BIOINFORMATICS ORIGINAL PAPER

Vol. 23 no. 22 2007, pages 3048–3055
doi:10.1093/bioinformatics/btm435

Genetics and population analysis

HAPLOPOOL: improving haplotype frequency estimation through DNA pools and phylogenetic modeling

Bonnie Kirkpatrick[1,*], Carlos Santos Armendariz[2], Richard M. Karp[1,3] and Eran Halperin[3]

[1]Department of Electrical Engineering and Computer Sciences, UC Berkeley, CA, [2]Computer Science Department, Universidad Rey Juan Carlos, Madrid, Spain and [3]International Computer Science Institute, Berkeley, CA, USA

Received on June 7, 2007; revised on August 3, 2007; accepted on August 16, 2007

Advance Access publication September 25, 2007

Associate Editor: Keith Crandall

ABSTRACT

Motivation: The search for genetic variants that are linked to complex diseases such as cancer, Parkinson's, or Alzheimer's disease, may lead to better treatments. Since haplotypes can serve as proxies for hidden variants, one method of finding the linked variants is to look for case-control associations between the haplotypes and disease. Finding these associations requires a high-quality estimation of the haplotype frequencies in the population. To this end, we present, HAPLOPOOL, a method of estimating haplotype frequencies from blocks of consecutive SNPs.

Results: HAPLOPOOL leverages the efficiency of DNA pools and estimates the population haplotype frequencies from pools of disjoint sets, each containing two or three unrelated individuals. We study the trade-off between pooling efficiency and accuracy of haplotype frequency estimates. For a fixed genotyping budget, HAPLOPOOL performs favorably on pools of two individuals as compared with a state-of-the-art non-pooled phasing method, PHASE. Of independent interest, HAPLOPOOL can be used to phase non-pooled genotype data with an accuracy approaching that of PHASE.

We compared our algorithm to three programs that estimate haplotype frequencies from pooled data. HAPLOPOOL is an order of magnitude more efficient (at least six times faster), and considerably more accurate than previous methods. In contrast to previous methods, HAPLOPOOL performs well with missing data, genotyping errors and long haplotype blocks (of between 5 and 25 SNPs).

Availability: The HAPLOPOOL software is available at: http://haplopool.icsi.berkeley.edu/haplopool/

Contact: bbkirk@eecs.berkeley.edu

Supplementary information: Supplementary data are available at *Bioinformatics online*.

1 INTRODUCTION

Human genetic variation is key to understanding complex heritable diseases. Much of this variation can be characterized by single nucleotide polymorphisms (SNPs), which are evidence of mutations that occurred in the past and were passed on through heredity. Recent progress in technology for high-throughput SNP genotyping provides an opportunity to understand the genetic basis of complex disease through whole-genome association studies. In these studies, hundreds of thousands of SNPs are genotyped for the cases and the controls, and discrepancies between the haplotype distributions indicate an association between a genetic region and the disease. Advances in high-throughput genotyping are a mixed blessing. As more SNPs are genotyped for a given cost, more individuals are needed to overcome the increase in genome-wide false positives. For example, when correcting for multiple hypotheses with the Bonferroni correction, an additional SNP decreases the genome-wide critical value. To find the etiology of complex disease, association studies need thousands of individuals (Carlson *et al.*, 2004). With today's genotyping costs, a well-powered whole genome association study may cost millions of dollars. Finding new ways to reduce the burden of genotyping is critical for large rigorous association studies.

One step in this direction is the use of haplotypes. SNPs in close physical proximity to each other are often correlated (in *Linkage Disequilibrium*), and the variation of the *haplotype* (sequence of alleles in consecutive SNP sites along a chromosomal region) is known to be of limited diversity. Consider a haplotype and a nearby SNP. The haplotype can give evidence for the presence of an allele at the SNP, even when the SNP has not been genotyped (de Bakker *et al.*, 2006). As a result, the identification and analysis of haplotypes (The International HapMap Consortium, 2003), is currently playing a key role in trait and disease associations studies (Morris and Kaplan, 2002).

Pooled DNA is another natural strategy for reducing genotyping costs. Some technologies are able to determine the SNP-allele frequencies with high precision in pooled samples, thereby replacing many individual genotype measurements with one consolidated analysis. These technologies extract the DNA from a pool of individuals using approximately equal amounts of DNA from each individual. This bulked DNA is genotyped and the frequency of an allele in each position is given (Sham *et al.*, 2002). Therefore, for pools of size l, the cost of genotyping is reduced by a factor of l.

DNA pooling is not without caveats. First, DNA pools lose the information of the individual genotypes, and hence they lose haplotype frequency information. Second, the measurement error in estimating allele frequencies for DNA pools can be quite high [with a variance between 0.02 and 0.04

*To whom correspondence should be addressed.

(Sham *et al.*, 2002)], and accurately determining allele frequencies for a large DNA pool is infeasible. For these reasons, large DNA pools are currently used in association studies as a screening procedure (Barcellos *et al.*, 1997). The cases and the controls are pooled separately. SNPs for which there is a large discrepancy between the pool allele frequencies in the cases and the controls are individually genotyped in a validation stage. Unfortunately, these large pools loose nearly all the haplotype frequency information. Furthermore, inaccuracies in allele frequency estimations from DNA pools result in a large number of false positives carried along to the validation stage.

For a middle ground, we suggest using disjoint pools of a small number of unrelated individuals (two or three per pool). We show that a careful analysis of small DNA pools reveals haplotype frequency information. We leverage on the error rate for DNA pools of two individuals that is comparable to the individual genotyping error rate (Norton *et al.*, 2002). Although Barratt *et al.* (2002) concluded that pools of 50 individuals were optimal for the trade-off between the pool size and the statistical power for case-control studies, pooling errors were inaccurately assumed to equally affect large and small pools. Let the error σ be the SD of allele frequencies across repeated calls for the same SNP. The ability of the clustering algorithms to correct the error depends on whether 2σ is larger than the difference in frequency between allowable frequency calls. For pools of two individuals, there are five possible allele frequency values (0, 0.25, 0.5, 0.75 and 1), with a frequency difference of 0.25 between neighboring frequency calls. Thus, an accuracy of $\sigma < 0.125$ will ensure a low rate of incorrect allele frequency calls (<1%). Several related works (Germer *et al.*, 2000; Le Hellard *et al.*, 2002) demonstrate that σ is within this threshold for pools of two individuals.

