Haplotype Reconstruction in Large Pedigrees with Untyped Individuals through IBD Inference

XIN LI and JING LI

ABSTRACT

Haplotypes, as they specify the linkage patterns between dispersed genetic variations, provide important information for understanding the genetics of human traits. However, haplotypes are not directly obtainable from current genotyping platforms, which pushes extensive investigations of computational methods to recover such information. Two major computational challenges arising in current family-based disease studies are large family sizes and many ungenotyped family members. Traditional haplotyping methods can neither handle large families nor families with missing members. In this article, we propose a method that addresses these issues by integrating multiple novel techniques. The method consists of three major components: pairwise identical-by-descent (IBD) inference, global IBD reconstruction, and haplotype restoring. By reconstructing the global IBD of a family from pairwise IBD and then restoring the haplotypes based on the inferred IBD, this method can scale to large pedigrees, and more importantly it can handle families with missing members. Compared with existing approaches, this method demonstrates much higher power to recover haplotype information, especially in families with many untyped individuals. Availability: <http://sites.google.com/site/xinlishomepage/pedibd>.

Key words: haplotype inference, identical-by-descent (IBD), inheritance, linkage disequilibrium.

1. INTRODUCTION

HUMANS ARE DIPLOID, WITH TWO HOMOLOGOUS CHROMOSOMES EACH FROM ONE PARENT. When inherited from a parent to a child, genes on one chromosome tend to stay together unless meiotic recombination breaks such linkage. Haplotypes, as they represent such linkage information, are critical for understanding the genetics of human diseases (Bader, 2001; Frazer et al., 2007; Morris and Kaplan, 2007). However haplotype information is not obtained from current genotyping platforms and needs to be recovered through computational methods. Current disease studies in families pose two major challenges to haplotype inference.

First is family size. In order to detect enough recombination breakpoints to narrow down the disease loci, it is desirable to recruit as many family members as possible to a study. However, current haplotyping methods are not scalable to large families, as most of them need computing time exponentially growing with family sizes. The sluggish performance on large families is attributed to the Lander-Green algorithm (Lander and Green, 1987), which serves as their methodological backbone. Recent optimizations to the

Lander-Green scheme (Abecasis et al., Gudbjartsson et al., 2005; Kruglyak et al., 1996; Sobel and Lange, 1996) reduce the absolute computing time but does not alter its inherent exponential nature. There exist another group of methods, which exploit the Mendelian law of inheritance instead of enumerating the inheritance patterns and work much more efficiently (Li and Jiang, 2003; Li and Li, 2009; Xiao et al., 2007). However, these methods heavily depend on direct parent-child relationships.

Second, this limitation leads to another inescapable problem—ungenotyped family members. In a typical family-based study, many individuals in a family are not available for genotyping because they are deceased or otherwise not recruited. Given untyped family members, the Mendelian constraints cannot be applied effectively. Most rule-based methods handle this issue by enumerating the genotypes of these untyped individuals, which turns out to be computationally infeasible if many family members are missing.

We have recently developed an algorithm to efficiently infer identical-by-descent (IBD) status among family members (Li et al., 2010). The approach overcomes these computational challenges by first constructing hidden Markov models (HMMs) for all relative pairs with genotypes, and then constructing the global (pedigree-wise) IBD relationship from the inferred pairwise IBD relationships from these HMMs. We use the “inheritance-generating function” to trace the inheritance relationship between any two relatives, this innovation allows us to bypass the enumeration of genotypes of these untyped family members. The inheritance-generating function can be efficiently calculated using a recursive formula similar to the calculation of kinship coefficients. Essentially, the method solves the computational problem of large pedigrees. However, at the final step, the approach uses an enumerative procedure to restore the inheritance from pairwise IBD, which again is exponential to the family size. In this article, we replace it with a much more efficient algorithm based on graph partitioning. On top of that, we integrate our previous linear system based haplotyping method (Li and Li, 2009) into the framework to recover allelic phases of each individual. The haplotyping method exhausts all available constraints imposed by inheritance and genotypes, which maximizes the usage of information in a family. All together, this work constitutes a new haplotyping scheme which can efficiently reconstruct haplotypes at a genome-wide level in a large family with many untyped individuals. The two-stage IBD inference and the subsequent haplotype reconstruction significantly alleviate the computational burden complicated by large families. We evaluate the effectiveness of our method on both real and synthetic datasets. On families with many untyped individuals, our method exhibits significantly higher power in recovering haplotypes as compared with other state-of-art haplotyping methods. The proposed method also demonstrates good scalability on large pedigrees which other methods cannot handle.

