Journal of Bioinformatics and Computational Biology
Vol. 6, No. 1 (2008) 241–259
© Imperial College Press

Imperial College Press
www.icpress.co.uk

A SURVEY ON HAPLOTYPING ALGORITHMS FOR TIGHTLY LINKED MARKERS

JING LI*
Electrical Engineering and Computer Science Department
Case Western Reserve University
Cleveland, OH 44106, USA
jingli@case.edu

TAO JIANG
Department of Computer Science
University of California, Riverside
Riverside, CA 92521, USA
jiang@cs.ucr.edu

Received 28 March 2007
Revised 28 July 2007
Accepted 28 August 2007

Two grand challenges in the postgenomic era are to develop a detailed understanding of heritable variation in the human genome, and to develop robust strategies for identifying the genetic contribution to diseases and drug responses. Haplotypes of single nucleotide polymorphisms (SNPs) have been suggested as an effective representation of human variation, and various haplotype-based association mapping methods for complex traits have been proposed in the literature. However, humans are diploid and, in practice, genotype data instead of haplotype data are collected directly. Therefore, efficient and accurate computational methods for haplotype reconstruction are needed and have recently been investigated intensively, especially for tightly linked markers such as SNPs. This paper reviews statistical and combinatorial haplotyping algorithms using pedigree data, unrelated individuals, or pooled samples.

Keywords: Haplotype inference; SNP.

1. Introduction

With the completion of the Human Genome Project,[1,2] an almost complete human genomic DNA sequence has become available, which is essential to understanding the functions and characteristics of human genetic material. An important next step in human genomics is to determine the genetic variation among humans as well as the correlation between genetic variation and phenotypic variation such

[*] Corresponding author.

as disease status, quantitative traits, etc. To achieve this goal, an international collaboration — the International HapMap Project[3] — was launched in October 2002. The main objective of the HapMap Project is to identify the haplotype structure of humans and common haplotypes among populations. However, the human genome is a diploid and, in practice, haplotype data are not collected directly, especially in large-scale sequencing projects (mainly due to cost considerations); instead, genotype data are collected routinely in large sequencing projects. Hence, efficient and accurate computational methods and computer programs for the inference of haplotypes from genotypes are highly needed.

The existing computational methods for haplotyping fit into two broad categories: statistical methods and combinatorial (or rule-based) methods. Both methodologies can be applied to pedigree data, population data, or pooled samples. An earlier review paper[4] discussed haplotype inference on pedigree data, but the methods mentioned did not directly address the problem for tightly linked markers (i.e. single nucleotide polymorphisms or SNPs). Many developments have been made since then. There have been a few review papers on haplotype inference in recent years,[5–7] but all of them focus mainly on combinatorial formulations and solutions[a]; two of them, Gusfield[6] and Halldórsson *et al.*,[7] deal only with unrelated population data. This paper will review both statistical and combinatorial algorithms for three different types of data: pedigree data, population data, and pooled samples. It is organized as follows. A biological background of the problem will first be introduced in Sec. 1. Haplotype inference algorithms for three different types of input data will be discussed in three separate sections. A brief summary of genomic applications using haplotype information and possible future directions on haplotype inference will be presented in Sec. 5. Some commonly used haplotyping programs on the Internet will be listed at the end.

1.1. *Genetic background*

The genome of an organism consists of chromosomes, which are double-stranded DNAs. Locations on a chromosome can be identified using markers, which are small segments of DNA with some specific features (or a single nucleotide for SNP). The position of a marker on the chromosome is called a marker locus, and a marker state is called an allele. A set of markers and their positions define a genetic map of chromosomes.[9] There are many types of markers; the two most commonly used markers are microsatellite markers and SNP markers. Different sets of markers have different properties, such as the total number of different allelic states at one locus, frequency of each allele, distance between two adjacent loci, etc. A microsatellite marker usually has several different alleles at a locus (called multi-allelic); while an SNP marker can be treated as biallelic, which has two alternative states. There

[a] It was pointed out by a reviewer that a recent published survey[8] has discussed haplotype assembly and inference from both combinatorial and statistical points of view.

are millions of SNPs, but only hundreds of microsatellite markers, so the average distance between two SNPs is much smaller than the average distance between two microsatellite marker loci. By tightly linked markers, we mainly refer to SNPs. With advances in genotyping techniques, SNP markers are increasingly common in gene fine mapping and whole-genome association studies.

In diploid organisms, chromosomes come in pairs. The status of two alleles at a particular marker locus of a pair of chromosomes is called a marker genotype. The genotype information of a locus is denoted using a set (a, b), where a and b are integers representing allele identifiers (IDs). For example, for biallelic markers like SNPs, $a, b \in \{1, 2\}$. If the two alleles are the same, the genotype is homozygous; otherwise, it is heterozygous. A haplotype consists of all alleles, one from each locus, that are on the same chromosome. Figure 1 illustrates the above concepts.

The Mendelian law of inheritance states that the genotype of a child must come from the genotypes of its parents at each marker locus. In other words, the two alleles at each locus of a child have different origins: one is from the father (paternal allele), and the other from the mother (maternal allele). Such information is also called the phase of the two alleles, which cannot be obtained directly from genotypes. Usually, for a tightly linked region, a child inherits a complete haplotype from each parent. However, recombination may occur, where the two haplotypes of a parent get shuffled due to a crossover of chromosomes and one of the shuffled copies is passed on to the child; such an event is called a recombination event, and its result is called a recombinant. Figure 2 illustrates an example in which the paternal haplotype of member 3 is the result of a recombinant.

Mathematically, the genotypes of an individual for a given region with m loci can be represented by an m-dimensional vector g, where each of its elements is a pair of alleles. A haplotype is simply a vector of alleles. The genotype vector is actually composed of a maternal haplotype (h_{m}) and a paternal haplotype (h_{p}), i.e. $g = (h_{\mathrm{m}}, h_{\mathrm{p}})$; however, such information is lost when obtaining an individual's genotypes due to the limitation of the current genotyping techniques. The goal of haplotype inference is to reconstruct a haplotype pair based on constraints imposed by genotypes of family members or some mathematical models. It is easy to see that, without further constraints, an individual with k heterozygous loci will have $2^{(k-1)}$

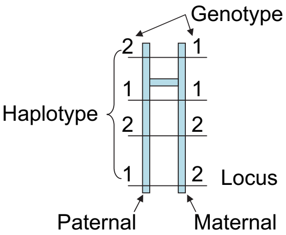


Fig. 1. The structure of a pair of chromosomes from a mathematical point of view.

