Original Paper

Human Heredity

Hum Hered 2009;67:248–266
DOI: 10.1159/000194978

Received: January 31, 2008
Accepted after revision: August 8, 2008
Published online: January 27, 2009

Haplotyping Methods for Pedigrees

Guimin Gao[a] David B. Allison[a] Ina Hoeschele[b]

[a]Department of Biostatistics, Section on Statistical Genetics, University of Alabama at Birmingham, Birmingham, Ala.,
[b]Virginia Bioinformatics Institute and Department of Statistics, Virginia Tech, Blacksburg, Va., USA

Key Words
Haplotype inference · Pedigree · Family data

Abstract
Haplotypes provide valuable information in the study of diseases, complex traits, population histories, and evolutionary genetics. With the dramatic increase in the number of available single nucleotide polymorphism (SNP) markers, haplotype inference (haplotyping) using observed genotype data has become an important component of genetic studies in general and of statistical gene mapping in particular. Existing haplotyping methods include (1) population-based methods, (2) methods for pooled DNA samples, and (3) methods for family and pedigree data. The methods and computer programs for population data and pooled DNA samples were reviewed recently in the literature. As several authors noted, family and pedigree datasets are abundant and have unique advantages. In the past twenty years, many haplotyping methods for family and pedigree data have been developed. Therefore, in this contribution we review haplotyping methods and the corresponding computer programs suitable for family and pedigree data and discuss their applications and limitations. We explore the connections among these methods, and describe the challenges that remain to be addressed.

Introduction

Importance of Haplotyping

A haplotype consists of the alleles at multiple linked loci (one allele at each locus) on the same chromosome. A multi-locus genotype, consisting of the genotypes at multiple linked loci, has known phase if its two haplotypes are known. A genotype with known parental origins of the alleles is an ordered genotype. Observed marker data in pedigrees or populations often consist of unordered genotypes and haplotypes are not directly observable. Haplotype information is valuable for many analysis methods in the study of diseases, complex traits, population histories, and evolutionary genetics [1]. For linkage analysis (with low-density markers in a long chromosomal region), haplotype inference can dramatically increase the information content over that attributable to any single marker [2]; haplotypes for a pedigree can be used to estimate identity by descent probabilities among pedigree members which provide the basis of many linkage analysis methods. For association studies or linkage disequilibrium (LD) tests (with high-density markers), there is clear evidence that using haplotypes in a (short) chromosomal region of interest can be more powerful than using individual markers in the analysis of complex traits in some circumstances [e.g., 1, 3–5]. Haplotype information can also be used to identify genotyping errors, through identification of double recombina-

KARGER

Fax +41 61 306 12 34
E-Mail karger@karger.ch
www.karger.com

0001–5652/09/0674–0248$26.00/0

Accessible online at:
www.karger.com/hhe

Ina Hoeschele
Virginia Bioinformatics Institute, Virginia Tech
Blacksburg, VA 24061-0477 (USA)
Tel. +1 540 231 3135, Fax +1 540 231 2606
E-Mail inah@vbi.vt.edu

tions in short chromosomal regions [6–8]. In addition, haplotype information was applied to infer missing genotypes in pedigrees. Burdick et al. [9] described a method combining sparse marker data for all individuals in a pedigree from a linkage scan and high-resolution SNP genotypes for a subset of the pedigree members to infer unobserved genotypes for related individuals by using haplotype information. This method provides a cost-effective way to use many existing linkage scan family collections for association studies.

Haplotypes in diploid species can be determined by experimental molecular techniques that are time consuming and/or expensive and therefore not suitable for large-scale application [5]. In silico haplotyping, which infers haplotypes from observed genotype data by statistical and computational methods, is thus be valuable if the estimation is accurate [1].

Existing haplotyping methods include population-based methods, methods for pooled DNA samples, and methods for family and pedigree data. Methods and software for population data and pooled DNA samples were recently reviewed [10, 11]. Family and pedigree datasets are abundant and have unique advantages [12]. Here we therefore review haplotyping methods and computer programs for pedigree data (including family data) and discuss their applications and limitations. We explore the connections among these methods and describe the challenges that remain to be addressed.

Haplotyping Methods for Pedigrees

Haplotyping in a pedigree refers to the reconstruction of the unknown true haplotype configuration from observed data (genotype data and pedigree structure). The space of all consistent haplotype configurations for a pedigree is often very large, in particular for a large pedigree with missing data. Statistical methods and genetic rule-based methods were developed to estimate the true haplotype configuration by identifying a single (or a set of) most likely, consistent haplotype configuration(s). A consistent haplotype configuration is an assignment of ordered genotypes to all members in the pedigree at all loci, such that the assignment is consistent with all observed data [13] and Mendelian segregation. For the sake of brevity, we use haplotype configuration to denote a consistent haplotype configuration. A haplotype configuration can be specified by a set of ordered genotypes $\mathbf{G} = (G_{i,j};\ i = 1, ..., n,\ j = 1, ..., L)$, where $G_{i,j}$ is the ordered genotype of pedigree member i at locus j, n is the pedigree size, and L is the number of markers.

Haplotyping methods for pedigrees include likelihood-based methods for a long chromosomal region with relatively low-density markers and genetic rule-based algorithms which are more appropriate for tightly linked markers in a small chromosomal region. Likelihood-based methods reconstruct configurations by maximizing the likelihoods or conditional probabilities of the configurations. Rule-based algorithms reconstruct configurations by minimizing the total number of recombinants in the pedigree data. Some of rule-based algorithms can account for marker-marker LD by estimating haplotype frequencies in founders.