To give a clear picture of the method, we summarize its steps in a flowchart (Fig. 1). The method consists of three steps: pairwise IBD inference from genotypes, global IBD reconstruction from pairwise IBD, and haplotype reconstruction from global IBD. Within each step, a computational technique is employed, namely, the HMM, graph partition, and disjoint-set data structures. We will present the details of each of the three steps in Sections 2.1, 2.2, and 2.3. We will show the performance of the method in Section 3.

2. METHODS

2.1. Inference of pairwise IBD

The first step involves how to infer pairwise IBD sharing between relatives. This method is introduced in Li et al. (2010). We briefly reiterate its essential elements here for the completeness of the article. The method to infer pairwise IBD involves two key components: (a) construct an HMM for pairs of relatives, and (b) incorporate population level linkage disequilibrium into the model.

a. HMM for a pair of relatives. The IBD status between two individuals can be modeled using a 2-state HMM, with transition probabilities between IBD and non-IBD states determined by the degree of relatedness between them. In order to quantify such a relationship, we propose the “inheritance-generating function” to summarize all possible inheritance paths between two individuals. Intuitively, the longer the inheritance path between the two alleles, the less the probability that they descend from the same ancestral allele, as a longer inheritance path involves more segregations. In addition to transition probabilities, to fully parameterize the proposed HMM, we also specify the emission probabilities. Given that two alleles are IBD, they must be the same genotype. On the other hand, if two alleles are not IBD, it is purely due to chance for them to be the same genotype, and such a probability can actually be determined by the allele

FIG. 1. Flowchart of the method.

frequencies at this locus. We further extend the model to incorporate more complex situations of missing genotypes and genotyping errors by refining the emission probabilities. HMMs between a pair of individuals can then be derived based on HMMs between pairs of alleles. The decoding process is done by applying routine procedures of hidden Markov models, we use the Viterbi algorithm for maximum likelihood inference and the forward-backward algorithm for point-by-point posterior probabilities. Both approaches take time linear to the number of markers. Results from both decoding algorithms will be utilized later.

b. Incorporating LD. Linkage disequilibrium at the population level, which largely reflects distant ancestral sharing among individuals, may produce short identical haplotype segments among seemingly unrelated individuals. We quantify such allelic dependence by adding an additional state (called the “LD state”) to the HMM, which explicitly tags short stretches of IBD originating from non-family relatedness. By doing so, one can make full use of the information embedded in the whole range of all available markers. Furthermore, by directly modeling linkage disequilibrium as distant ancestral sharing, we end up with a unified hidden Markov framework, with more versatile power to fit the data because it allows synergistic interaction between IBD and LD state. The parameters related to the LD state in the model are learned from the targeted data, where we estimate the closeness of two unrelated individuals based on their genome-wide allelic sharing. The transition probabilities from the IBD state to the non-IBD and LD state are proportionally distributed according to their prior probabilities since non-IBD and LD both refer to alleles of distinct founder origins and are thus indistinguishable on a single family basis. With the help of the LD state, we can dissect the effects arising from relatedness or linkage disequilibrium.

2.2. From pairwise IBD to global IBD

Given that two individuals share one allele IBD, there is still ambiguity which one of the two homologous alleles of an individual is shared with the other individual. However, we want to further recover this information, or more specifically, we want to reconstruct the global IBD which explicitly labels each of the two homologous alleles of an individual with their ancestral alleles. In this section, we will first introduce a method for an ideal situation where all pairwise IBD relationships are consistent. In the second part of this subsection, we will present an alternative backup approach for inconsistent data by utilizing posterior decoding.

FIG. 2. Partition of the neighbors of an individual into two ancestral groups, where each group forms a clique.