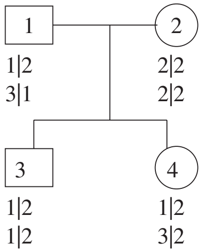


Fig. 2. An example of a recombination event. A family with two parents (members 1 and 2) and two children (members 3 and 4) is represented by a pedigree graph. The data consist of two loci, with the genotype/haplotype information listed below each member. The notation $a|b$ means that the phase information at the locus has been resolved, and we know that allele a is from the father and allele b is from the mother.

different haplotype pairs which are consistent with its genotypes. For example, for $g = \langle(1,2),(1,2)\rangle$, both haplotype pair $h_1 = \langle 1,1\rangle$, $h_2 = \langle 2,2\rangle$ and haplotype pair $h_1 = \langle 1,2\rangle$, $h_2 = \langle 2,1\rangle$ are consistent with g. For a pair of haplotypes h_1, h_2 consistent with a genotype g, we write $g = h_1 \oplus h_2$.

2. Haplotype Inference from Pedigree Data

A pedigree is an extended family where individuals are related by a parenthood relation. A pedigree can be naturally modeled using a directed acyclic graph, with nodes representing individuals and edges representing parent–child relationships. In addition, the genotype of each member in a pedigree is given with possibly some missing alleles. It is generally assumed that pedigree data consist of no mutations, thus the genotypes are consistent with the Mendelian law of inheritance; this is realistic in practice, given the moderate sizes of most available human pedigrees. With genotype information from parents, the phase of a child at a particular locus may be determined in many cases, but there are other cases where the phase of a child cannot be determined (e.g. when both parents and a child have the same heterozygous genotypes). Missing data further complicate the situation and increase the total freedom in a pedigree.

2.1. *Maximum likelihood approach*

The maximum likelihood (ML) principle can be naturally applied here. Given a pedigree with genotype information for each member, with possibly missing data, the goal of the ML approach is to identify the most likely haplotype pair for each individual. For each haplotype assignment of a pedigree, the calculation of the probability involves two terms: the first term is the founder probability, and the second one is the transmission probability. More specifically, let H denote a consistent haplotype assignment of the pedigree. For individual i in the pedigree, let $({}_ih_{\mathrm{m}}, {}_ih_{\mathrm{p}})$

denote its haplotype pair. Then,

$$Pr(H) = \prod_{i} Pr({}_ih_{\mathrm{m}}, {}_ih_{\mathrm{p}}) \prod_{t,j,k} Pr({}_th_{\mathrm{m}}|{}_jh_{\mathrm{m}}, {}_jh_{\mathrm{p}}) Pr({}_th_{\mathrm{p}}|{}_kh_{\mathrm{m}}, {}_kh_{\mathrm{p}}), \tag{1}$$

where the product on i ranges over all founders, and the product on $\{t, j, k\}$ ranges over all offspring-mother-father trios. Under Hardy–Weinberg equilibrium and linkage equilibrium assumptions, the founder probability can be calculated from population frequencies of alleles, i.e. $Pr({}_ih_{\mathrm{m}}, {}_ih_{\mathrm{p}}) = p({}_ih_{\mathrm{m}})p({}_ih_{\mathrm{p}})$, where $p({}_ih_{\mathrm{m}})$ and $p({}_ih_{\mathrm{p}})$ are haplotype frequencies which are products of their allele frequencies. The gamete transmission probabilities $Pr({}_th_{\mathrm{m}}|{}_jh_{\mathrm{m}}, {}_jh_{\mathrm{p}})$ and $Pr({}_th_m|{}_jh_{\mathrm{m}}, {}_jh_{\mathrm{p}})$ can be calculated based on recombination fractions between marker intervals. Notice that for tightly linked markers like SNPs, the probability of recombination events between two adjacent markers is extremely small, so the ML approach is in favor of haplotype configurations with few recombinations.

In traditional linkage analysis, the likelihood calculation involves the summation over all possible haplotype assignments. So, theoretically, one can simultaneously output the haplotype assignment with maximum likelihood when performing linkage analysis. Two exact algorithms have been proposed to calculate the probability of a pedigree for linkage analysis. The Elston–Stewart algorithm[10] takes advantage of the Markov property based on pedigree structure: given parents' genotype information, the genotypes of a child are independent from the genotypes of its ancestors. The algorithm is linear in pedigree sizes, but exponential in the number of genetic loci. The Lander–Green algorithm[11] takes advantage of the Markov property based on marker loci: under the assumption of no interference, the phase of a marker only depends on the phase of its previous locus. This algorithm is linear in the number of genetic loci, but exponential in pedigree sizes. Although much improvements[12–14] have been made for both algorithms, the exact calculations of likelihood and haplotype inference for complex pedigrees with substantial missing data are still computationally infeasible. Even approximation algorithms employing important sampling techniques such as simulated annealing[4] are not efficient enough for complex pedigrees with large sizes.

The calculation of the likelihood of a pedigree, as well as the calculation of haplotype configurations, in most available tools for linkage analysis (for example, GeneHunter, SimWalk2, and S.A.G.E.; the links to these programs are provided in Sec. 5) assumes linkage equilibrium between markers. The assumption is unrealistic for tightly linked markers such as SNPs because it is well known that most SNP markers are in linkage disequilibrium (LD). The effect of violation of such an assumption has only been investigated very recently. Abecasis and Wigginton[15] have proposed a new approach that can directly model LD between markers during multipoint analysis of human pedigrees. The algorithm first partitions all of the SNPs into clusters based on their LD. For each cluster, the authors assume that there is no recombination and use haplotypes to incorporate LD within a cluster. LD between clusters is ignored and the likelihood can be calculated using the

Lander–Green algorithm, while taking each cluster essentially as a marker. Their simulation results show that the approach resolves previously described biases in multipoint linkage analysis with SNPs which are in LD. Therefore, although it is a natural formulation, the ML approach is not very suitable for SNP data. This leads to a discrete formulation to minimize the total number of recombination events in each pedigree, as will be discussed below.