Likelihood-based methods always assume Hardy-Weinberg equilibrium (HWE) at individual loci, and they often assume linkage equilibrium among markers because most of these methods were developed for long chromosomal regions with relatively low-density markers (say 2–10 cM microsatellite maps for linkage scan) [2, 8, 13–21]. To deal with denser clustered (SNP) markers, Abecasis and Wigginton [22] developed a likelihood-based method which can account for marker-marker LD within clusters of tightly linked markers and model recombination between clusters while assuming no recombination within clusters. All these likelihood-based methods are mainly used for linkage analysis.

The recent availability of high-throughput technologies (e.g., 500K SNP arrays) for genotyping at a large number of tightly linked SNP markers poses new challenges for haplotyping. Traditional haplotyping methods developed for pedigrees or families with low-density markers assuming linkage equilibrium can produce inaccurate results [7, 22–24] when applied to high-density markers with moderate to large amounts of LD. Thus, population-based haplotyping approaches handling LD among tightly linked (SNP) markers were extended to use trio or nuclear family data [1, 5, 23]. The haplotypes inferred by these methods can be used for genome-wide association studies and are valuable for the study of population and evolutionary genetics.

In the context of linkage analysis assuming linkage equilibrium among markers, the true configuration does not necessarily have the highest likelihood or minimum number of recombinants among all consistent haplotype configurations [2, 14]. The haplotype configurations estimated by the statistical methods or genetic rule-based algorithms are not guaranteed to include the true configuration, and a configuration with minimum number of recombinants may not have the highest likelihood [2, 14], in particular when the distances between adjacent markers are quite variable. In addition, in some situations

the possible ordered genotypes of an unordered individual-marker (i.e., an individual-marker with unordered genotype as a result of the observed genotype being heterozygous or missing) may all have equal probability conditional on the observed data in a pedigree, where an individual-marker denotes a combination of a specific individual and a specific marker locus. This phenomenon can be referred to as ambiguity at the individual-marker [25]. Arbitrarily choosing an ordered genotype, as many haplotyping computer programs do, does not account for the ambiguity and can of course result in assigning an incorrect ordered genotype to the individual-marker [25]. Furthermore, it is possible that many different haplotype configurations for the pedigree have the same highest likelihood value due to ambiguity at some individual-markers. Identifying arbitrarily a single or a small subset of these haplotype configurations may cause misassignments of haplotypes to some pedigree members. Computer programs providing a single (or a small subset of) haplotype configuration(s) without pointing out the individual-markers with ambiguity may mislead the users [25]. The missassigned haplotypes can potentially introduce errors into both linkage and association studies [25]. It is noteworthy that the most popular computer software programs including GENEHUNTER [15], SimWalk2 [2, 18], and Merlin [8] report the most likely haplotype configuration(s) in their haplotyping module but perform linkage analyses using all possible or a large sample of configurations rather than only the reported most likely configuration(s) in their haplotyping module [25].

Criteria for Evaluating Haplotyping Methods

Simulation studies are typically performed to evaluate the accuracy of haplotyping methods. Because haplotyping algorithms are computationally very demanding, computer running time is often an additional criterion for evaluating the efficiency of the haplotyping methods.

In the context of linkage analysis with low-density markers, comparing haplotyping methods in terms of the likelihood values of their identified configurations or in terms of the effects of the identified configurations on the accuracy of disease gene or quantitative trait locus (QTL) mapping is more important than comparing the identified configurations to the true haplotype configuration [26].

For high-density markers in LD, typically in a short chromosomal region, criteria such as switch error, incorrect genotype percentage, incorrect haplotype, and χ^2 distance [1] can be used for comparing haplotyping methods. These criteria are based on evaluating the differences between the identified haplotypes and the true haplotypes for each individual.

For evaluation of rule-based methods, the numbers of recombinants in the identified haplotype configurations replace the likelihood values of these configurations as an evaluation criterion.

In the following sections, we first introduce several basic likelihood-based pedigree analysis algorithms and haplotyping methods based on these algorithms. Usually these haplotyping methods can analyze large numbers of markers in pedigrees of small or moderate size, or very small numbers of markers in large, simple pedigrees. Second, we describe stochastic and deterministic likelihood-based approximation approaches for haplotyping in large complex pedigrees with large numbers of loci. The methods reviewed in these two sections are primarily used for data on large chromosomal regions with low-density markers in the context of linkage analysis. Third, we review genetic rule-based haplotyping methods, which are often used for tightly linked markers in small chromosomal regions. Forth, we describe haplotyping methods for (genome-wide) high-density (SNP) markers which were extended from population-based approaches to using nuclear families or trios. These methods account for marker-marker LD and are designed for inferring haplotypes for parents (founders) rather than haplotype configurations for entire families. Finally, we summarize the haplotyping methods reviewed in this article and discuss future directions.

Likelihood-Based Methods for Long Chromosomal Regions: I. Basic Algorithms and Their Applications to Haplotyping

For sparse markers in a long chromosomal region (tens to hundreds of centimorgans) in the context of linkage analysis, likelihood-based haplotyping methods often assume linkage equilibrium among markers in addition to HWE at each marker and adopt Haldane's no interference model of recombination [27]. The likelihood-based methods described in this and the next sections often ignore LD among markers and therefore may produce inaccurate haplotype inference [7], in particular for denser markers, but these methods are very useful in the context of linkage analysis.