To construct the global IBD in a family, a simple approach is to enumerate all possible inheritance patterns and check their consistency with pairwise IBD, which was implemented in Li et al. (2010). However, this algorithm is not efficient. Here, we introduce a new approach in a graph theory setting. We first define an IBD graph to organize the relationships among all individuals in a pedigree. In brief, all individuals sharing two alleles will be merged into one node. Individuals sharing one allele will be connected by an edge. The groups of people who share the maternal or paternal allele of an individual can be recognized by finding two distinct cliques among its neighbors. By starting at one individual and iteratively propagating the paternal and maternal partition onto the neighboring individuals, we can settle the global IBD sharing. We will discuss the details of this approach in the following order: First, we will formally define how to construct an IBD graph. Second, we will describe how to partition the neighbors of an individual into paternal sharing and maternal sharing groups and how to propagate such information further onto neighbors' neighbors. Third, we will give a proof of the correctness of this procedure. Fourth, and last, we will discuss some special cases not covered in the algorithm.

We construct an IBD graph based on the pairwise IBD sharing between family members. Individuals sharing two alleles IBD with each other are identical; thus, we use a single node to represent them. We use an edge to indicate the relationship of sharing one allele IBD. Formally speaking, let $G = (V, E)$, where $V = \{v_i | v_i = \{i\} \cup \{j | IBD(i, j) = 2\}\}$, $(v_a, v_b) \in E$ if $IBD(i, j) = 1, i \in v_a, j \in v_b$. Here, we assume the pairwise IBD relationships are consistent; therefore, picking whichever two individuals respectively from two nodes, their relationships should be coherent.

Before getting into details of the algorithm, we first introduce some basic notations. We use $A = \{a_1, \dots, a_n\}$ to indicate n different ancestral alleles in a family.¹ We define $A_i = \{x_i^1, x_i^2\}$ to be the ancestral configuration of individual i , where $x_i^1, x_i^2 \in A$ specifies the ancestral sources of each of the two homologous alleles. We assume homologous alleles are of different ancestral origins, i.e., $x_i^1 \neq x_i^2$. We define an operation $a_k \rightarrow A_i$ to indicate assigning ancestral allele a_k to whichever x_i^1 or x_i^2 that is not yet assigned.

The algorithm starts by picking one individual from the family. Assume that this individual has two homologous alleles of distinct ancestral sources, which we denote as $\{a_1, a_2\}$, $a_1 \neq a_2$. Consider all of its neighbors in graph G , they must either carry a_1 or a_2 , and based on this we can partition them into two groups, which we denote as N_1 and N_2 . It is not hard to notice that both N_1 and N_2 are fully connected cliques, whereas between N_1 and N_2 there are a restricted number of edges, or more specifically, any node in N_1 can have at most one connection to N_2 and vice versa. Figure 2 gives such an example showing the neighbors of a node and how they form two cliques. This feature can help us perform such a partition quite efficiently. Once we obtain this partition, we can subsequently assign ancestral alleles to each of the two

¹Notice that the ancestral alleles are just labels to be assigned to different individual alleles; they are not in any particular order, nor explicitly associated with any particular founders in a family.

groups and further to their neighbors. These two procedures constitutes two basic steps of the algorithm: we call the initializing step “seeding” and the iterative step “propagation.” We formally define the procedures of “seeding” and “propagation” in Algorithms 1 and 2.

Algorithm 1 Seeding

```

Find an individual  $s$  with more than 4 neighbors,  $|N(s)| > 4$ .
Partition  $N(s)$  into two cliques:  $N_1(s)$  and  $N_2(s)$ .
for  $k = 1, 2$  do
     $a_k \rightarrow A_s$ 
    for each individual  $A_j \in N_k(s)$  do
         $a_k \rightarrow A_j$ 
    end for
end for

```

Algorithm 2 Propagation

```

 $K = 2$ 
while there is any individual that is partially assigned do
    Find the next partially assigned individual  $A_i = \{a_m, x\}$ .
     $K = k + 1$ 
     $a_k \rightarrow A_i$ 
    for each neighbor  $j$  of individual  $i$  do
        if  $a_m \notin A_j$  then
             $a_k \rightarrow A_j$ 
        end if
    end for
end while

```

In the propagation step, we make a queue to store all the newly assigned yet not fully assigned individuals. In this way all individuals will be visited at most twice in this process, hence the propagation step can be finished in linear time with respect to the number of individuals. We can prove by induction that after the seeding and the propagation step all ancestral alleles are correctly assigned to each individual.