2.2. *Recombination/Crossover minimization*

Given the fact that ML-based approaches are usually time-consuming and the assumptions that they require do not always hold for tightly linked markers, rule-based approaches to minimize recombination/crossover in a pedigree have recently received much attention. The minimum recombination principle basically states that genetic recombination is rare, thus haplotypes with fewer recombinants should be preferred in a haplotype reconstruction.[16–18] For tightly linked markers such as SNPs, the principle is well supported by experimental data. For example, recently published results[19–21] demonstrate that, in the case of human, the number of distinct haplotypes is very limited relative to the number of all possible haplotype combinations. Moreover, the genomic DNA can probably be partitioned into long blocks such that recombination within each block is rare or even nonexistent during the history. For pedigree data, one can safely assume recombination is rare for a much larger region. In the literature, the crossover minimization formulation is also called the minimum-recombinant haplotype configuration (MRHC) problem.[17,22,23]

2.2.1. *Algorithms for MRHC*

Sobel *et al.*[4] proposed a simulated annealing algorithm while taking a pseudo-likelihood function, regarding the number of recombinants as the energy function. O'Connell[16] worked on an important special case of the MRHC problem; he assumed that the input data contain haplotype solutions with zero recombinants, and the goal of his program was to find all such solutions. This special case is called the zero-recombinant haplotype configuration (ZRHC) problem, and will be discussed separately because of its importance for tightly linked markers. Tapadar *et al.*[18] utilized the genetic algorithm to attack the same problem. Qian and Beckmann[17] proposed a rule-based algorithm to reconstruct haplotype configurations for pedigree data, based on local minimization of each nuclear family. Although their program MRH performs well for small pedigrees and achieves better results than some previous algorithms,[16,18] its effectiveness scales poorly, especially for data with biallelic markers.

In a series of papers,[22–26] the authors proposed several algorithms based on different assumptions about the real data. They developed an iterative heuristic algorithm, called block extension, for MRHC that is much more efficient than MRH. The experiments showed that the block-extension algorithm can compute an optimal solution or nearly optimal solution when the minimum number of recombinants required is small[22,23]; however, its performance deteriorates significantly when the input data require more (e.g. four or more) recombinants. For pedigrees with small sizes or pedigrees with a small number of markers, they developed two dynamic programming (DP) algorithms.[24] The running time for the first DP algorithm is linear in the size of a pedigree and the time for the second one is linear in the number of markers, which resemble the Elston–Stewart algorithm and the Lander–Green algorithm for the statistical analysis, respectively. For the most general case of the problem, the authors designed an effective integer linear programming (ILP) formulation of MRHC. It integrates missing data imputation and haplotype inference together, and employs a branch-and-bound strategy that utilizes a partial order relationship and some other special relationships among variables to decide the branching order; the partial order relationship is discovered in the preprocessing of constraints by taking advantage of some special properties in the ILP formulation. A directed graph is built based on the variables and their partial order relationship. By identifying and collapsing strongly connected components in the graph, the algorithm can greatly reduce the size of an ILP instance. Nontrivial lower and upper bounds on the optimal number of recombinants are estimated at each branching node to prune the branch-and-bound search tree. When multiple solutions exist, a best haplotype configuration is selected based on an ML approach. The algorithm also incorporates the marker interval distance into the formulation whenever it is known, thus overcoming the inadequacy of many rule-based algorithms which ignore the important information. The test results on simulated data demonstrate that the algorithm is very efficient in practice. A comparison of the algorithm with a well-known statistical approach, SimWalk2,[4] on simulated data with evenly and unevenly spaced markers also demonstrates the effectiveness and soundness of the ILP algorithm.[26]

2.2.2. *Algorithms for ZRHC*

For the zero-recombinant haplotype configuration (ZRHC) problem, the goal is to enumerate all haplotype solutions that require no recombinant if such solutions exist. It was first introduced in O'Connell.[16] The formulation seems under a more stringent biological assumption, but it is actually more practical for tightly linked markers such as SNP data. An efficient algorithm for ZRHC could also be useful for solving the general MRHC problem as a subroutine, when the number of recombinants is expected to be small. Note that recent work on haplotype inference for population data based on perfect phylogenies also assumes that the data are recombination-free.[26,27] When the solution for ZRHC is not unique, it would really be useful to be able to enumerate all of the solutions instead of finding only one feasible solution, so that the solutions can be examined in subsequent analysis (e.g. likelihood distribution of haplotypes,[25,26] linkage between different haplotype blocks, etc.) by geneticists.

O'Connell[16] presented an exponential-time algorithm for ZRHC based on exhaustive enumeration. It works by eliminating all impossible genotypes. Zhang *et al.*[27] developed a program for ZRHC that combines logic rules and the expectation-maximization (EM) algorithm. Li and Jiang[22] introduced an $O(m^3n^3)$ time algorithm by formulating ZRHC as a system of $O(mn)$ linear equations, with mn variables over the finite field of $F(2)$ and applied Gaussian elimination. Although this cubic-time algorithm is reasonably fast, it is inadequate for large-scale pedigree analysis where both m and n can be in the order of tens or even hundreds, and we may have to examine many pedigrees and haplotype blocks; there are, for example, over five million SNP markers in the public database dbSNP.