In addition, the use of small pools is advantageous since information about quantitative traits can be analyzed by grouping the individuals into pools of similar characteristics. For example, if the distribution of a trait is normal, individuals falling into the same SD can be grouped into pools. The idea is that individuals with similar quantitative values might have similar genetic factors that are correlated with the expression of the trait.

For a fixed genotyping budget, a question that arises is whether we gain information by using small DNA pools rather than traditional genotyping. By asking this question, two assumptions must hold (1) that the cost of genotyping a pool is roughly equivalent to the cost of genotyping an individual, and (2) that the cost of genotyping dominates the cost of sample collection and largely determines the cost of a study. The results here show that one can get more accuracy by estimating haplotype frequencies from DNA pools of two or three individuals. We introduce a method, HAPLOPOOL, which estimates haplotype frequencies from a set of DNA pools of l individuals each. These frequencies are estimated from a set of n DNA pools of l samples each (a total of nl individuals). We compare them to haplotype frequencies estimated by the state-of-the-art phasing method PHASE (Stephens *et al.*, 2001) on a set of n non-pooled genotypes. In order for PHASE to achieve the same level of accuracy as HAPLOPOOL, one must perform 45% more non-pooled genotype experiments. This may seem counterintuitive at first, as it is widely assumed that haplotype frequency information is lost when DNA pools are used. Note, we compare the haplotype frequency estimates to the haplotype frequencies in the entire population and not in the sample.

Using small DNA pools has been suggested in three previous works (Hoh *et al.*, 2003; Ito *et al.*, 2003; Yang *et al.*, 2003). In all cases, the expectation-maximization (EM) algorithm was used to infer haplotype frequencies from DNA pools of a small number of individuals. Our method is different in a number of points. First, our algorithm incorporates a perfect phylogeny model, which helps to increase the accuracy of frequency estimates. Second, we make use of an EM algorithm only for small subsets of the SNPs, thus getting partial solutions, which we combine into one global solution using linear regression. Third, our method is more computationally efficient and runs at least six times faster than previous methods. For previous methods on pools of size 2, running times on some genomic regions spanning more than 10 SNPs may take hours. The running time of HAPLOPOOL on these instances is a matter of seconds. Fourth, our method is more accurate and capable of dealing well with missing data and genotyping errors.

Yang *et al.* (2003) have results that are most directly comparable to the results in this article. Although they also noted the efficiency of pooling, their method was limited in the number of SNPs allowed in a data set and was only compared to itself. Evidently, it is hard to interpret their results since the performance of their method on genotypes may be inferior to state-of-the-art methods such as PHASE. It has not been shown that their method, for pools of size 2, is more efficient than PHASE on genotypes. Furthermore, Yang *et al.* only show results for haplotypes spanning three SNPs, while our results extend to dozens of SNPs. Our results also hold in the presence of high error rates (up to 5%) in the allele frequencies determined from the pools.

2 METHODS

HAPLOPOOL is based on three different layers. First, we calculate a set of potential haplotypes in the population based on a perfect phylogeny model and augmented by a variant of the greedy algorithm for haplotype inference (Halperin and Karp, 2004). Then, we use these haplotypes to estimate the haplotype frequencies for many subsets of the SNPs in the region by applying an EM algorithm. Finally, we combine this information to infer the haplotype frequencies over the entire set of SNPs using linear regression. Since the EM algorithm for a small number of SNPs is efficient, the performance of our algorithm is not limited by the running time of the EM algorithm. The effectiveness of our method stems from three things: the accuracy of EM for a small number of SNPs, the ability to identify common haplotypes using the perfect phylogeny model and the greedy method, and the accurate deduction of global haplotype information from the estimates on subsets of SNPs.

In order to describe the method we need to introduce some notation. We assume that we are given a set of n pools, each consisting of l individuals (l will typically be 1, 2 or 3). For each pool, m SNPs are genotyped. The result of the genotyping is a set of n vectors, $a_1, \ldots, a_n$, where $a_n \in \{0, 1, \ldots, 2l\}^m$. The value a_{ij}, the jth coordinate of a_i, corresponds to the minor allele frequency of that SNP in pool i. We denote the minor allele frequency of the jth SNP as $p_j = 1/(2nl) \sum_i a_{\{ij\}}$. For simplicity of the exposition, we assume that the SNPs are

ordered by their minor allele frequencies, i.e. $p_1 \leq p_2 \leq p_m$. We denote a haplotype for the set of m SNPs by a binary string of length m. For a haplotype $h \in \{0,1\}^m$, the minor allele is present in position i if $h_i = 1$. There may be 2^m possible haplotypes, corresponding to all binary strings of length m.

Even though in theory 2^m haplotypes are possible, we see in practice that within a short genomic region there is a limited diversity. We leverage on the high correlations between SNPs in close proximity, which indicates a small number of haplotypes covering at least 80% of the population (Patil *et al.*, 2001). This assumption and its relationship to accuracy will be explored in the Results section using simulated data. Therefore, we search for a restricted set of plausible haplotypes, $\mathcal{H} = \{h_1, \ldots, h_d\}, (d \ll 2^m)$, which may explain our set of pools. Once such a set is found, we describe an algorithm that will estimate their frequencies $\mathcal{F} = \{f_1, \ldots, f_d\}$ in the population. Due to the limited diversity, in most cases we expect to find that d is quite small (no more than 20), and therefore our algorithms can run efficiently on these haplotypes.

2.1 The perfect phylogeny model

In order to guarantee highly accurate estimates of the haplotype frequencies in the region, it is crucial to find the correct set of haplotypes, or at least the most common haplotypes. We begin our search for the set of haplotypes $\mathcal{H}$, by first finding a set $\mathcal{H}_c$ on a sub-case of the coalescent model (Kingman, 1982), the perfect phylogeny model.