Lemma 1. *After the seeding and propagation step, for any ancestral allele a_i , it is assigned to all of the individuals who carry this allele and none of the individuals who do not carry this allele.*

Proof. Basis: In the seeding step, $A_s = \{a_1, a_2\}$, any individual containing ancestral alleles a_1 or a_2 must be a neighbor of s , and on the other hand, any neighbor of s must either share a_1 or a_2 with s and we have partitioned them accordingly into two groups, therefore they will all receive proper assignment.

Induction hypothesis: Assume that at step k of the propagation, ancestral alleles $\{a_1, \dots, a_k\}$ are all correctly assigned.

Induction: At step $k + 1$, assume $A_j = \{a_m, x\}$, $m \leq k$ is an individual not yet fully assigned, applying the induction hypothesis, x must be an ancestral allele not in $\{a_1, \dots, a_k\}$. Thus, we let $x = a_{k+1}$. Any neighbor of j which does not contain a_m must contain a_{k+1} , we can safely assign a_{k+1} to them. On the other hand, any individual which contains a_{k+1} must be a neighbor of j , therefore at step $k + 1$, any individual containing ancestral allele a_{k+1} receives proper assignment. ■

The iterative propagation step will not stop until all nodes in a connected component are fully assigned. Different connected components of the IBD graph can be handled independently by applying the “seeding” and “propagation” procedures individually on each of them. Notice that it is not necessary to distinguish paternal alleles from maternal alleles when assigning ancestral allele types. Since pairwise IBD relationships stay unchanged over a chromosomal region if no recombination occurs, it is not necessary to process each locus individually, instead we generate one single global IBD configuration for all loci of a recombinant-free segment.

There are two situations we have not yet addressed, the first is when we cannot find a node with more than 4 neighbors in the “seeding” step. If a node has 4 or fewer neighbors, the partition of these neighbors can be ambiguous. Figure 3 shows such an example, where either way of partition can be a possible

FIG. 3. Alternative partitions for fewer than 5 neighbors.

configuration. In this case, we should consider both of these two partitions and we will eventually have two alternative IBD assignments in the end. The second situation is when an individual is inbred, i.e., inheriting two homologous alleles from the same ancestor. Since the paternal and maternal alleles are identical, this individual can have only one group of relatives, i.e., one clique in the IBD graph. In this situation, we only need to propagate its allele to this single group of neighbors and beyond that everything stays the same in the “seeding” and “propagation” processes. One situation that can be confused with the inbreeding is when an individual has actually two distinct ancestral alleles but one of them is not shared with any other individual. In this case, the inbreeding and non-inbreeding situations are not distinguishable. Therefore, in the case that one person has only one clique of neighbors, the assignment of one of its two alleles is always ambiguous. Both assignments in such a case are valid.

Finding optimal inheritance for error-prone data. The graph partition procedure introduced above takes the pairwise IBD relationships for sure (e.g., using Viterbi decoding) and assumes all pairwise relationships are consistent. However when there are errors, we may not be able to find a global IBD configuration which satisfies all pairwise relationships. In these situations, we should have a backup plan which can tolerate possible inconsistencies. Here, we introduce an alternative enumerative approach utilizing results from posterior decoding. The problem is essentially formulated as an optimization problem where the search space consists of all possible inheritance patterns and the optimization criterion is defined below. Intuitively, we try to accommodate as many high probability pairwise IBD relationships as possible. We define a target function to aggregate the information over all pairwise relationships. In a straightforward way, we can use the following pseudo-likelihood function which encapsulates all pairwise relationships by multiplying their posterior probabilities.

$$L(\{A_1, \dots, A_n\}) = \prod_{i,j} L(\text{IBD}(A_i, A_j)),$$

where the product is over all unordered pairs of individuals in a family and A_i is a specific ancestral allele assignment of individual i . By maximizing the target function, we are essentially trying to accommodate as many pairwise relationships of higher confidence levels as possible while sacrificing a few of those of lower confidence levels.