This challenge has motivated recent efforts of searching for more efficient algorithms for ZRHC. Several attempts have been made by Chin and Zhang[28] and Li *et al.*,[29] but the authors failed to prove the correctness of their algorithms in all cases, especially when the input pedigree has mating loops. Chan *et al.*[30] proposed a linear-time algorithm, but the algorithm only works for pedigrees without mating loops (i.e. the tree pedigrees). Xiao *et al.*[31] very recently presented a much faster algorithm for ZRHC with running time $O(mn^2+n^3\log^2 n\log\log n)$. Their construction begins with a new system of linear equations over $F(2)$. Although the system still has $O(mn)$ variables and $O(mn)$ equations, it can be reduced to an effectively equivalent system with $O(mn)$ equations and at most $2n$ variables by exploring the underlying pedigree graph structure. By using standard Gaussian elimination, this implies an improved algorithm for ZRHC with running time $O(mn^3)$. However, the authors were able to reduce the number of equations further to $O(n\log^2 n\log\log n)$ (assuming that $m \geq \log^2 n\log\log n$, which usually holds in practice) by giving an $O(mn)$ time method for eliminating redundant equations in the system. Although such a fast elimination method is not known for general systems of linear equations, it was achieved by again taking advantage of the underlying pedigree graph structure and recent progress on low-stretch spanning trees in Elkin *et al.*[32] The algorithm actually runs in $O(mn^2+n^3)$ time when the input pedigree is a tree pedigree with no mating loops (which is often true for human pedigrees), or when there is a locus that is heterozygous across the entire pedigree. Moreover, the algorithm produces a general solution[b] to the original system of linear equations at the end that represents all feasible solutions to the ZRHC problem.

2.2.3. *Complexity results*

It has been shown that MRHC is computational-hard.[23,24] Further properties about the complexity and approximability of MRHC have been recently studied by Liu *et al.*[33,34] It was shown that the MRHC for simple pedigrees, where each member

[b] A general solution of any linear system is denoted by the span of a basis in the solution space to its associated homogeneous system, offset from the origin by a vector, i.e. by any particular solution.

has at most one mate and at most one child (i.e. binary-tree pedigrees), is NP-hard; and that the MRHC on two-locus pedigrees or binary-tree pedigrees with missing data cannot be approximated unless P = NP. The authors also proved that the MRHC on two-locus pedigrees without missing data cannot be approximated within any constant ratio under the Unique Games Conjecture, but can be approximated within the ratio $O(\sqrt{\log n})$; moreover, they showed that the MRHC for tree pedigrees without missing data cannot be approximated within any constant ratio under the Unique Games Conjecture, too. Some hardness and approximation results are also given in Liu *et al.*[33,34] for the MRHC on pedigrees where each member has a bounded number of children and mates, as is often the case in real pedigrees.

3. Haplotype Inference from Population Data

In this section, we consider the haplotype inference (HI) problem based on population data, i.e. unrelated individuals. In addition to the haplotype assignment of each individual, we are also interested in the estimation of population haplotype frequencies. These two problems are related and are usually solved simultaneously. Without constraints from family members, an individual with m heterozygous loci has 2^{m-1} consistent haplotype pairs. Furthermore, the HI problem from population data is meaningful only for tightly linked markers, where correlations among markers exist. Still, one cannot directly infer which pair is more likely to be the true haplotypes for a given individual. Instead, genetic models based on the evolution of human population history have to be adopted, directly or indirectly, to define the extent of "optimality" of a particular consistent solution for given samples.

The HI problem from unrelated individuals was first addressed by Clark[35] in 1990, and a rule-based algorithm was proposed in his paper. We will start our discussion from this simple but widely used algorithm. Gusfield investigated mathematical properties of Clark's algorithm[36]; and later proposed a discrete model called the perfect phylogeny haplotype (PPH) problem, which implicitly adopts the coalescent model with no recombination.[37] Another commonly used discrete model in the literature is called the pure-parsimony approach,[38–40] which intends to find a solution with the smallest number of distinct haplotypes. In addition to discrete models, statistical approaches have also been applied on the HI problem. Two different groups[41,42] have proposed an ML approach and employed the EM algorithm to find a haplotype solution. Bayesian approaches[43,44] have also been applied on the HI problem by incorporating an informative prior based on population genetics models.

3.1. *Clark's algorithm*

Observe that for a set of m markers, if an individual is homozygous at all loci or if it has only one heterozygous locus, the haplotype pair of the individual is trivially determined since there is only one consistent pair of haplotypes. If there is at least one such individual, one can obtain an initial set of haplotypes H from the input data. For each genotype vector g with more than one heterozygous locus, Clark's idea was to use a haplotype h_1 in H that is consistent with g to obtain the haplotype pair for g, i.e. $g = h_1 \oplus h_2$, and include h_2 in H; the algorithm will then iterate until no more genotypes can be resolved. It is possible that the algorithm cannot even start if there are no individuals with 0 or 1 heterozygous locus. The algorithm also does not guarantee that every genotype will eventually be resolved. Different sequences of application of Clark's rule might give different results. Gusfield[36] investigated mathematical properties of Clark's algorithm, and proved that finding the sequence(s) of application of Clark's rule with a minimum number of unresolved genotypes is NP-hard.

3.2. *Perfect phylogeny model*

Later, Gusfield[37] introduced a perfect phylogeny model for the HI problem, based on two assumptions. First, the model assumes that, for a set of tightly linked SNPs, historical recombination events do not exist; experimental results and population genetics models generally support this assumption. Second, the model adopts the standard assumption of infinite sites in population genetics, which basically means that, at each SNP site, mutation can only occur at most once. Under these two assumptions, the $2n$ haplotypes from n individuals can be organized into a rooted tree called perfect phylogeny. Each leaf of the tree represents a haplotype. Each interior edge is labeled by at least one SNP, and each SNP labels exactly one edge. A path from the root to a leaf spells all of the mutant sites of the haplotype at the leaf from the ancestral haplotype at the root (usually not given). The PPH problem finds, given a set of genotypes, a set of haplotypes that admits a perfect phylogeny. Gusfield[37] presented an algorithm by reducing the problem to a graph realization problem with an almost linear running time in theory, but the implementation of the algorithm is too complex to be practical. Since then, a couple of algorithms have been proposed. Two groups[45,46] have independently proposed two algorithms with the same running time $O(nm^2)$, where n is the number of individuals and m is the number of SNPs. More recently, Ding *et al.*[47] presented a linear algorithm for the problem, and the algorithm has been implemented in a program called LPPH. For future directions, it is desirable to extend the perfect phylogeny model to allow recombination and missing data.