In the perfect phylogeny model, we assume that the ancestral history of the haplotypes is described by a rooted phylogenetic tree, in which each node corresponds to a haplotype, and each edge corresponds to a SNP. The root of the tree represents the ancestral haplotype of the population. Every edge in the tree corresponds to one mutation that occurred at one point in history, from a parent haplotype to its child. Thus, two haplotypes that are connected by an edge differ in exactly one position. There are many possible phylogenies, but the perfect phylogeny model adds the restriction that there are no recombinations, back mutations or recurrent mutations. In other words, there is exactly one edge in the tree that corresponds to each of the SNPs; this edge corresponds to the one time in history when a mutation occurred at this SNP (see Fig. 1). The perfect phylogeny model has been used for phasing (Halperin and Eskin, 2004), but the approach taken there cannot be extended to DNA pools. In addition, the discussion about deviations from perfect phylogeny was informal. Here, we present a formal discussion of how to fit a perfect phylogeny model to the data while allowing deviations from perfect phylogeny. We also introduce an algorithm applicable both to genotype data and to DNA pools.

For a perfect phylogeny tree T, we introduce the following notation. Since every node corresponds to a haplotype, we will denote the nodes of the tree by their corresponding haplotypes. For a node h, we denote by T_h the subtree rooted at h. We say that a SNP s is in T_h if the edge corresponding to s is in the subtree T_h, or if it is the parent edge of h. The root of the tree will be denoted by r, and we consider trees with roots consisting of only major alleles (0-alleles). In practice, this restriction does not affect the generality of the model, since any unrooted perfect phylogeny on binary characters can be rooted at the haplotype of major alleles to yield a rooted perfect phylogeny.

In order to find the set of haplotypes $\mathcal{H}_c$, we search for a perfect phylogeny tree T that provides the 'best' explanation for the data. Put differently, for every tree T, we compute the likelihood of the data being generated by that particular tree. A tree, together with its haplotypes and haplotype frequencies, fully specifies the likelihood of the data. The haplotypes, $\mathcal{H}_c$, of a tree are induced by the nodes of the tree and are said to be *compatible* with the perfect phylogeny tree that induced them. We can use the tree to estimate the frequencies of these haplotypes in the following way. For a haplotype h, let i_0 be the SNP corresponding to the parent edge of h, and let $i_1, \ldots, i_k$ be the SNP edges leading to the child haplotypes of h (see Fig. 1). Since the mutation represented by i_0 occurs only once in the tree, p_{i_0} is the total frequency of all the haplotypes in the subtree T_h. The same holds for each p_{i_j} and the subtrees rooted at the children of h. Therefore, we estimate the frequency of h as $f_h = p_{i_0} - \sum_{j=1}^{k} p_{i_j}$. For the root r, we assume that $p_{i_0} = 1$, and thus the sum of the frequencies of all haplotypes is exactly 1.

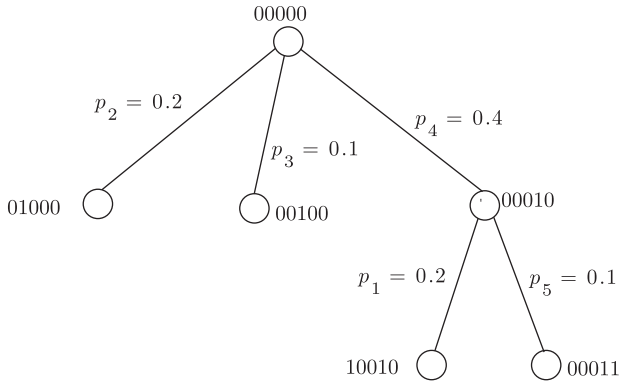

Fig. 1. Perfect phylogeny example. Each node of this perfect phylogeny is labeled with a haplotype. The edge representing the mutation of SNP j is labeled with the minor allele frequency p_j. We compute the haplotype frequencies from the allele frequencies. For example, the frequency of haplotype 00010 is computed as $p_4 - (p_1 + p_5)$. Once the haplotype frequencies are computed, the tree gives a model for a specific population of haplotypes.

The resulting set of frequencies $\mathcal{F} = \{f_1, \ldots, f_d\}$ for the compatible haplotypes is now used to compute the likelihood of the tree T. First, if one of the resulting haplotype frequencies is negative, we set the likelihood to be zero. Otherwise, the likelihood of the data given the tree allows the data to deviate from the perfect phylogeny and is defined as follows. Let a *configuration* of haplotypes for a pool be a vector $c = (c_1, \ldots, c_d)$, where c_j is the count for haplotype h_j in the pool such that $0 \leq c_j \leq 2l$, and $\sum_j c_j = 2l$. Again, we will denote the allele at SNP k in haplotype j as $h_{j,k} \in \{0,1\}$. The *mutation number* of the configuration c with respect to pool i is $mut(c,i) = \sum_{k=1}^{m} |a_{i,k} - \sum_j h_{j,k} c_j|$. Informally, $mut(c,i)$ is the number of recurrent mutations needed to explain the pool frequencies by the compatible haplotypes in the configuration. With these definitions, the likelihood function can be written as

$$L(a_1, \ldots, a_n \mid T) = \prod_{i=1}^{n} \left(\max_{\{\text{configuration } c\}} \epsilon^{mut(c,i)} \cdot \prod_{j=1}^{d'} \frac{f_j^{c_j}}{c_j!} \right), \qquad (1)$$

where ϵ is the probability for a mutation, which we assume is known. The mutation probability is the prior probability of observing a point-mutation in any of the perfect phylogeny haplotypes, and it provides a penalty for trees with haplotypes that deviate a great deal from the data. This probability includes both the probability of a point-mutation actually occurring in the population and the probability of there being a genotyping measurement error that leads to an incorrect allele count. In order to compute the likelihood of the tree T, we find the most likely configuration for each pool, i.e. the configuration that maximizes the expression $\epsilon^{mut(c,i)} \cdot \prod_{j=1}^{d'} \frac{f_j^{c_j}}{c_j!}$.

Since there is an exponential number of possible configurations, it is not feasible to try every possible configuration for every possible pool. Therefore, for each pool, we use a linear-time dynamic programming algorithm on the tree to find the most likely configuration, and thus we obtain the likelihood of a given tree (details in the Supplementary Material). We implicitly compare all possible trees while constructing

the max likelihood tree from top to bottom. Since we want rooted perfect phylogenies, we restrict our search to trees in which the allele frequency of a SNP is larger than the allele frequency of any SNP in its subtree. Recall that we assume that $p_1 \geq p_2 \geq , \ldots, \geq p_m$. Thus, we start with the tree S_0 consisting of the single vertex r. We then compute a sequence $S_1, S_2, \ldots, S_m$, where S_j is a set of perfect sub-phylogenies over the SNPs $1, \ldots, j$. S_{j+1} is obtained from S_j as follows: in all possible ways, we augment each sub-phylogeny in S_j by adding a new leaf and attaching it to the sub-phylogeny by an edge labeled $j+1$; we compute the likelihood of each resulting sub-phylogeny, and retain the 500 with the highest likelihood. Finally, the highest-scoring phylogeny in S_m is used to estimate the haplotype frequencies.