Exact methods [e.g., 8, 15–17] based on the Lander-Green algorithm [28] can be used for large numbers (thousands) of loci in pedigrees of small or moderate size

(for example, 15–40 members on a dual 2.4-GHz IBM Blade with 2 GB of internal memory, see also [17]). The exact method of Fishelson et al. [29] using a Bayesian network approach can accommodate pedigrees with a few hundred individuals and with small numbers of markers, small pedigrees with a few hundred markers, or pedigrees with moderate sizes and moderate number of makers (say 150 individuals and 5 markers, 5 individuals and 150 markers, or 50 individuals and 10 or more markers). Approximation methods, including stochastic and deterministic approaches, can be applied to large complex pedigrees with several hundreds or thousands of members and with many maker loci [e.g., 2, 13, 14, 18–21].

Notation

For a pedigree with n_1 non-founders, the inheritance vector at a locus j, $\boldsymbol{v}_j$, represents gene flow in the pedigree through a sequence of $2n_1$ binary digits, i.e., $\boldsymbol{v}_j = (p_{1,j}, m_{1,j}, p_{2,j}, m_{2,j}, ..., p_{n_1,j}, m_{n_1,j})$, where the binary elements $p_{k,j}$ and $m_{k,j}$ describe the outcome of the paternal and maternal meioses, respectively, giving rise to non-founder k (k = 1, ..., n_1). For example, $p_{k,j} = 0$ or 1 denotes that the grandpaternal or grandmaternal allele was transmitted to non-founder k in the paternal meiosis [15]. Elements $p_{k,j}$ and $m_{k,j}$ are called *meiosis indicators* [30]. Thompson [30] uses $S_{i,j}$ to denote the meiosis indicator for meiosis i at locus j; $S_{\cdot,j} = (S_{i,j}, i = 1, ..., I)$ to denote the set of meiosis indicators for all I (= $2n_1$) meioses in the pedigree at a single locus j ($S_{\cdot,j} = \boldsymbol{v}_j$ is the inheritance vector at locus j); $S_{i,\cdot} = (S_{i,j}; j = 1, ..., L)$ to denote the set of meiosis indicators for all L loci in a single meiosis i; and $\mathbf{S} = (S_{i,j}; i = 1, ..., I, j = 1, ..., L) = (\boldsymbol{v}_1, ... \boldsymbol{v}_L)$ to denote meiosis indicators for all meioses at all loci. Similarly, $G_{\cdot,j}$ denotes a set of ordered genotypes for all pedigree members at a single locus j, and $G_{k,\cdot}$ denotes a set of ordered genotypes for all loci in a single pedigree member k. We also use g_k to denote a particular genotype of individual k at a single locus or at multiple loci; this genotype can be ordered or unordered.

Sobel and Lange [18] referred to a realization of the indicator matrix **S** as a descent graph that specifies the paths of gene flow in a pedigree. For a single locus and a single founder, the descent graph consists of two (binary) gene flow trees, each rooted at a (different) founder allele node. They also defined a descent state that specifies both the paths of gene flow and the actual founder alleles dropped down every path of gene flow. Only a descent state (not a descent graph) can specify a haplotype configuration **G**. A haplotype configuration can be consistent with multiple descent states because the configuration cannot specify the grandparental origin of an allele at a locus if the allele descended from a parent with a homozygous genotype. We note that herein we use the same symbol (e.g., $S_{\cdot,j}$) to denote a variable (or vector) and any of its realizations.

An Exhaustive Enumeration Method

For a simple, small and fully typed pedigree with a small number of linked loci, it is not difficult to implement an exhaustive enumeration method which enumerates all consistent haplotype configurations, calculates their likelihoods, and identifies the most probable configurations [2]. This exhaustive enumeration method quickly becomes computationally infeasible for larger pedigrees, larger numbers of markers, or pedigrees with missing data.

Viterbi Algorithm, Lander-Green Algorithm and Their Application to Haplotyping for Large Numbers of Loci in Pedigrees of Small or Moderate Size

Viterbi algorithm. For a hidden Markov model (HMM), let q_t denote the (hidden) state and O_t the observation symbol at time t (t = 1, 2, ..., T). Let $\boldsymbol{\Omega}_\mathbf{s}$ denote the space of all possible (hidden) states and $\boldsymbol{\Omega}_\mathbf{o}$ the space of all possible observation symbols, then $q_t \in \boldsymbol{\Omega}_\mathbf{s}$, and $O_t \in \boldsymbol{\Omega}_\mathbf{o}$. The joint probability of a realization of the state sequence $\mathbf{Q} = (q_1, q_2, ..., q_T)$ and the observation sequence $\mathbf{O} = (O_1, O_2, ..., Q_T)$ can be calculated as

$$P(\mathbf{Q}, \mathbf{O}) = P(q_1, q_2, ..., q_T, O_1, O_2, ..., O_T)$$
$$= P(q_1)P(O_1 \mid q_1)P(q_2 \mid q_1)P(O_2 \mid q_2) ... P(q_T \mid q_{T-1})P(O_T \mid q_T),$$

where $P(q_1)$ is the initial state probability, $P(O_t \mid q_t)$ is the probability of observation symbol O_t at time t given the state q_t (t = 1, 2, ..., T), and $P(q_{t+1} \mid q_t)$ is the state transition probability from state q_t at time t to state q_{t+1} at time t + 1 (t = 1, 2, ..., T–1).

For the given observation sequence $\mathbf{O} = (O_1, O_2, ..., Q_T)$, the Viterbi algorithm [31] finds the single best (hidden) state sequence, which maximizes $P(\mathbf{Q} \mid \mathbf{O})$, or equivalently $P(\mathbf{Q}, \mathbf{O})$. The Viterbi algorithm includes a forward step of maximization over each element (variable) q_t of **Q** in one direction ($q_1, q_2, ..., q_T$) by maximizing the product of all factors that depend on q_t and storing the maximum values related to q_t, and a backtracking step assigning an optimal value to each q_t in the reverse direction (see Appendix I for details).