Since we need to maximize the target function over all possible inheritance patterns of a family, the search space could be rather huge. A straightforward search involves the enumeration of 2^{2k} transmissions, where k is the number of non-founders in a family. To address this problem, we use a branch-and-bound strategy to escape unnecessary configurations. We take advantage of the property that a partial assignment always has a larger likelihood than that of a full assignment.

$$L(\{A_{i_1}, \dots, A_{i_m}\}) \geq L(\{A_1, \dots, A_n\}), \{A_{i_1}, \dots, A_{i_m}\} \subset \{A_1, \dots, A_n\}$$

Therefore, once the fit of a partial assignment drops below the optimal frontier, we can safely discard this assignment and backtrack.

2.3. Haplotype reconstruction

Genotypes can be phased at each marker according to the corresponding global IBD. Intuitively, we can first focus on individuals carrying homozygous alleles because they are naturally phased and we can thus

Type 1	Type 2	Constants
Assume individuals $\mathbf{x}, \mathbf{y}$ have ancestral allele assignments $x = (a_x^1, a_x^2) \quad y = (a_y^1, a_y^2)$	any heterozygous individual: $\mathbf{x}$	any homozygous individual: $\mathbf{x}$
if $a_x^p = a_y^q, \quad p, q \in \{1, 2\}$ then $x(p) = y(q)$	$x(1) = x(2) + 1$	$x(1) = x(2) = \text{constant}$

FIG. 4. A binary linear system of alleles. $x(1), x(2)$ refer to the two homologous alleles of an individual x . The shown system is for one locus. Homologous alleles $x(1), x(2)$ of an individual x across different loci are organized by their indices (1 or 2), same indices form a haplotype.

resolve the genotypic values of the ancestral alleles inherited by them. This information is subsequently used to phase the other individuals sharing the same ancestral alleles and so on so forth. However, from a strict mathematical perspective, both homozygous and heterozygous loci carry some information to resolve these uncertainties. To be more specific, the constraints imposed by the global IBD and genotypes actually form a binary linear system.

Given the global IBD, the alleles of family members form two basic relationships and both can be explicitly formulated by binary linear equations. The first type of relationship is imposed by shared ancestry, which enforces that descendant alleles originating from the same ancestral allele should be the same. This type of relationship can be expressed as equivalence in a linear equation (Fig. 4, Type 1). The second type of relationship is imposed by heterozygous alleles, which dictates that the paternal allele and maternal allele of a heterozygous individual must be complementary to each other (Fig. 4, Type 2). This relationship can be expressed as $+1$ equivalence in a binary equation. A binary system naturally embeds the property that double complements should lead back to equivalence. As type 1 constraints are imposed by the global IBD, they are segment-wise, i.e., same for each locus of a recombinant-free region. On the other hand, type 2 constraints are locus-specific as they are related to genotypes. At each locus, the entire constraint system will appear as illustrated in Figure 4, where two types of constraints and the constants are enforced. We treat each pair of heterozygous alleles and each missing allele as variables and each homozygous allele as a constant, we build the binary linear system by enforcing both types of constraints.

We can solve these equations just like solving a conventional linear system, however we may do it in a more efficient manner using a special data structure called disjoint-set. The general idea behind this is that we use disjoint-sets to represent independent variable sets and manipulate these sets (using “union-find” algorithm) to encode different constraints. By doing so, we can quickly generate a solution or detect inconsistency of the system in linear time. We omit further technical details here, the algorithmic procedures for efficiently solving such a system are described in Li and Li (2009). Figure 5 shows an example how the phases of each pair of alleles are resolved in a family by enforcing the constraints imposed by the global IBD.

3. EXPERIMENTS

We evaluate the performance of our method on both synthetic and real datasets. To examine its power to recover haplotypes, we run the method on a family with simulated genotypes. The family is drawn from a real data study assayed on the Affymetrix 6.0 SNP chip. This study has a total of 24 families and we use the available allele frequencies and haplotype segments from all these families to generate the appropriate

FIG. 5. Genotypes phased according to the global IBD.

founder haplotypes in order to mimic the actual linkage disequilibrium in the data. Recombination rate is simulated at 1cM per Mb. The SNP map, missing genotype rate and typing error rate used in the simulation are exactly the same as those of the real data, which we assume are typical of the Affymetrix 6.0 SNP chip.