2.2 The EM algorithm

Once the most likely tree has been found, we use the frequencies $\mathcal{F} = \{f_1, \ldots, f_d\}$ as the estimate of the haplotype frequencies in the population. When the tree needs a small mutation number to explain all of the pools, we observe that the model gives very accurate estimates for the haplotype frequencies. However, when more recurrent mutations are needed, evidently the set of haplotypes in that region cannot be reasonably explained by the perfect phylogeny model. Then, we obtain a set of haplotypes $\mathcal{H}_g$ using a greedy approach (details in the Supplementary Material), which is based on the algorithm described in Halperin and Karp, (2004).

Once the set of haplotypes $\mathcal{H}$ is determined by the perfect phylogeny model and the greedy algorithm, we estimate the haplotype frequencies of subsets of SNPs using the EM algorithm. For simplicity of the presentation, we will describe the EM algorithm applied to the entire set of m SNPs. Once again, we use the vector $c = (c_1, \ldots, c_d)$ to represent a configuration for a pool. A configuration is said to be *valid* in pool i if and only if $\sum_j c_j = 2l$ and for every SNP $1 \leq k \leq m$, $a_{i,k} = \sum_j h_{j,k} c_j$. After making the assumption of Hardy–Weinberg equilibrium, the likelihood function can be written simply as

$$L(a_1, \ldots, a_n | f_1, \ldots, f_d) = \prod_{i=1}^{n} \sum_{\{c \text{ valid in pool } i\}} \frac{(2l)!}{c_1! c_2! \cdots c_d!} f_1^{c_1} f_2^{c_2} \cdots f_d^{c_d}$$

The EM algorithm finds a set of frequencies $\{f_1, \ldots, f_d\}$ that maximizes the likelihood. In each iteration we have an estimate $\hat{f}_1, \ldots, \hat{f}_d$. Then, the EM algorithm alternates between computing

$$\beta_j^i = \sum_{\{c \text{ valid in pool } i\}} c_j \frac{(2l)!}{c_1! c_2! \cdots c_d!} \hat{f}_1^{c_1} \hat{f}_2^{c_2} \cdots \hat{f}_d^{c_d} \quad (2)$$

and assigning

$$\hat{f}_{h_j} = \sum_{i=1}^{n} \frac{\beta_j^i}{\sum_{m=1}^{d} \beta_m^i} \quad (3)$$

until the algorithm converges.

2.3 Combining estimates on projected haplotypes

When considering pools of size l, the number of possible valid configurations is roughly d^{2l}, where d is the number of plausible haplotypes. Thus, for pools of more than two individuals, it is essential that d be relatively small. To mitigate this, we focus on the information contained in subsets of the available SNPs. For a subset of SNPs $P \subseteq \{1, \ldots, m\}$, we project the haplotypes in $\mathcal{H}$ onto the columns of P. This results in haplotypes on subsets of the SNPs, or *projected haplotypes*, $\mathcal{H}^P = \{h_1^P, \ldots, h_d^P\}$, where $h_i^P \in \{0,1\}^{|P|}$ is the binary string h_i projected to the set of columns P. Clearly, different haplotypes of $\mathcal{H}$ may have the same projection, and thus, $\mathcal{H}^P$ may contain fewer than d haplotypes. In a similar manner, we obtain projections of the pool allele count vectors $a_i^P \in \{0,1\}^{|P|}$ for every pool i. The projected haplotype set $\mathcal{H}^P$ together with the projected allele count vectors a_i^P are then used as an input for the EM algorithm. For each such subset of SNPs, the EM algorithm provides an estimation of the frequencies of the haplotypes of $\mathcal{H}^P$.

For each given subset P of the columns, the haplotype frequencies of $\mathcal{H}^P$ give a linear constraint on the haplotype frequencies of the haplotypes in $\mathcal{H}$. For each haplotype $h \in \mathcal{H}^P$, we know that $\sum_{i: h_i^P = h} f_i = f_h$, where f_i is the frequency of h_i in the population, and f_h is the frequency of h. For instance, if P contains only one SNP j, then the above constraints imply that the sum of all haplotype frequencies with 1 in SNP j should be the frequency of the minor allele 1 in that SNP.

We use the EM algorithm to compute projected haplotype frequencies of all single SNPs, all pairs of SNPs, a set of $\binom{m}{2}$ random triples of SNPs, and a set of $\binom{m}{2}$ random quadruples of SNPs. Each of these runs gives a set of linear constraints. We use this strategy, because in practice it gives the finest-grained estimate of projected haplotypes that reliably increases the accuracy of our algorithm. Using longer projected haplotypes hurts the running time without providing more accuracy. In addition, we use the linear constraint that the sum of all frequencies is exactly 1. Thus, we end up with a $\{0,1\}$ matrix C with d columns, and a vector b that corresponds to the projected haplotype frequencies. The resulting haplotype frequencies could then be computed by solving the set of equations $Cx = b$.

Unfortunately, due to errors in the frequency estimation for the different sets of SNPs, the different solutions are inconsistent, and therefore no solution exists to the equation $Cx = b$. We thus consider the least squares solution for this equation, i.e. we find $x \geq 0$ such that $\|Cx - b\|_2$ is minimized. A similar approach was taken by Barratt *et al.* (2002) and Pe'er and Beckmann (2003), where the equations corresponded to single SNPs, and the setting involved large DNA pools.

The above least squares procedure treats all constraints as if the components of the vector b are distributed according to a standard Gaussian. However, clearly, some constraints are more informative than others, and some are more error-prone than others. For instance, the constraints imposed by single SNP projections are much more accurate than the constraints imposed by pairs of SNPs. In general, we find that the accuracy of the EM algorithm declines when the number of SNPs gets larger. It is therefore useful to give more weight to the constraints that are based on haplotypes projected over smaller numbers of SNPs. We rewrite the objective function as $x^* = \text{argmin}_x \sum_i w_i \left(b_i - \sum_j C_{ij} x_j\right)^2$, where w_i represents the weight given for the ith constraint. Each w_i is the inverse of the variance expected of the residual, and it serves to standardize the error. If $w_i = 1$ for all i, then this is simply the least squares formulation describe above. For an efficient practical implementation, we found that we could fix the weights as $w_i = 3m$ for single SNP projections and $w_i = 1$ for projections of two or more SNPs.