Lander-Green algorithm. To handle large numbers (thousands) of markers in pedigrees, Lander and Green [28] proposed an algorithm based on an HMM (see also

[32]). The Lander-Green algorithm considers the inheritance pattern across a set of loci, $\mathbf{S} = (\boldsymbol{v}_1, ..., \boldsymbol{v}_L)$, which is not explicitly observable, as the state sequence in the HMM, with recombination causing state transitions between two adjacent loci. The space of hidden states $\boldsymbol{\Omega}_s$ is the set of possible realizations of the inheritance vector at a locus. The genotypes of all pedigree members at a single locus are treated as an observation, and the observed marker data at all loci $\mathbf{M} = (M_{\cdot,1}, ..., M_{\cdot,L})$ are treated as the observation sequence in the HMM, where $M_{\cdot,j}$ denotes the observed marker data of all pedigree members at locus j. For a pedigree of small or moderate size, the likelihood of the observed marker data $\mathbf{M}$, $P(\mathbf{M}) = \Sigma_s P(\mathbf{S}, \mathbf{M})$, can be calculated efficiently by using the HMM (see also [30]).

Haplotyping Methods based on Viterbi algorithm and Lander-Green algorithm. The HMM described in Lander and Green algorithm was used to reconstruct haplotype configurations at L loci [15] based on choosing the optimal inheritance vectors by an approximate approach and an exact approach. The approximate approach selects the most likely inheritance vector at each locus j, $\boldsymbol{v}_j$, such that the marginal distribution $P(\boldsymbol{v}_j \mid \mathbf{M})$ is largest, where $P(\boldsymbol{v}_j \mid \mathbf{M})$ can be calculated by a forward-backward process [30, 31]. The approximate method is simple and fast, and tends to yield results similar to those from the exact method. The exact approach treats all loci jointly and selects the most likely set of inheritance vectors at all loci $\mathbf{S} = (\boldsymbol{v}_1, ..., \boldsymbol{v}_L)$ such that the joint distribution $P(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_L \mid \mathbf{M})$ is largest. The exact approach was implemented by using the Viterbi algorithm [31]. The Viterbi algorithm has theoretical appeal because it finds the globally most likely inheritance pattern [15]. Both the approximate and the exact methods involve computing and/or saving intermediate conditional probabilities of all possible assignments of the inheritance vector at each locus; the computing time, memory and disc space requirements increase linearly with the number of loci but exponentially with the number of non-founders [15].

After the optimal inheritance vectors at all loci (the optimal descent graph **S**) have been determined, a haplotype configuration with the highest likelihood can be obtained by identifying a vector of alleles for all founders at each locus that is compatible with **S** and the observed genotypes in the pedigree and has the highest probability (product over the allele frequency of the alleles in the vector) [18].

These two haplotyping methods were implemented in a widely used software package, GENEHUNTER [15], which can handle large numbers of loci but only pedigrees of small size (approximately $\leq$15, see also [17]).

Abecasis et al. [8] extended the scope of the Viterbi algorithm and the Lander-Green algorithm to larger data sets by using sparse gene flow trees, which are a reduced representation of the full binary trees that combines gene flow patterns with identical outcome into symmetric and premature leaf nodes. Computing time, memory and disc space requirements are reduced so that pedigrees of moderate size (e.g., about 30 members on a dual 2.4-GHz IBM Blade with 2 GB of internal memory, see also [17]) and with large numbers (thousands) of markers can be analyzed efficiently. The method has been implemented in the popular software package, Merlin [8], which can produce both exact and approximate estimates of the most likely haplotype configuration for a pedigree. For a large number of dense markers where the probability of observing several recombinants between consecutive markers in a pedigree is close to zero, Merlin can construct approximate solutions by restricting the number of recombination events between consecutive markers. For example, if assuming two or fewer recombination events between two consecutive markers, then in the HMM only those assignments of the inheritance vector at marker $j + 1$ are considered that have at most two recombinants with the assignment $\boldsymbol{v}_j$ at marker j. This approximation is quite accurate for dense markers and reduces computing time significantly [8].

The Lander-Green algorithm was also implemented for haplotyping in other software programs, including Mendel (www.genetics.ucla.edu/software) and Allegro [16, 17]. These programs can handle large numbers (thousands) of markers but only pedigrees of small to moderate size. Allegro version 2 (Allegro2) is based on multiterminal binary decision diagrams (MTBDDs) [33] that are a generalization of binary decision diagrams (BDDs) and offer compact storage of probability distributions through two properties: uniqueness and nonredundancy [17, 34, see also below]. Some of the compactness is also captured by the gene flow trees used in Merlin. Simulation studies of Gudbjartsson et al. [17] show that Allegro2 can analyze some pedigrees with 36 members that Merlin (version 0.10.2) cannot analyze on a dual 2.4-GHz IBM Blade with 2 GB of internal memory. A BDD represents a Boolean function as a single rooted, directed acyclic graph [35]. In the graph, the terminal vertices are labeled 0 or 1, corresponding to the possible function values. Each nonterminal vertex v is labeled by a variable X_v ($X_v = 0$ or 1). The function value is determined by tracing a path from the root to a terminal vertex, following the appropriate edge

from each vertex. BBDs are usually compact and canonical (i.e., unique) symbolic representations, where representation of redundant information is completely avoided and all identical substructures are shared [36]. An MTBDD is an extension of a BDD from Boolean function values to values of any finite domain, which has multiple terminals with values other than 0 and 1. In Allegro2, an MTBDD is used to represent the probability distribution $p(\boldsymbol{v}_j \mid M_{\cdot,j})$ over possible realizations of the inheritance vector $\boldsymbol{v}_j = (p_{1,j}, m_{1,j}, p_{2,j}, m_{2,j}, ..., p_{n_1,j}, m_{n_1,j})$ at marker locus j in a pedigree [34]. The MTBDD contains vertices, each of which corresponds to a particular meiosis at locus j and is labeled by a variable that is an element of $\boldsymbol{v}_j$. When there is little information about the genotypes at locus j, the resulting MTBDD becomes small as most of possible realizations of the inheritance vector $\boldsymbol{v}_j$ have the same probability and are therefore captured by the sharing of the structure [34].