3.3. *Pure parsimony*

The pure parsimony approach has also been investigated by researchers[38–40] in the computational biology community. Under this criteria, the goal is to find a minimum set of distinct haplotypes that can resolve all of the given genotypes. The rationale of the parsimony principle for the HI problem is also based on the same observation that, in human populations, the number of observed distinct haplotypes is far smaller than the total number of all possible haplotypes. Unlike the perfect phylogeny model, which has an optimal linear time algorithm, the computation of the diversity minimization problem turns out to be hard. It has been shown[40] that, in theory, the problem not only has no practical exact algorithms, but it also has no practical approximation algorithms. Gusfield[38] formulated the problem using the integer linear programming approach, which can find optimal solutions for instances with small sizes. Wang and Xu[39] proposed a branch-and-bound algorithm, and experimental results have shown it to be effective for practical problems.

All three combinatorial formulations for HI using population data have been reviewed in detail by Gusfield.[6] In addition to discrete approaches, statistical models have been studied in the literature. We will introduce the ML model and Bayesian approaches in the next two subsections.

3.4. *Maximum likelihood*

The ML approach[41,42] takes haplotype population frequencies as unknown parameters which need to be inferred. The goal is to estimate values of haplotype frequencies that maximize the probability of observing the given genotype data. Assume that all of the individuals are independent; then, the likelihood of the data is just the multiplication of the probability of each individual. Under the assumption of random mating and Hardy–Weinberg equilibrium, the probability of observing a particular genotype from an individual is the summation of the product of two haplotype frequencies for all haplotype pairs that are consistent with the genotype:

$$L(G) = \prod_{i}^{n} \sum_{h_s \oplus h_t = g_i} p(h_s)p(h_t). \tag{2}$$

When the maximum likelihood estimates (MLEs) cannot be readily obtained from an analytical derivation like in this case, numerical methods are commonly used. A widely accepted approach to obtain the MLEs is the EM algorithm. The EM algorithm is an iterative method that consists of two steps (E-step and M-step) in each iteration. In the context of haplotype inference, it takes the haplotype frequencies as parameters and the phase of each individual as missing data. If the phase of each individual is known, the MLE of the frequency of a particular haplotype is just the fraction of that haplotype occurring in the samples; on the other hand, if haplotype frequencies are known, the probability of observing an individual with a phased haplotype pair is just the product of the frequencies of the two haplotypes under the assumption that haplotypes are in Hardy–Weinberg equilibrium.

The EM algorithm starts with an initial (probably arbitrary) assignment of haplotype frequencies, $p^0(h_1), p^0(h_2), \ldots, p^0(h_k)$. In the E-step of the ith iteration, it calculates the expected counts ($n^i_{h_s}$) of a haplotype h_s from samples, assuming that the haplotype frequencies are true values:

$$n^i_{h_s} = \sum_{g: g = h_s \oplus h_t} \frac{p^i(h_s)p^i(h_t)}{\sum_{h_u, h_v : h_u \oplus h_v = g} p^i(h_u)p^i(h_v)}, \tag{3}$$

where the first summation on the right-hand side is over all individuals whose genotypes are consistent with h_s, and the second summation is over all consistent haplotype pairs for a particular individual with genotype g. In the M-step, the haplotype frequencies are updated based on the expected counts $n^i_{h_s}$:

$$p^{i+1}(h_s) = \frac{n^i_{h_s}}{2n}. \tag{4}$$

The algorithm iterates until it converges or it reaches the maximum number of iterations allowed. To estimate the haplotype pair of each individual, one can pick the pair with the largest probability based on the estimated haplotype frequencies. Theoretically, the EM algorithm is guaranteed to converge to a (local) maximum in linear time, but the number of variables in this case (i.e. haplotype frequencies) can be exponentially large with respect to the number of loci in the region. So, a direct implementation of the EM algorithm for the HI problem usually cannot deal with data of more than 25 loci. It is also a known fact that the EM algorithm might converge to a local optimal point instead of a global one. Users are recommended to start with different initial values, and pick the solution with the ML probability. Furthermore, the EM algorithm cannot provide the estimates of variances of the MLEs in general, unless the number of loci is small.

3.5. *Bayesian approaches*

Unlike the ML method, where parameters are unknown points in a parameter space, Bayesianists treat parameters as random variables. The goal of Bayesian inference is to estimate the posterior distribution of parameters given data we observed, assuming some known prior knowledge about parameters before seeing data. Point estimations can be obtained by taking expectations of the posterior distribution. Let $Pr(p(H))$ denote the prior distribution of haplotype frequencies, and $Pr(p(H)|G)$ denote the posterior distribution of haplotype frequencies given genotype data G. The posterior distribution can be calculated via Bayes' theorem:

$$Pr(p(H)|G) = \frac{Pr(G|p(H))Pr(p(H))}{Pr(G)}. \tag{5}$$

The prior probability $Pr(p(H))$ is assumed to be known, and the probability of data given a particular set of parameters $Pr(G|p(H))$ is easy to calculate. While the calculation of the overall probability $Pr(G)$ involves multidimensional integrations or a summation over an exponentially large number of terms, it is infeasible in many cases. Important sampling techniques such as Markov chain Monte Carlo (MCMC) are commonly used in such situations.

Two Bayesian approaches[43,44] have been proposed for HI from population data, both of which use Gibbs sampling techniques to obtain an estimation of posterior distribution of haplotype frequencies. The algorithm of Stephens *et al.*[43] starts from an arbitrary haplotype solution of the given genotypes, and iteratively updates a randomly selected individual assuming that all of the other individuals have their correct haplotype assignments. The algorithm of Niu *et al.*[44] starts from an initial assignment of haplotype frequencies; at each iteration step, it first samples a pair of compatible haplotypes for each individual, and then updates the haplotype frequencies based on the haplotype solution of each individual. The two methods differ mainly from the prior distributions they assume[48]: Stephens *et al.*[43] used a prior approximating the coalescent model, while Niu *et al.*[44] used the Dirichlet prior. Under the coalescent model, haplotypes to be sampled will tend to be more similar to previously sampled haplotypes, a property that has been used in Clark's algorithm. Experiments[48] have shown that estimations based on the coalescent model are more accurate than those based on the Dirichlet prior. Both algorithms have been implemented into computer programs (i.e. Phase and Haplotyper) that have been widely used.