2.4 The combined HAPLOPOOL algorithm

Summarizing the above algorithms, we ended up implementing three versions of the HAPLOPOOL algorithm, as described in Figure 2. The first version includes the perfect phylogeny alone, and both the haplotypes and the frequencies are derived from the tree, as described above. In the second version, we use the perfect phylogeny together with greedy to generate $\mathcal{H} = \mathcal{H}_c + \mathcal{H}_g$, as described above, and the EM together with the linear regression to compute the frequencies of these haplotypes. In the third version we use greedy alone to generate $\mathcal{H} = \mathcal{H}_g$, and we calculate the frequencies using the EM and the linear regression.

These three versions allow us to measure the effect of the perfect phylogeny model on the accuracy of the algorithm (those results are presented in the Supplementary Material). The first version, the perfect

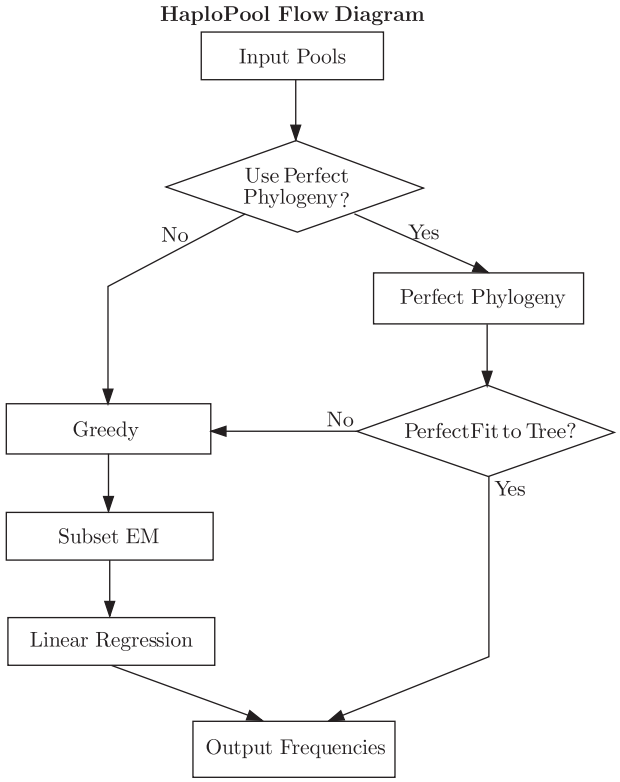

Fig. 2. HaploPool flow diagram. The four procedures located in the rectangular boxes are the core of the HaploPool implementation. There are three possible paths from the input to the output in this flow diagram. We compared all three in the Supplementary Material and found the best results when always using the perfect phylogeny and letting the mutation number determine whether the regression portion of the algorithm is executed. When the mutation number is small, the perfect phylogeny frequencies are very good, and the algorithm returns those.

phylogeny only, outperforms the other versions as long as there are no recurrent mutations needed to model the data ($\sum_i mut(c,i) < 1$). When the number of recurrent mutations gets larger, the combination of perfect phylogeny with greedy (second variant of the algorithm) gives the best results. Thus, the HaploPool algorithm chooses which algorithm to use for the final frequency estimation based on the number of recurrent mutations needed for the perfect phylogeny.

3 RESULTS

3.1 Data sets

To evaluate the performance of HaploPool, we examined two data sets for which estimates of the haplotype frequencies were already available. These two data sets were derived by simulation from the phased haplotypes of the unrelated CEU individuals from HapMap (The International HapMap Consortium, 2003) phase 1. The CEU individuals (mother, father, child trios) were phased using PHASE (Stephens *et al.*, 2001). The first data set consisted of SNPs from the first 9 mb of Chromosome 19. The second data set contained two of the ENCODE regions, ENm010 and ENm013, which were each 500 kb regions of Chromosome 7. For our study, we took the phased haplotypes for the 60 unrelated parents as the starting point to simulate unrelated individuals. For each data set, Chr19 and ENCODE, we simulated a total of 600 unrelated haplotypes using the Li and Stephens (Li and Stephens, 2003) recombination model to simulate haploid gametes from recombined haplotypes already in the data set. The haplotype frequencies of these 600 unrelated haplotypes made up our *gold-standard* population haplotype frequencies against which we compared all frequency estimates. From the haplotype frequencies of the unrelated haplotypes, we drew (with replacement) the diploid individuals that made up the pool or genotype samples that were given as input to each phasing algorithm. The chromosomal regions of each data set were partitioned based on physical distance into 10 kb blocks. In order to consider a more realistic case where the SNP density is less than one SNP every 300 bases, we retained every second SNP in the ENCODE regions. These simulated data sets had a variety of block sizes ranging from 2 to roughly 45 SNPs. Other partitioning methods were considered in the Supplementary Material.

We compared the accuracy of HaploPool to three state-of-the-art algorithms for haplotype frequency estimation from pooled data. Three programs, LDPooled (Ito *et al.*, 2003), EHP (Yang *et al.*, 2003) and Pools2 (Hoh *et al.*, 2003), use the EM algorithm to estimate the haplotype frequencies. Pools2 actually estimates the multi-locus genotype frequencies, and we used PHASE to estimate the haplotype frequencies from these genotypes. For pools of size 1 (genotypes), we also compared our results to those of PHASE.

3.2 Accuracy of the estimations

In order to compare the accuracy of frequency estimation between the different methods, we compared each method to the gold-standard frequencies observed in the set of simulated haplotypes. We compared the predicted haplotype frequencies, f, from each data set of n pools to the gold-standard population-level haplotype frequencies, g. We used two natural measures: the l_1 and χ^2 distances. The l_1 distance between two distributions f and g is defined as $l_1(f,g) = \sum_{i=1}^{d} |f_i - g_i|$. The χ^2 distance between the two distributions is simply the result of the χ^2 statistic where g is the expected distribution, i.e. $\chi^2(f,g) = \sum_{i=1}^{d} (f_i - g_i)^2/g_i$ where d is the number of gold-standard haplotypes.