A method accounting for marker-marker LD in pedigrees with clustered markers. For pedigree data, the haplotyping methods assuming linkage equilibrium among markers can produce inaccurate results, in particular when applied to tightly linked (SNP) markers in LD [7, 22–24], and use of this haplotype information in linkage or association studies can adversely affect mapping accuracy [7]. Because highly dense SNP genotyping may now be more cost-effective than sparse microsatellite typing and hence be used for haplotyping and linkage analysis in pedigrees, Abecasis and Wigginton [22] proposed a practical approach that is cable of utilizing such data. The method assumes that markers can be organized into nonoverlapping clusters of consecutive markers, markers in the same cluster may be in LD, LD among markers in different clusters can be ignored, and the recombination rate within each cluster is near zero. The method uses haplotype frequencies to describe patterns of LD within each cluster and models recombination between clusters by a HMM, based on the Lander-Green algorithm with the modification that inheritance vectors are built at the cluster level rather than at the individual locus level. Haplotype frequencies within each cluster are estimated by a novel application of the gene-counting-based EM algorithm to pedigree data.

The haplotyping method of Abecasis and Wigginton [22] combines the advantages of both population-based haplotyping approaches (see below) and haplotyping methods for pedigrees assuming linkage equilibrium: The former can account for LD among markers within each cluster, and the latter can accommodate recombination between clusters in a pedigree in a long chromosomal region. This method uses lowly polymorphic but abundant SNP markers to form a smaller number of highly polymorphic cluster- or super-loci, which increases the accuracy of haplotyping and linkage analysis compared to traditional methods. Their haplotyping method has been implemented in the Merlin package [8] used for linkage analysis. As noted earlier, this software employs the Lander-Green algorithm, and therefore it allows large numbers of markers while being restricted to pedigrees of small or moderate size. Of course, this method is also suitable for nuclear family, trio or sibship data with high-density markers. A possible extension of the haplotyping method of Abecasis and Wigginton to large pedigrees is to apply the LM-sampler [30, see also below] to the each marker cluster.

We note that the primary purpose of the haplotyping method of Abecasis and Wigginton is to perform more accurate linkage analysis, rather than to perform genome-wide haplotype-based association studies. When only considering a small chromosomal region with tightly linked markers and treating the markers as a single cluster, then the haplotyping method of Abecasis and Wigginton is equivalent to the rule-based methods (see below) assuming no recombination and using founder population haplotyping frequencies, such as ZAPLO [37] and HAPLORE [38], except that ZAPLO [37] and HAPLORE may be able to handle larger pedigrees.

Peeling, Reverse Peeling, and Their Application to Haplotyping

For a large pedigree (without loops), Elston and Stewart [39] introduced a recursive approach, referred to as peeling, to simplify the calculation of the likelihood of observed phenotype vector $\mathbf{x} = (x_1, ..., x_n)$, $L = P(\mathbf{x}) = \Sigma_{\mathbf{g}} P(\mathbf{x} \mid \mathbf{g}) P(\mathbf{g})$, where $\mathbf{g} = (g_1, ..., g_n)$ is the genotype vector, n is pedigree size, and g_k is a specific genotype of individual k at a single locus or at multiple loci ($k = 1, ..., n$); this genotype can be unordered. For multiple loci, g_k represents a phase known multilocus genotype (with two know haplotypes). When pedigree size n is *large*, the number of possible assignments of $\mathbf{g}$ becomes too large to enumerate all possible assignments and calculate their likelihood values. The Elston-Stewart algorithm can be used to efficiently calculate likelihood L by representing it as a telescopic sum and eliminating the right-most sum in each step [see 40 (p. 184) and 41 for details].

The Elston-Stewart algorithm was extended to evaluate the likelihood of complex pedigrees with loops and to permit peeling in any optimal peeling sequence (order of pedigree members) [e.g., 40–44]. For these peeling algo-

rithms the computational demand increases linearly with the number of pedigree members but exponentially with the number of loci [40]. For larger pedigrees or complex loops, exact peeling (even at a single locus) can become computationally demanding or infeasible. Thus approximation peeling methods were proposed [e.g., 45–47].

The Elston-Stewart and Lander-Green algorithms can both be considered as special cases of the Lauritzen and Spiegelhalter evidence propagation algorithm [48, 49], but they use different latent variables (genotypes and meiosis indicators, respectively) and peel in a different direction [50]. The Elston-Stewart algorithm works by moving through the pedigree members (pedigree peeling) while the Lander-Green algorithm works along the chromosome (chromosome peeling).

Reverse peeling. The (pedigree and chromosome) peeling algorithms can be used to sample jointly a set of latent variables (genotypes or meiosis indicators) by a procedure known as reverse peeling [30, 44]. Reverse peeling calculates and saves conditional probabilities of the latent variables in one direction by using peeling and samples realizations of the latent variables in the reverse direction. We describe two typical examples using reverse peeling below; the ideas of these examples have been widely used for haplotyping in pedigrees.