An important contribution in Niu *et al.*[44] is the introduction of the partition–ligation technique, which is an application of the divide-conquer technique that can reduce the computational burden for large data sets. The same idea has been incorporated into other algorithms such as Phase V2.0 and the EM algorithm.[49]

4. Haplotype Inference from Pooled Samples

As a strategy of reducing the genotyping cost, pooling individual samples has been shown to be efficient in estimating population allele frequencies and long distance (LD) coefficients.[50] For HI, it is obvious that pooling samples adds more ambiguity. Nevertheless, several groups[51,52] have investigated the efficiency and cost-effectiveness of estimating haplotype frequencies from pooled DNA data. In general, suppose $K \geq 1$ independent individuals are pooled together, where $K = 1$ corresponds to the strategy with no pooling. The genotype at each locus can be represented by the number of allele 1, i.e. an integer g such that $0 \leq g \leq 2K$. The primary goal is to estimate haplotype frequencies for the region of interest. Although the most likely haplotype configurations (the $2K$ haplotypes in each pool that are consistent with the input genotypes) might be inferred, haplotypes for each individual cannot be constructed in general. Because the accuracy of haplotype frequency estimates decreases with an increase of K, it is not worthwhile to pool a large number of samples if the accuracy deteriorates too much. To compare the cost-effectiveness of different strategies, Yang *et al.*[52] defined a simple measure named relative efficiency: $R(K) = K \times v_1 / v_K$, where v_1 and v_K are the mean squared errors for samples without pooling and with pooling of a size K, respectively. Pooling is only meaningful when $R(K) \geq 1$.

To obtain haplotype frequency estimates from pooled samples, in theory, both the ML approach and Bayesian-based approaches can be applied, with the haplotype pair of an individual replaced by the haplotype configuration of a pool. As a matter of fact, Quade *et al.*[53] proposed an algorithm that views pedigree data and population data as special cases of pooled samples. Yang *et al.*[52] also applied the ML method on pooled data, and adopted the EM algorithm for the estimation of haplotype frequencies. The algorithm iteratively updates population haplotype frequencies and haplotype configurations of each pool. Because the number of distinct haplotypes increases exponentially with the number of loci m, and the number of distinct haplotype configurations in each pool increases exponentially with the size of each pool K, the algorithm is only practical for problems with small sizes ($m \leq 15$ and $K \leq 6$). In terms of cost-effectiveness, simulation results in Yang *et al.*[52] showed that the relative efficiency $R(K)$ increases the most when $K = 2$ or 3, and the pooling strategy is more effective for SNPs with high LD and for SNPs with moderate or large minor allele frequencies. In addition to the loss of genotype and haplotype information for each individual, one other limitation of pooling strategies in genome association studies is the loss of phenotype information, especially when multiple measures of different quantitative traits have been recorded for each individual.

5. Discussion

Haplotype information can be used in many applications in biomedical research. For example, because nearby SNPs have high correlations, only a subset of tag SNPs is needed to approximate all common SNPs in genome-wide association studies.[54] A smaller number of SNPs corresponds to a smaller number of tests, which usually means higher power for association analysis.[54] Therefore, tag SNP selection is an important task for genome-wide association studies, and the selection of tag SNPs is usually based on haplotype structures and haplotype frequencies.[55,56] Furthermore, it has been shown that haplotype-based methods may provide higher power than single SNP-based methods under certain conditions, and various methods have been proposed to directly use haplotype information in disease gene association mapping.[57–60] Accurate haplotype estimations are essential for the success of such methods.

Many models and algorithms for haplotype reconstruction and haplotype frequency estimation have been discussed in this review. Different formulations have adopted different assumptions, and different programs have different time complexities; this makes a fair comparison of all programs a difficult task. Attention should be paid in choosing appropriate programs for each specific dataset. In general, haplotype inference is still a computation-intensive problem, especially when the data consist of substantial missing alleles. Faster and more accurate algorithms for each model are still in great need. We believe that one particular type of data has not been adequately addressed in recent studies. In biomedical research, it is common to collect multiple individuals from each family; examples of such designs include parent-child trios, affected siblings, and parent-child pairs. Constraints from relatives can greatly reduce the number of possible haplotype pairs of each individual, as well as ambiguities caused by missing alleles. Existing approaches usually only apply the Mendelian principles whenever possible to infer phase information at each individual locus, and then select only one individual from each family. Actually, many more constraints can be explored, and much useful information has been discarded by including only one individual from each family. Therefore, new algorithms designed specifically for such data have a great potential to be more efficient and more accurate than existing ones.

6. Software Available on the Web

Table 1. Commonly used software tools.

Name	Comments	Website
SimWalk2[4]	Pedigree	http://watson.hgen.pitt.edu/ docs/simwalk2.html/
Merlin[15]	Pedigree, linkage, statistics, incorporating LD	http://www.sph.umich.edu/ csg/abecasis/Merlin/
S.A.G.E.[61]	Pedigree, linkage, EM	http://darwin.case.edu/
GeneHunter[12]	Pedigree, linkage, Lander–Green	http://www.fhcrc.org/science/labs/ kruglyak/Downloads/index.html/
PedPhase[26]	Pedigree, MRHC/ZRHC	http://www.eecs.case.edu/ jxl175/haplotyping.html/
HAPLORE[27]	Pedigree, ZRHC	http://www.soph.uab.edu/Statgenetics/ People/KZhang/HAPLORE/index.html/
Phase[43]	Population, Bayesian coalescent prior	http://www.stat.washington.edu/ stephens/software.html/
Haplotyper[44]	Population, Bayesian Dirichlet prior	http://www.people.fas.harvard.edu/ junliu/Haplo/docMain.htm/
PPH and LPPH[6]	Population, PPH	http://wwwcsif.cs.ucdavis.edu/gusfield/

Acknowledgments

This work was supported in part by NIH R01 LM008991 and a start-up fund from Case Western Reserve University to J. Li. The authors also thank two anonymous reviewers for their constructive suggestions.