Here, we examine the contribution of different relatives in resolving the haplotypes of an individual. We start with a family with only three typed individuals and gradually increase the number of individuals one by one in each subsequent experiment. We compare the effectiveness of our method (named PED-IBD) with that of MERLIN. MERLIN (Abecasis et al., 2002) is a popular linkage analysis software package implementing the Lander-Green algorithm. Figure 6 shows the family structures and the ratio of phased loci. In the first setting, three of the family members are genotyped however none of them form direct parent-child relationships. In this situation, our method can correctly phase approximately 22% of the heterozygous loci. In comparison, MERLIN cannot recover any loci. The purpose of the first setting is to examine the power of each method when no parent-child Mendelian constraints can be obtained. PED-IBD has obviously higher phased ratio than MERLIN in the beginning three settings. The performance of MERLIN catches up only after direct parent-child pairs are added as in the last two settings. This phenomenon suggests that MERLIN relies heavily on close relationships to resolve the uncertainty, but our method can make better use of all available information in the pedigree. Comparing the precision, which is the correctly phased loci out of all phased loci, two methods are similar, with the precision of our method at 94.04%, 93.69%, 96.40%, and 97.81%, and MERLIN at 94.21%, 94.41%, 95.38%, and 95.90% for the last four settings. We also simulate a big family of 21 members with 11 genotyped individuals, MERLIN quit halfway in running this family presumably because of the exponential memory requirement or time complexity involved. Figure 7 shows the ratio of phased loci yielded by our method on different members of the family. The left two columns show the overall ratio of total phased loci and correctly phased loci of this family. The rest of the columns are results from individual family members. The result agrees with a common sense that individuals with more close relatives generally get higher ratios of their allelic phases resolved. Direct parent-child relationships also offer a major contribution here, where individuals having a genotyped parent or child have significantly more phased loci than others. The running time of the method scales quadratically with the number of genotyped individuals and linearly with the number of markers. On this specific family of 11 genotyped individuals, the program takes around 5 minutes to finish 10 K markers on a regular PC.

FIG. 6. Proportion of total phased and correctly phased loci in families with different numbers of genotyped individuals. From left to right, four grouped bars represent total phased loci and correctly phased loci of PED-IBD, total phased loci and correctly phased loci of MERLIN, out of all heterozygous loci. In the pedigree diagrams, shaded nodes indicate genotyped individuals.

FIG. 7. Proportion of total phased and correctly phased loci for different members of a family. Shaded nodes are genotyped individuals.

The second data set we use is from a real data study of hypertension. These families and their members are collected according to familial aggregations of the disease therefore the family sizes and their structures reflect a typical pattern in many family based studies. Here we want to evaluate the power of our method under a realistic distribution of family structures, as this may provide some empirical basis in assessing the effectiveness of the method for other real datasets. We have a total of 196 families, among them 141 families have more than 1 typed members and there are an average of 4 typed individuals in each of these families. All families are genotyped on Affymetrix 6.0 SNP array. We examine four factors on the effectiveness of phasing: family size (number of typed members), relationship between family members, missing genotypes and genotyping errors. Statistics (Fig. 8 and 9) of different families are binned according to the number of typed individuals in each family. The line in each figure indicates the averaged values of all families in each bin.

Figure 8a shows the proportion of loci that are phased out of all heterozygous loci given different numbers of typed individuals in a family. In general more typed individuals add more information to the family and lead to a higher phased ratio. However, the relationships between individuals also make a difference. Breaking down the phased ratios in families of two typed members (Fig. 8b), we can observe that parent-child relationships are much more powerful than others in resolving the phase uncertainty. Full-sibship does not offer any gain than half-sibship because full siblings can occasionally share both paternal and maternal haplotypes at certain chromosomal regions become mutually non-informative. The influence of missing genotypes is minor, as shown in Figure 9a, most of the missing genotypes can be imputed. Genotyping errors turns out to be the final bottleneck. As demonstrated in Figure 8a, most loci can be unambiguously phased given a large enough family size, however that proportion quickly approaches an

FIG. 8. (a) Proportion of heterozygous loci that are phased in different families. (b) Proportion of phased loci for different relationships.

upper bound. The total phased ratio fluctuates around 96% for families above the size of 5. To see how this major drawback is caused by genotyping errors, examining genotypes against the inheritance patterns of these families (Fig. 9b), we can observe that around 3% of loci are not consistent with inheritance. Large families are generally more sensitive to typing errors because the effect of an error propagates through inheritance. Therefore, among these four factors, genotyping errors are the biggest drawback on the effective yield of haplotypes.