Example 1: Ploughman and Boehnke [44] proposed a method using reverse (pedigree) peeling to sample jointly the genotype vector $\mathbf{g} = (g_1, ..., g_n)$ at a single locus for a pedigree of n members. Given the observed phenotype vector $\mathbf{x} = (x_1, ..., x_n)$, where x_i can be a marker phenotype (the genotype data at a set of markers) and/or a trait phenotype for pedigree member i, the joint probability of the genotype vector $\mathbf{g}$ can be factored into univariate conditional probabilities according to a peeling sequence, or

$$P(\mathbf{g}|\mathbf{x}) = \prod_{k=1}^{n} p(g_k|g_1,\ldots,g_{k-1},\mathbf{x}) = \prod_{k=1}^{n} p(g_k|g_1,\ldots,g_{k-1},x_k,\ldots,x_n). \quad (1)$$

The conditional probabilities are calculated by (exact) peeling and saved in the order of the peeling sequence, and then all g_k are sampled from the corresponding conditional probabilities $p(g_k \mid g_1, ..., g_{k-1}, x_k, ..., x_n)$ in the reverse order.

For a pedigree without loops, pedigree can be peeled nuclear family by nuclear family in a sequence of nuclear families (a peeling sequence) such that each family which is peeled next has only one connector, where a nuclear family is a sibship and its parents, and a connector is an individual that is a member of multiple nuclear families. In that case, each conditional probability $p(g_k \mid g_1, ..., g_{k-1}, x_k, ..., x_n)$ can be simplified and depends on the genotypes of at most two other individuals. This process can reduce the number of terms to be saved.

An optimal peeling sequence can be found with the method of Fernandez and Fernando [51], which uses algorithms for determining an optimal order of elimination in sparse systems of equations, but this method is not feasible for large pedigrees with complex loops which require approximate peeling (see below).

When using exact peeling, reverse (pedigree) peeling can be computationally intensive or even infeasible for pedigrees with complex loops. To overcome this problem, Fernandez et al. [47] described an approximate reverse peeling strategy based on an approximate peeling method, iterative peeling [45, 46]. Approximate reverse peeling is fast and accurate for large pedigrees with complex loops. It has been widely used in animal breeding. This method might be also useful for improving some of the existing Monte Carlo methods using exact reverse peeling in human genetics.

Example 2: The objective here is to sample jointly a set of meiosis indicator vectors $\mathbf{S} = (S_{\cdot,j}, j = 1, ..., L)$ based on chromosome peeling [30]. The joint distribution of $\mathbf{S}$ given observed marker data $\mathbf{M}$ can be written as

$$\mathrm{P}(\mathbf{S}\,|\,\mathbf{M}) = \prod_{j=L}^{1} \mathrm{P}\left(S_{\cdot,j}\,|\,S_{\cdot,j+1},\ldots,S_{\cdot,L},\mathbf{M}\right) = \prod_{j=L}^{1} \mathrm{P}\left(S_{\cdot,j}\,|\,S_{\cdot,j+1},M_{\cdot,1},\ldots,M_{\cdot,j}\right), \quad (2)$$

where, when $j = L$, $\mathrm{P}(S_{\cdot,j} \mid S_{\cdot,j+1}, M_{\cdot,1}, ..., M_{\cdot,j})$ is defined as $\mathrm{P}(S_{\cdot,L} \mid M_{\cdot,1}, ..., M_{\cdot,L})$. It is assumed that $S_{\cdot,j}$ has a first-order Markov structure and that $(M_{\cdot,k}, k = j + 1, ..., L)$ and $S_{\cdot,j}$ are mutually independent given $S_{\cdot,j+1}$. Obtaining realizations of S involves a forward computation of the left-conditional probabilities $\mathrm{P}(S_{\cdot,j} \mid M_{\cdot,1}, ..., M_{\cdot,j})$ in the order $j = 1, 2, ..., L$, calculating the conditional probabilities of $S_{\cdot,j}$ in equation (2) as $\mathrm{P}(S_{\cdot,j} \mid S_{\cdot,j+1}, M_{\cdot,1}, ..., M_{\cdot,j}) = c \cdot \mathrm{P}(S_{\cdot,j}, \mid M_{\cdot,1}, ..., M_{\cdot,j}) \cdot \mathrm{P}(S_{\cdot,j+1} \mid S_{\cdot,j})$, and sampling $S_{\cdot,j}$ in the reverse order starting with $S_{\cdot,L} \sim \mathrm{P}(S_{\cdot,L} \mid M_{\cdot,1}, ..., M_{\cdot,j})$, where c is a normalization constant [see 30 for details]. Because at each locus j, the left-conditional probabilities $\mathrm{P}(S_{\cdot,j}, \mid M_{\cdot,1}, ..., M_{\cdot,j})$ for all possible realizations of $S_{\cdot,j}$ must be saved and the number of possible realizations of $S_{\cdot,j}$ increases with the number of non-founders, this method becomes computationally infeasible for large pedigrees. The advantage of this method is that it can be applied to analyzing large numbers of markers.

An advantage of (exact) reverse peeling is that it is unbiased [44] and based on the exact calculation of conditional probabilities. A sampling scheme for a latent vector $\mathbf{g} = (g_1, ..., g_n)$ given observed data $\mathbf{x}$ is called unbiased if

the probability of sampling a realization of **g** is equal to the conditional probability $P(\mathbf{g} \mid \mathbf{x})$; i.e., samples generated by the scheme are obtained directly from the target distribution $P(\mathbf{g} \mid \mathbf{x})$. In contrast, a Markov chain Monte Carlo (MCMC) sampler may be biased when only a subspace of the target distribution is sampled due to theoretical reducibility (a Markov chain is irreducible only if every state is accessible from every state) or practical reducibility (very slow mixing). Thus, reverse peeling can generate more accurate estimates compared with a MCMC sampler. Here we will use 'joint sampling' or 'joint updating' to refer to an unbiased sampling scheme for a latent vector.