The results for the ENCODE regions are given in Figure 3. This plot illustrates two points. First, HaploPool consistently performed more accurately than Pools2 and LDPooled, while being as accurate as EHP. HaploPool was also much faster than the other computational methods for inferring haplotype frequencies from pools (see Supplementary Material). Second, we see that the most accurate estimation of the haplotype frequencies was achieved by using HaploPool on pools of size 4. As the pool size grew, the accuracy improved, provided that there were no errors in the genotyping. Even though PHASE gave slightly better results than HaploPool for genotype data, it would still be beneficial to genotype DNA pools and use HaploPool. For the same number of genotyping experiments, one can get more accurate information about the population

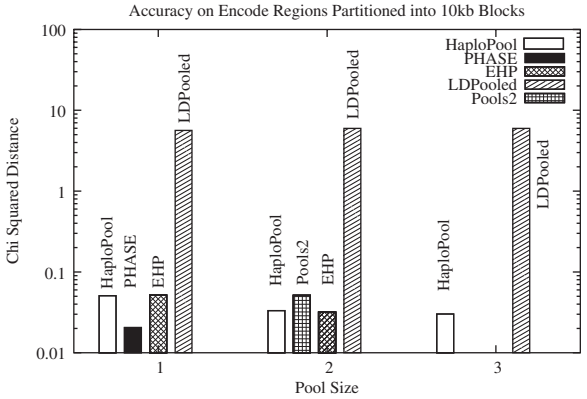

Fig. 3. Accuracy of haplotype frequency estimates on encode regions (10 kb partitions). This figure demonstrates a clear improvement in accuracy with larger pool sizes. After simulating a large population to obtain the gold-standard frequencies for comparison, we partitioned the regions into blocks of limited diversity using 10 kb physical distance. Each program ran on a random sample of 60 pools and returned frequency estimates. The χ^2 distance was computed from these frequencies and the gold-standard frequencies. Note that EHP appears only in this figure due to a running time in excess of a week on a 4-core 1.6 GHz Intel Xeon Server.

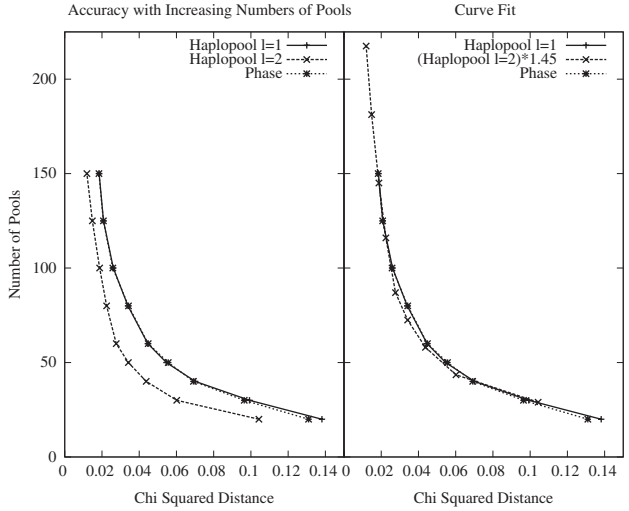

Fig. 4. The number of input pools affects accuracy. The left panel shows the accuracy of HAPLOPOOL and PHASE on the Encode data set (partitioned by physical distance) as the number of pools increase. In the right panel, we see an adjusted curve for HAPLOPOOL, $l=2$, where the y-values were multiplied by 1.45. We observe that the three curves are nearly identical. Thus, one would have to genotype 45% more individuals in order for PHASE to achieve the same accuracy as HAPLOPOOL.

from which the sample was drawn, and thus would gain power in the association study.

In order to quantify the efficiency of using DNA pools and HAPLOPOOL, we computed the number of genotyping experiments needed by traditional genotyping in order to achieve the same accuracy as HAPLOPOOL. In other words, we compared the scenario in which we use HAPLOPOOL to estimate the haplotype frequencies from n pools of two samples each, with the scenario in which we use PHASE to estimate the haplotype frequencies from αn genotypes. As shown in Figure 4, these two scenarios provided similar levels of accuracy of the haplotype frequencies when $\alpha \approx 1.45$. We found empirically that the accuracy of a study that genotypes $2n$ individuals using n DNA pools, of two samples each, is equivalent to the accuracy of a study that genotypes $1.45n$ individuals using traditional genotyping. Since the cost of genotyping a pool is close to the cost of genotyping an individual, the cost of genotyping in the second scenario has increased by close to 45%. Again, we are comparing the cost of genotyping either pooled or non-pooled DNA, because we assume that the cost of genotyping dominates the cost of sample collection and largely determines the cost of a study.

3.3 Haplotype assignments for pools

Our method is capable of predicting the haplotypes that comprised each input pool. This is done by using the estimated frequencies to choose the maximum-likelihood haplotype configuration for each pool. Haplotypes are determined for the whole pool, rather than for each individual. To find the haplotype prediction error, we count the haplotypes in each prediction that did not appear when the pool was simulated. On the data set that is a 10-kb partitioning of two ENCODE regions of Chromosome 7, our method correctly assigned haplotypes to genotype data with an average error rate of 1.3% and SD of 2.6%. Pools of size 2 had slightly more errors with a mean error rate of 2.6% and a SD of 4%.

3.4 Frequency estimations under noise

Even though the above results seem very promising, a justified objection to the use of DNA pools may be that the error rates and no-call rates for DNA pools of two, three or four individuals are higher than the rate for standard genotyping. The error and no-call rate for standard genotyping varies according to the platform, DNA quality and maintenance and other factors, but it is reasonable to expect it to be on the order of 1–2% missing data, and <1% error rate (Norton *et al.*, 2002).

In order to measure the effect of missing data and genotyping errors on the accuracy of the haplotype frequency estimation, we first simulated missing data by randomly masking out each allele count independently with probability p. We measured the performance of the algorithms for values of p ranging from 0.01 to 0.05 (data not shown). For these instances, we compared the results of HAPLOPOOL and PHASE as the other methods could not handle missing data. The missing data rate did not affect the results substantially. With 5% missing data and a fixed genotyping budget, the accuracy of HAPLOPOOL on pools of size 2 is much better than the accuracy of PHASE on genotype data with no missing data.