References

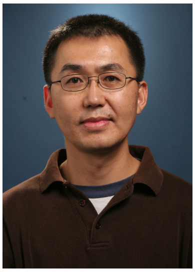
1. International Human Genome Sequencing Consortium, Initial sequencing and analysis of the human genome, *Nature* **409**(6822):860–921, 2001.
2. Venter JC *et al.*, The sequence of the human genome, *Science* **291**(5507):1304–1351, 2001.
3. The International HapMap Consortium, The International HapMap Project, *Nature* **426**:789–796, 2003.
4. Sobel E, Lange K, O'Connell J, Weeks D, Haplotyping algorithms, in Speed T, Waterman M (eds.), *Genetic Mapping and DNA Sequencing*, IMA Volumes in Mathematics and Its Applications, Vol. 81, Springer-Verlag, New York, pp. 89–110, 1996.
5. Bonizzoni P, Della Vedova G, Dondi R, Li J, The haplotyping problem: An overview of computational models and solutions, *J Comput Sci Technol* **18**(6):675–688, 2003.

6. Gusfield D, An overview of combinatorial methods for haplotype inference, in Istrail S, Waterman M, Clark A (eds.), *Computational Methods for SNPs and Haplotype Inference*, Lecture Notes in Computer Science, Vol. 2983, Springer-Verlag, Berlin, pp. 9–25, 2004.
7. Halldórsson BV, Bafna V, Edwards N, Lippert R, Yooseph S, Istrail S, A survey of computational methods for determining haplotypes, in Istrail S, Waterman M, Clark A (eds.), *Computational Methods for SNPs and Haplotype Inference*, Lecture Notes in Computer Science, Vol. 2983, Springer-Verlag, Berlin, pp. 26–47, 2004.
8. Zhang XS, Wang RS, Wu LY, Chen L, Models and algorithms for haplotyping problem, *Curr Bioinformatics* **1**(1):105–114, 2006.
9. Sham P, *Statistics in Human Genetics*, Oxford University Press, New York, 1998.
10. Elston RC, Stewart J, A general model for the genetic analysis of pedigree data, *Hum Hered* **21**:523–542, 1971.
11. Lander ES, Green P, Construction of multilocus genetic linkage maps in humans, *Proc Natl Acad Sci USA* **84**:2363–2367, 1987.
12. Kruglyak L, Daly MJ, Reeve-Daly MP, Lander ES, Parametric and nonparametric linkage analysis: A unified multipoint approach, *Am J Hum Genet* **58**:1347–1363, 1996.
13. Gudbjartsson DF, Jonasson K, Frigge ML, Kong A, Allegro, a new computer program for multipoint linkage analysis, *Nat Genet* **25**(1):12–13, 2000.
14. Abecasis GR, Cherny SS, Cookson WO, Cardon LR, Merlin — Rapid analysis of dense genetic maps using sparse gene flow trees, *Nat Genet* **30**(1):97–101, 2002.
15. Abecasis GR, Wigginton JE, Handling marker–marker linkage disequilibrium: Pedigree analysis with clustered markers, *Am J Hum Genet* **77**:754–767, 2005.
16. O'Connell JR, Zero-recombinant haplotyping: Applications to fine mapping using SNPs, *Genet Epidemiol* **19**(Suppl 1):S64–S70, 2000.
17. Qian D, Beckmann L, Minimum-recombinant haplotyping in pedigrees, *Am J Hum Genet* **70**(6):1434–1445, 2002.
18. Tapadar P, Ghosh S, Majumder PP, Haplotyping in pedigrees via a genetic algorithm, *Hum Hered* **50**(1):43–56, 2000.
19. Daly MJ, Rioux JD, Schaffner SF, Hudson TJ, Lander ES, High-resolution haplotype structure in the human genome, *Nat Genet* **29**(2):229–232, 2001.
20. Gabriel SB *et al.*, The structure of haplotype blocks in the human genome, *Science* **296**(5576):2225–2229, 2002.
21. Helmuth L, Genome research: Map of the human genome 3.0, *Science* **293**(5530):583–585, 2001.
22. Li J, Jiang T, Efficient rule-based haplotyping algorithms for pedigree data, in *Proc RECOMB'03*, pp. 197–206, 2003.
23. Li J, Jiang T, Efficient inference of haplotypes from genotypes on a pedigree, *J Bioinform Comput Biol* **1**(1):41–69, 2003.
24. Doi K, Li J, Jiang T, Minimum recombinant haplotype configuration on pedigrees without mating loops, *Proc Workshop on Algorithms in Bioinformatics (WABI)*, Budapest, Hungary, pp. 339–353, 2003.
25. Li J, Jiang T, An exact solution for finding minimum recombinant haplotype configurations on pedigrees with missing data by integer linear programming, in *Proc RECOMB'04*, pp. 101–110, 2004.
26. Li J, Jiang T, Computing the minimum recombinant haplotype configuration from incomplete genotype data on a pedigree by integer linear programming, *J Comput Biol* **12**:719–739, 2005.