Two deterministic haplotyping methods using reverse peeling. Sobel et al. [2] proposed a conditional probability haplotyping method based on example 1. The only difference is that now an ordered haplotype pair at multiple loci or an ordered multilocus genotype ($G_{i,\cdot}$) instead of $\mathbf{g}_i$ for each pedigree member i is considered. For pedigree member i, the method identifies a single $G_{i,\cdot}$ that yields the largest conditional probability $P(G_{i,\cdot} \mid G_{1,\cdot}, ..., G_{i-1,\cdot}, \mathbf{x})$ given the set of ordered haplotype pairs ($G_{1,\cdot}$, ..., $G_{i-1,\cdot}$) assigned to the first $i-1$ pedigree members and observed data **x**. The identified vector ($G_{1,\cdot}$, ..., $G_{n,\cdot}$) provides a haplotype configuration that is often optimal for the pedigree. This haplotyping method can be applied to a large, simple pedigree with very a small number of marker loci.

Similarly, based on example 2, a possible deterministic approach for haplotyping in pedigrees of small to modest sizes using reverse chromosome peeling is to identify an approximately optimal descent graph or realization of $\mathbf{S} = (S_{\cdot,j}, j = 1, ..., L)$ by identifying each realization $S_{\cdot,j}$ that maximizes $P(S_{\cdot,j} \mid S_{\cdot,j+1}, M_{\cdot,1}, ..., M_{\cdot,j})$. However, the Viterbi algorithm implemented in the GENEHUNTER [15] and Merlin [8] software packages should perform better for haplotyping because it is an exact method that finds the globally most likely descent graph.

Bayesian Network and Its Application to Haplotyping

A Bayesian network specifies a joint distribution over a set of variables of interest by consisting of a directed acyclic graph (DAG), a family of (conditional) probability distributions, and their parameters [29, 52, 53]. The DAG is composed of vertices and directed edges, where each vertex v (= 1, ..., n) corresponds to a variable X_v (X_v can be a vector), and it does not have directed cycles. The graph represents conditional independence relationships (e.g., a vertex is independent of its ancestors given its parents) or direct relationships (among vertices that are not mediated by any other vertices).

In a Bayesian network, if there is a directed edge from vertex u to vertex v ($u \rightarrow v$), then u (X_u) is a parent (parent variable) of v, and v (X_v) is a child (child variable) of u. For a vertex v, let $pa(X_v)$ denote the set of parent variables of X_v, and $ch(X_v)$ denote the set of child variables of X_v. For the set of variables $\mathbf{X} = (X_1, ..., X_m)$ in a Bayesian network, suppose that $(X_1, ..., X_m)$ is ordered such that each child variable follows its parent variables. The joint probability of a realization for the set of variables $\mathbf{X} = (X_1, ..., X_n)$ can be factorized as

$$P(\mathbf{X}) = P(X_1, \ldots, X_m) = \prod_{v=1}^{m} P\left(X_v \mid pa(X_v)\right), \tag{3}$$

where $pa(X_v)$ is a small set, each X_v only depends on its parent variables, and **X** can contain some observed variables [54]. In equation (3), if $pa(X_v)$ is the empty set, then $P(X_v \mid pa(X_v)) = P(X_v)$.

An elim-mpe algorithm. Let $\mathbf{X}^* = (X_1^*, ..., X_n^*)$ denote the subset of unknown variables of **X** (i.e., $\mathbf{X}^* \subseteq \mathbf{X}$, $n \leq m$). For a given variable ordering $X_1, ..., X_m$, to find $\mathbf{X}^{*0} = (X_1^{*0}, ..., X_n^{*0})$ such that for $\mathbf{X}^* = \mathbf{X}^{*0}$, $P(\mathbf{X})$ is maximized, Dechter [54] described an elim-mpe algorithm that contains a first step which eliminates variables by maximization over each variable in the order $X_n^*, ..., X_1^*$ and stores intermediate maximum values (points) related to each variable, and a second step which assigns optimal values to the unknown variables in the reverse order $X_1^*, ..., X_n^*$ (see Appendix II for details).

The Viterbi algorithm is only used for an HMM (a special Bayesian network) and can be viewed as a special case of the elim-mpe algorithm, which is applicable to any Bayesian network. For a pedigree, the Viterbi algorithm can be implemented on the Bayesian network of an HMM [55] that contains a vertex corresponding to the unknown (vector) variable $S_{\cdot,j}$ and a vertex corresponding to the known (vector) variable $M_{\cdot,j}$ for each locus j ($j = 1, ..., L$). The variable elimination order is the order of the marker loci (see fig. 1).

The variable elimination in the elim-mpe algorithm can be viewed as another type of peeling that peels (eliminates) variables by maximization whereas the usual peeling methods peel variables by summation as in the Elston-Stewart algorithm. The elim-mpe algorithm only identifies the assignments of $(X_1^*, ..., X_n^*)$ having the highest probability. In contrast, reverse peeling can sample any realization of the $(X_1^*, ..., X_n^*)$ from their joint distribution.

A drawback of the elim-mpe algorithms is that for the elimination of any X_i^*, the algorithm requires considerable memory for storing the intermediate values of

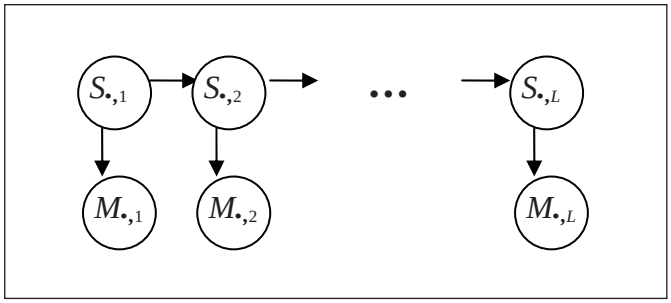

Fig. 1. Bayesian network of HMM on a pedigree [55].