4. DISCUSSION

In this article, we introduce a new method to efficiently reconstruct haplotypes in large families allowing many ungenotyped individuals. Our approach consists of three major steps: pairwise IBD inference, global IBD reconstruction and haplotype restoring. By taking a two-step—genotype to pairwise IBD, pairwise IBD to global IBD—approach, we can significantly reduce the time complexity for resolving the IBD sharing pattern among family members. This makes our method well scalable to large families which traditional methods cannot handle. The haplotyping method is based on linear systems, it exhausts all available constraints imposed by global IBD and genotypes, thus maximizes the usage of information in a family. Comparing with other popular methods, our method has a much higher power to recover allelic phases in families with many missing members. On a real dataset of 196 families, the method yield more

FIG. 9. (a) Proportion of missing loci that are imputed. (b) Proportion of consistent loci.

than 90% phased loci on families with more than five typed individuals. The proposed method is an important advance to bridge the technical gap in existing methods on large families containing untyped members.

ACKNOWLEDGMENTS

We would like to thank Dr. Xiaofeng Zhu for helpful discussions. This research is supported by National Institutes of Health/National Library of Medicine (grant LM008991) and in part by National Institutes of Health/National Center for Research Resources (grant RR03655).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

- Abecasis, G.R., Cherny, S.S., Cookson, W.O., et al. 2002. Merlin—rapid analysis of dense genetic maps using sparse gene flow trees. *Nat. Genet.* 30, 97–101.
- Bader, J. 2001. The relative power of SNPs and haplotype as genetic markers for association tests. *Pharmacogenomics* 2, 11–24.
- Frazer, K., Ballinger, D., Cox, D., et al. 2007. International HapMap Consortium A second generation human haplotype map of over 3.1 million SNPs. *Nature* 449, 851–861.
- Gudbjartsson, D.F., Thorvaldsson, T., Kong, A., et al. 2005. Allegro version 2. *Nat. Genet.* 37, 1015–1016.
- Kruglyak, L., Daly, M.J., Reeve-Daly, M.P., et al. 1996. Parametric and nonparametric linkage analysis: a unified multipoint approach. *Am. J. Hum. Genet.* 58, 1347–1363.
- Lander, E.S., and Green, P. 1987. Construction of multilocus genetic linkage maps in humans. *Proc. Natl. Acad. Sci. USA* 84, 2363–2367.
- Li, J., and Jiang, T. 2003. Efficient inference of haplotypes from genotypes on a pedigree. *J. Bioinform. Comput. Biol.* 1, 41–69.
- Li, X., and Li, J. 2009. An almost linear time algorithm for a general haplotype solution on tree pedigrees with no recombination and its extensions. *J. Bioinform. Comput. Biol.* 7, 521–545.
- Li, X., Yin, X., and Li, J. 2010. Efficient identification of identical-by-descent status in pedigrees with many untyped individuals. *Bioinformatics* 26, i191.
- Morris, R., and Kaplan, N. 2002. On the advantage of haplotype analysis in the presence of multiple disease susceptibility alleles. *Genet Epidemiol.* 23, 221–233.
- Sobel, E., and Lange, K. 1996. Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker-sharing statistics. *Am. J. Hum. Genet.* 58, 1323–1337.
- Xiao, J., Liu, L., Xia, L., et al. 2007. Fast elimination of redundant linear equations and reconstruction of recombination-free mendelian inheritance on a pedigree. *Proc. 18th Annu. ACM-SIAM Symp. Discrete Alg.* 664.

Address correspondence to:

Jing Li

Department of Electrical Engineering and Computer Science

Case Western Reserve University

Cleveland, OH 44106

E-mail: jingli@case.edu