27. Zhang K, Sun F, Zhao H, HAPLORE: A program for haplotype reconstruction in general pedigrees without recombination, *Bioinformatics* **21**:90–103, 2005.
28. Chin F, Zhang Q, Haplotype inference on tightly linked markers in pedigree data, unpublished manuscript, 2005.
29. Li X, Chen Y, Li J, An efficient algorithm for the zero-recombinant haplotype configuration problem, unpublished manuscript, 2006.
30. Chan MY, Chan W, Chin F, Fung S, Kao M, Linear-time haplotype inference on pedigrees without recombinations, *Proc of the 6th Annual Workshop on Algorithms in Bioinformatics (WABI06)*, pp. 56–67, 2006.
31. Xiao J, Liu L, Xia L, Jiang T, Fast elimination of redundant linear equations and reconstruction of recombination-free Mendelian inheritance on a pedigree, *Proc 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, New Orleans, LA, pp. 655–664, 2007.
32. Elkin M, Emeky Y, Spielman DA, Teng S, Lower-stretch spanning trees, *Proc 37th ACM Symposium on Theory of Computing (STOC'05)*, pp. 494–503, 2005.
33. Liu L, Chen X, Xiao J, Jiang T, Complexity and approximation of the minimum recombination haplotype configuration problem, *Proc 16th International Symposium on Algorithms and Computation (ISAAC'05)*, pp. 370–379, 2005.
34. Liu L, Chen X, Xiao J, Jiang T, Complexity and approximation of the minimum recombination haplotype configuration problem, *Theor Comput Sci* **378**:316–330, 2007.
35. Clark AG, Inference of haplotypes from PCR-amplified samples of diploid populations, *Mol Biol Evol* **7**(2):111–122, 1990.
36. Gusfield D, Inference of haplotypes from samples of diploid populations: Complexity and algorithms, *J Comput Biol* **8**:305–323, 2001.
37. Gusfield D, Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions, in *Proc RECOMB'02*, pp. 166–175, 2002.
38. Gusfield D, Haplotype inference by pure parsimony, in *Proc Combinatorial Pattern Matching Conference'03*, pp. 144–155, 2003.
39. Wang L, Xu L, Haplotype inference by pure parsimony, *Bioinformatics* **19**:1773–1780, 2003.
40. Lancia G, Pinotti MC, Rizzi R, Haplotyping populations by pure parsimony, *INFORMS J Comput* **16**(4):C348–C359, 2004.
41. Excoffier L, Slatkin M, Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population, *Mol Biol Evol* **12**:921–927, 1995.
42. Hawley ME, Kidd KK, HAPLO: A program using the EM algorithm to estimate the frequencies of multi-site haplotypes, *J Hered* **86**:409–411, 1995.
43. Stephens M, Smith NJ, Donnelly P, A new statistical method for haplotype reconstruction from population data, *Am J Hum Genet* **68**(4):978–989, 2001.
44. Niu T, Qin Z, Xu X, Liu JS, Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms, *Am J Hum Genet* **70**:157–159, 2002.
45. Bafna V, Gusfield D, Lancia G, Yooseph S, Haplotyping as perfect phylogeny: A direct approach, Technical Report, University of California, Davis, Davis, CA, 2002.
46. Eskin E, Halperin E, Large scale recovery of haplotypes from genotype data using imperfect phylogeny, in *Proc RECOMB 2003*, pp. 104–113, 2003.
47. Ding Z, Filkov V, Gusfield D, A linear-time algorithm for perfect phylogeny haplotyping, in *Proc RECOMB'05*, pp. 585–600, 2005.
48. Stephens M, Donnelly P, A comparison of Bayesian methods for haplotype reconstruction from population genotype data, *Am J Hum Genet* **73**:1162–1169, 2003.

49. Qin Z, Niu T, Liu J, Partitioning-ligation–expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms, *Am J Hum Genet* **71**:1242–1247, 2002.
50. Pfeiffer RM, Rutter JL, Gail MH, Struewing J, Gastwirth JL, Efficiency of DNA pooling to estimate joint allele frequencies and measure linkage disequilibrium, *Genet Epidemiol* **22**(1):94–102, 2002.
51. Wang S, Kidd KK, Zhao H, On the use of DNA pooling to estimate haplotype frequencies, *Genet Epidemiol* **24**(1):74–82, 2003.
52. Yang Y, Zhang J, Hoh J, Matsuda F, Xu P, Lathrop M, Ott J, Efficiency of single-nucleotide polymorphism haplotype estimation from pooled DNA, *Proc Natl Acad Sci USA* **100**(12):7225–7230, 2003.
53. Quade SR, Elston RC, Goddard KA, Estimating haplotype frequencies in pooled DNA samples when there is genotyping error, *BMC Genet* **6**:25, 2005.
54. de Bakker PI, Yelensky R, Pe'er I, Gabriel SB, Daly MJ, Altshuler D, Efficiency and power in genetic association studies, *Nat Genet* **37**:1217–1223, 2005.
55. Patil N *et al.*, Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21, *Science* **294**:1719–1723, 2001.
56. Zhang K, Deng M, Chen T, Waterman M, Sun F, A dynamic programming algorithm for haplotype partitioning, *Proc Natl Acad Sci USA* **99**(11):7335–7339, 2002.
57. McPeek MS, Strahs AH, Assessment of linkage disequilibrium by the decay of haplotype sharing, with application to fine-scale genetic mapping, *Am J Hum Genet* **65**(3):858–875, 1999.
58. Liu JS, Sabatti C, Teng J, Keats BJ, Risch N, Bayesian analysis of haplotypes for linkage disequilibrium mapping, *Genome Res* **11**:1716–1724, 2001.
59. Li J, Jiang T, Haplotype-based linkage disequilibrium mapping via direct data mining, *Bioinformatics* **21**:4384–4393, 2005.
60. Tzeng JY, Wang CH, Kao JT, Hsiao CK, Regression-based association analysis with clustered haplotypes through use of genotypes, *Am J Hum Genet* **78**(2):231–242, 2006.
61. S.A.G.E., Statistical Analysis for Genetic Epidemiology, http://darwin.cwru.edu/sage/, 2007.

Jing Li is an Assistant Professor in the Department of Electrical Engineering and Computer Science at Case Western Reserve University, Cleveland, OH, USA. He obtained a Ph.D. in Computer Science from the University of California, Riverside in 2004, and a B.S. in Statistics from Peking University, China, in 1995. His research interest is in the area of computational biology. More specifically, he focuses mainly on the design and implementation of efficient computational and statistical algorithms for the characterization of DNA variation in human populations and for the identification of correlations of DNA variation and phenotypic variation.



Tao Jiang received a B.S. in Computer Science and Technology from the University of Science and Technology of China, Hefei, China, in July 1984; and a Ph.D. in Computer Science from the University of Minnesota–Twin Cities, Minneapolis, MN, USA, in 1988. He was a faculty member at McMaster University, Hamilton, Ontario, Canada, from 1989 to 2001, and is now Professor of Computer Science and Engineering at the University of California, Riverside. He is also a member of the UCR Institute for Integrative Genome Biology, a member of the Center for Plant Cell Biology, a principal scientist at the Shanghai Center for Bioinformation Technology, and Changjiang Visiting Professor at Tsinghua University. Tao Jiang's recent research interests include algorithms, computational molecular biology, bioinformatics, and computational aspects of information gathering and retrieval. More information about his work can be found at http://www1.cs.ucr.edu/ jiang/