accuracy from genotype data. The accuracies of genotype and haplotype estimates appear to converge.

4 Discussion

Sequencing technologies would seem to solve the phasing problem by yielding haplotype data. However, if we wish to consider diploid inheritance with recombination, the phasing problem remains, even when we are given chromosome-length haplotype data. This is demonstrated by reduction of the phasing problem for genotypes to the phased version of the same problem for three common pedigree problems. This theoretical result is due largely to the unavailability of genetic material for deceased individuals.

Three pedigree calculations were discussed: likelihood, maximum probability, and minimum recombination. Each of these calculations on haplotype data have the same computational complexity as the same computation on genotype data. In the worst case, it takes only a single generation to remove the correlation between sites in the haplotype. This worst case provided the reduction that proves the the complexity results for the haplotype computations, and it worked equally well for all three pedigree computations.

The worst-case is not biologically realistic, since it requires roughly $2(m-1)$ recombinations for m sites in 4 meioses. This is very unlikely to occur under typical models for inheritance. To investigate more likely scenarios, sequences were simulated in a region of the genome surrounding a recombination breakpoint. From haplotype and genotype data, we estimated the distribution of the number of recombinations at the breakpoint and compared the estimates to the ground-truth for accuracy.

When typing everyone in the pedigree, the estimates from haplotype data were very accurate, because the haplotype data provides enough constraints to determine where the recombinations must have occurred. With decreasing numbers of typed individuals, the accuracy of haplotype-based estimates dropped until it seemed to converge to the genotype accuracy due to a lack of constraints. From the structure of the calculations, we observed that with fewer typed individuals there were more haplotype orientations to consider, and the haplotype calculation more closely resembled the genotype calculation. However, the haplotype calculation had more constraints and lost accuracy at a slower rate.

Several interesting open problems remain. First, approximation algorithms might be a useful approach for haplotypes on pedigrees. The existence of a linear-time algorithm when all individuals are haplotyped may suggest that the general haplotype problem instance could be amenable to approximation algorithms. Second, these proofs apply when there is no missing data in a genotyped individual (i.e. a proxy parent). The proof requires knowing whether the proxy parent is heterozygous or homozygous at each locus, and this is unknown when there is missing data. Third, there is an interesting case of mixed haplotypes and genotypes. For this case to be interesting, the ends of haplotypes must occur at different locations in different individuals in the pedigree. Otherwise, the haplotypes that start and end at the same positions in all individuals can easily be converted into multi-allelic genotypes, with an allele for each haplotype. The mixed haplotype-genotype problem is not amenable to the proof techniques used here. However, the haplotype HMM in Section 3.1 can easily be revised to handle the mixed case. This is important because the data produced by single polymer sequencing is more likely to resemble the mixed case than either the haplotype or the genotype cases.

Acknowledgments. I want to thank Richard Karp for reviewing a draft of the manuscript and the National Science Foundation for support through the Graduate Research Fellowship.

References

1. J.C. Barrett, S. Hansoul, D.L. Nicolae, J.H. Cho, R.H. Duerr, J.D. Rioux, S.R. Brant, M.S. Silverberg, K.D. Taylor, M.M. Barmada, and et al. Genome-wide association defines more than 30 distinct susceptibility loci for crohn's disease. *Nature Genetics*, 40:955–962, 2008.
2. J.T. Burdick, W. Chen, G.R. Abecasis, and V.G. Cheung. In silico method for inferring genotyeps in pedigrees. *Nature Genetics*, 38:1002–1004, 2006.
3. W.-M. Chen and G.R. Abecasis. Family-based association tests for genomewide association scans. *American Journal of Human Genetics*, 81:913 – 926, 2007.
4. G. Coop, X. Wen, C. Ober, J.K. Pritchard, and M. Przeworski. High-Resolution Mapping of Crossovers Reveals Extensive Variation in Fine-Scale Recombination Patterns Among Humans. *Science*, 319(5868):1395–1398, 2008.
5. R.C. Elston and J. Stewart. A general model for the analysis of pedigree data. *Human Heredity*, 21:523–542, 1971.
6. J. Eid, et al. Real-Time DNA Sequencing from Single Polymerase Molecules. *Science*, 323(5910):133–138, 2009.
7. D. Geiger, C. Meek, and Y. Wexler. Speeding up HMM algorithms for genetic linkage analysis via chain reductions of the state space. *Bioinformatics*, 25(12):i196, 2009.
8. B. Kirkpatrick, E. Halperin, and R.M. Karp. Haplotype inference in complex pedigrees. *Journal of Computational Biology*, page in press, 2010.
9. E.S. Lander and P. Green. Construction of multilocus genetic linkage maps in humans. *Proceedings of the National Academy of Science*, 84(5):2363–2367, 1987.
10. Jing Li and Tao Jiang. An exact solution for finding minimum recombinant haplotype configurations on pedigrees with missing data by integer linear programming. In *Proceedings of the 7th Annual International Conference on Research in Computational Molecular Biology*, pages 101–110, 2003.
11. Ng MY, Levinson DF, and et al. Meta-analysis of 32 genome-wide linkage studies of schizophrenia. *Mol Psychiatry*, 14:774–85, 2009.
12. A. Piccolboni and D. Gusfield. On the complexity of fundamental computational problems in pedigree analysis. *Journal of Computational Biology*, 10(5):763–773, 2003.
13. IG Romero and C Ober. CFTR mutations and reproductive outcomes in a population isolate. *Human Genet*, 122:583–588, 2008.
14. E. Sobel and K. Lange. Descent graphs in pedigree analysis: Applications to haplotyping, location scores, and marker-sharing statistics. *American Journal of Human Genetics*, 58(6):1323–1337, 1996.
15. Bhalchandra D. Thatte. Combinatorics of pedigrees, 2006.
16. J. Xiao, L. Liu, L. Xia, and T. Jiang. Efficient Algorithms for Reconstructing Zero-Recombinant Haplotypes on a Pedigree Based on Fast Elimination of Redundant Linear Equations. *SIAM Journal on Computing*, 38:2198, 2009.