We simulate genotyping error by adding a Gaussian error with SD σ to each called allele frequency. Let the correct allele frequency be a_{ij} where i is the pool and j is the SNP, and let $\hat{a}_{ij}$ be the miscalled frequency. Then, we pick the value $\hat{a}_{ij}$ from the distribution $\hat{a}_{ij} = a_{ij} + x$ where $x \sim \mathcal{N}(0, \sigma^2)$. After simulating

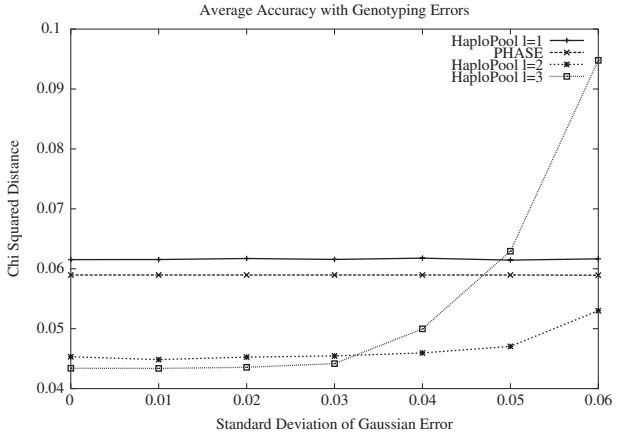

Fig. 5. Genotyping errors in chromosome 19 data set. Genotyping errors have a greater effect on accuracy when the pools are of a larger size. In order for the estimates to be unaffected by the error, σ must be smaller than the distance in frequency between calls that can be mistaken for each other. Since genotypes have a frequency leeway of 0.5, the calls, and thus the accuracy, are unaffected. Pools of size 2 provide the advantage of increased accuracy while remaining robust to bad calls when $\sigma < 0.05$.

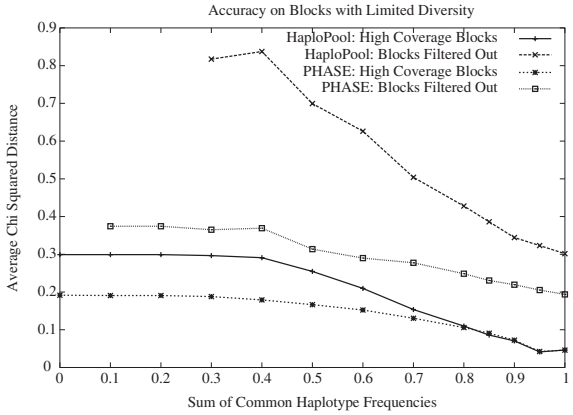

Fig. 6. Effect of haplotype diversity on genotype accuracy. Blocks whose HaploPool and PHASE estimates contained haplotypes of limited diversity were more accurate than blocks with more diversity. The lines marked with pluses and stars give the accuracy of HaploPool and PHASE, respectively, when given non-pooled genotypes on the blocks with diversity limited by the threshold on the x-axis. The x- and box-marked lines are the accuracies for each method on the blocks that were filtered out by the coverage threshold. The data used here was obtained by creating blocks of 10 SNPs each from the first 9 mb of Chromosome 19.

these perturbed allele frequencies, we discretize the resulting frequencies to produce perturbed allele counts that are consistent with the number of haplotypes in each pool. Figure 5 shows the results of these simulations for σ ranging from 0 to 0.06. Traditional genotyping is not affected by these values of σ, as the difference in frequency between the possible observations of frequencies (0,0.5 and 1) is much larger than the SD. Notably, Figure 5 shows that as long as the SD of the allele frequency call is smaller than 0.05, estimating haplotype frequencies from pools of size 2 is advantageous over standard genotyping.

3.5 Limited diversity indicates accurate frequency estimates

Both PHASE and HaploPool produced frequency estimates with varying degrees of accuracies over all the blocks. Indeed, in the case of a block containing a recombination hot-spot, our assumption of limited diversity would be violated and neither the perfect phylogeny model nor the regression method could be expected to produce an acceptable estimate. Therefore, it is useful to characterize the circumstances under which we can have high confidence in the predictions. This characterization is also applicable when haplotype frequencies are predicted from traditional genotype data.

We observe that limited diversity can serve as a predictor of the accuracy of haplotype frequency estimations. All the methods are more accurate (data not shown) in blocks for which the set of common haplotypes (with frequency at least 5%) accounts for more than 90% of the population (referred to as 90% *coverage*). In Figure 6, for the blocks with lower coverage, the haplotype frequencies are estimated poorly by HaploPool. The same figure shows a similar trend when using PHASE on the same data. The Supplementary Material give an example where the advantage of using pools is not substantial unless restricted to blocks of limited diversity.

Thus, when performing haplotype-based analysis in association studies, we recommend using the HapMap database to devise a partitioning for the genomic regions under consideration. These partitions either can be deliberately chosen to exclude known recombination hot-spots, or can be found automatically using a method that chooses block boundaries that respect limited diversity. Two such automatic partitioning methods are Hap (Halperin and Eskin, 2004) and Haploview (Barrett *et al.*, 2005). The influence on estimation accuracy of various partitioning methods is discussed in the Supplementary Material.

After collecting data and estimating haplotype frequencies, we recommend focusing haplotype-based statistical tests on blocks for which the coverage is at least 90%. Since blocks with less coverage tend to have less reliable haplotype estimates, *P*-values obtained from them could be inflated and misleading. Rather than risk false-negatives due to haplotype frequency estimation, we recommend using allele-frequency statistics on low coverage blocks. These recommendations are universal, and they apply regardless of whether the study uses pooled or non-pooled DNA and regardless of which phasing program is used.

4 DISCUSSION

In this article, we have presented HaploPool, a novel method for inferring haplotype frequencies from disjoint pools of unrelated individuals. HaploPool is based on the perfect phylogeny model combined with a regression analysis that aggregates frequency predictions for subsets of SNPs. Although the perfect phylogeny model has been used in the past for phasing (Halperin and Eskin, 2004), deviations from the model were treated in a less formal manner than in the present article, and previous algorithms could not extend easily to DNA pools. Our model, although designed to work with DNA pools, can also be used for phasing traditional genotype data.

It is not surprising that, relative to genotypes, pools allow more possibilities for incorrectly pairing alleles together in the same haplotype. Yet, when restricting our attention to pools of a fixed size, we were surprised to find that the perfect phylogeny model has greater ability to detect haplotype frequencies in the absence of recombination than the EM model. We hypothesize that the constraints given by the perfect phylogeny better model the data, thus providing better estimates than the constraints provided by the EM.

The results in this article illustrate that DNA pools can be used to obtain more accurate haplotype frequency estimates than traditional genotyping methods using an equal number of genotyping experiments. We show that by using pools of two or three individuals, one can obtain better frequency estimates than are available from performing the same number of genotyping experiments using traditional genotyping. Since pools of size 2 and three use the same genotyping technology, the cost for running the pooling experiments is roughly equivalent to running normal genotype experiments. There is a small increase in the cost due the quantization of the DNA samples, but as noted in Beckman *et al.* (2006), this cost (in their hands $5 per sample) is negligible compared to the cost of genotyping. Hence, according to our study, genotyping n pools of size 2 each is roughly equivalent in cost to the genotyping of n individuals using traditional genotyping, but it is roughly equivalent in accuracy to the genotyping of $1.45n$ individuals using traditional genotyping.

Haplotype frequency accuracy for all four methods examined here was highly dependent on the concentration of the haplotype distribution. For example, a block containing a recombination hot-spot would result in many distinct haplotypes in the sampled individuals and little concentration in the haplotype distribution. In blocks with little concentration, neither our methods nor the other comparable methods produced reliable haplotype frequency estimates. In this article, we defined a criterion for the reliability of a frequency estimate, which depends on the diversity of the haplotype distribution in the region. Since phasing errors can affect the power of an association study, we believe that the criterion of limited diversity may be useful for quality control.

The novel results offered here show that the accuracy of pooling is competitive with traditional genotyping even in the presence of errors. Both the improved running time and the better accuracy obtained by our method over previous approaches made this conclusion possible. Our comparison to existing programs shows that the other methods for haplotype frequency estimation from pools are probably not competitive with PHASE.

The effect of simulated errors suggests that pools of two individuals provide a good balance between increasing accuracy and decreasing robustness to error (Fig. 5). It appears that such small pools are robust to realistic values of genotyping errors. Since errors exhibit a large influence for pools of size greater than 2, we conclude that large pooling methods are not competitive with traditional genotyping. Clearly, the effect of errors on accuracy heavily depends on the specific genotyping platform, and we believe that benchmarking these methods on different platforms is essential.

ACKNOWLEDGEMENTS

B.K. was supported by the DOE Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308. E.H. and R.K. were supported by NSF grant IIS-0513599. We thank Gad Kimmel for the software to simulate haplotypes using recombination. In addition, we are grateful for the helpful suggestions made by the anonymous reviewers.

Conflict of Interest: none declared.

REFERENCES

Barcellos,L.F. *et al.* (1997) Association mapping of disease loci, by use of a pooled DNA genomic screen. *Am. J. Hum. Genet.*, **61**,734–747.

Barratt,B.J. *et al.* (2002) Identication of the sources of error in allele frequency estimations from pooled DNA indicates an optimal experimental design. *Ann. Hum. Genet.*, **66**,393–405.

Barrett,J.C. *et al.* (2005) Haploview: analysis and visualization of LD and haplotype maps. *Bioinformatics*, **21**,263–265.

Beckman,K.B. *et al.* (2006) Using DNA pools for genotyping trios. *Nucleic Acid Res.*, **34**,e129.

Carlson,C.S. *et al.* (2004) Mapping complex disease loci in whole-genome association studies. *Nature*, **429**,446–452.

de Bakker,P.I.W. *et al.* (2006) Transferability of tag SNPs to capture common genetic variation in DNA repair genes across multiple populations. *Pac. Symp. Biocomput.*

Germer,S. *et al.* (2000) High-throughput SNP allele-frequency determination in pooled DNA samples by kinetic PCR. *Genome Res.*, **10**,258–266.

HalperinE. and Eskin,E. (2004) Haplotype reconstruction from genotype data using imperfect phylogeny. *Bioinformatics*, **20**,1842–1849.

Halperin,E. and Karp,R.M. (2004) Perfect phylogeny and haplotype assignment. In *RECOMB '04: Proceedings of the Eighth annual International Conference on Resaerch in Computational Molecular Biology*, ACM Press. New York, USA, ISBN 1-58113-755-9, 10–19.

Hoh,J. *et al.* (2003) SNP haplotype tagging from DNA pools of two individuals. *BMC Bioinformatics*, **4**,14.

Ito,T. *et al.* (2003) Estimation of haplotype frequencies, linkage-disequilibrium measures, and combination of haplotype copies in each pool by use of pooled DNA data. *Am J. Hum. Genet.*, **72**,384–398.

Kingman,J.F.C. (1982) The coalescent. *Stoch. Proc. Appl.*, **13**,235–248.

Le Hellard,S. *et al.* (2002) SNP genotyping on pooled DNAs: comparison of genotyping technologies and a semi automated method for data storage and analysis. *Nucleic Acids Res.*, **30**,e74.

Li,N. and Stephens,M. (2003) Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genet.*, **165**,2213–2233.

Morris,R.W. and Kaplan,N.L. (2002) On the advantage of haplotype analysis in the presence of multiple disease susceptibility alleles. *Genet. Epidemiol.*, **23**,221–223.

Norton,N. *et al.* (2002) Universal, robust, highly quantitative snp allele frequency measurement in DNA pools. *Hum. Genet.*, **110**,471–478.

Patil,N. *et al.* (2001) Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, **294**,1719–1723.

Pe'er,I. and Beckmann,J.S. (2003) Resolution of haplotypes and haplotype frequencies from SNP genotypes of pooled samples. In *RECOMB' 03: Proceedings of the Seventh annual International Conference on Research in Computational Molecular Biology*, ACM Press. New York, USA, ISBN 1-58113-635-8, 237–246.

Sham,P. *et al.* (2002) DNA pooling: a tool for large-scale association studies. *Nat. Rev. Genet.*, **3**,862–871.

Stephens,M. *et al.* (2003) A new statistical method for haplotype reconstruction from population data. *Am. J. Hum. Genet.*, **68**,978–989.

The International HapMap Consortium (2003) The international HapMap project. *Nature*, **426**789–796.

Yang,Y. *et al.* (2003) Efficiency of single-nucleotide polymorphism haplotype estimation from pooled DNA. *Proc. Natl Acad Sci.*, **100**, 7225–7230.