$h_i(X_i^1, ..., x_i^{k_i})$ and $X_i^{*0}(X_i^1, ..., x_i^{k_i})$ for all possible $(X_i^1, ..., x_i^{k_i})$ (see Appendix II). Memory requirements can be minimized by minimizing the k_i values (the sizes of $(x_i^1, ..., x_i^{k_n})$). This can be accomplished by following an optimal peeling sequence (an optimal order of $(X_1^*, ..., X_n^*)$) as in the usual peeling approach. More generally, Fishelson et al. [29] described a method to determine an approximately optimal order of variables $(X_1^*, ..., X_n^*)$ by using a cost function (see below). In addition, to further reduce the amount of memory needed, Dechter [54] described a method combining variable elimination and *conditioning* (on a subset of variables), where the *conditioning* method enumerates all assignments of the subset of variables, performs the elim-mpe algorithm for each of these assignments, and then merges the results. *Conditioning* reduces the memory requirement but it increases computing time. We note that *conditioning* has also been used elsewhere to break a loop in a pedigree by enumerating all possible ordered genotypes of a member of the loop [42, 56].

An exact method using a Bayesian network. By using the elim-mpe algorithm combined with conditioning, Fishelson et al. [29] developed a maximum likelihood haplotyping method based on a Bayesian network representation of a pedigree. For each individual at each locus, the Bayesian network contains vertices corresponding to a known variable (observed genotype) and four unknown variables: paternal allele, maternal allele, and their corresponding meiosis indicators (for non-founders). The haplotyping method determines a (single) maximum-likelihood assignment to the unknown variables. The method also uses genotype elimination [e.g., 57] and allele recoding [58].

For the elim-mpe algorithm combined with conditioning, the ordering of $(X_1^*, ..., X_n^*)$ has a major effect on both computing time and memory requirements. Fishelson et al. [29] described a method to determine an approximately optimal order of the variables by using the undirected (moral) graph of a Bayesian network [52] and a cost function. The cost function of an ordering of the variables is an approximate measure of its computing time and memory requirements. The approximately optimal order of variables can be found without having to compute the conditional probabilities in equation (A2).

The haplotyping method of Fishelson et al. [29] was implemented in the software SUPERLINK, which provides an exact and flexible approach for the analysis of pedigree data. For example, it can accommodate pedigrees with a few hundred individuals and small numbers of markers, small pedigrees with a few hundred markers, or pedigrees with moderate sizes and moderate number of markers. This method should perform well in cases where pedigree peeling performs well. But it is not competitive with the Lander-Green chromosome peeling when analyzing thousands of markers in pedigrees of small or modest size. For an analysis of thousands of markers, the HMM Bayesian network should be used.

Likelihood-Based Methods for Long Chromosomal Regions: II. Approximation Approaches for Large Complex Pedigrees with Large Numbers of Loci

For large and complex pedigrees with large numbers of loci and in particular with substantial amounts of missing marker data, exact methods become infeasible. Therefore, MCMC methods were developed that sample haplotype configurations from their distribution conditional on the observed data and identify a single or a set of configurations with the highest likelihoods or conditional probabilities [2, 13, 18–20]. These methods can be applied to complex pedigrees with large numbers of loci but their computing time requirements can be high. Gao et al. [14] and Gao and Hoeschele [21] proposed a deterministic approximation method that can analyze large pedigrees (with thousands of members) and large numbers of loci efficiently but its current version does not permit missing data. As stated earlier, the methods described in this section ignore marker-marker LD (or essentially treat it as zero) and can cause inaccurate haplotype estimation.

Two MCMC Methods

Sobel et al. [2] and Sobel and Lange [18] proposed an MCMC method based on simulated annealing to identify a single, nearly optimal haplotype configuration. The

space of all consistent descent graphs over a pedigree is considered, and to each descent graph the likelihood of its most likely descent state is assigned (a descent graph can be consistent with multiple descent states). A Metropolis algorithm is employed to construct a Markov chain that uses certain transition rules for moves among descent graphs and converges to the correct equilibrium distribution. A starting descent graph can be produced by a genotype-elimination algorithm [2, 57]. A new descent graph proposal (another realization of the meiosis indicator matrix **S**) is generated in each cycle by changing the values of a subset of $S_{i,j}$ s in the current descent graph according to the transition rules. While the Metropolis algorithm is used to obtain a sample of (dependent) descent graphs from the equilibrium distribution (the conditional distribution of all consistent descent graphs), simulated annealing is employed to identify a single descent graph with the approximately maximal likelihood. This method was implemented in the widely used software program SimWalk2, which can analyze large complex pedigrees and large numbers of loci, but computing time can be substantial.

Lin and Speed [13] proposed another MCMC method, a Gibbs-Jump algorithm, to identify a set of haplotype configurations with high conditional probabilities rather than a single, approximately most likely configuration. A set of identified configurations with their corresponding conditional probabilities provides more information than the single, approximately most likely configuration, in particular when the most probable configuration is not guaranteed to be the true one [13]. The method samples each haplotype configuration (**G**) from the distribution of haplotype configurations given the observed genotype data (**M**) in a pedigree, $P(\mathbf{G} \mid \mathbf{M})$, by using a hybrid algorithm combining a Gibbs sampler with a Metropolis jumping kernel to achieve an irreducible Markov chain. In each cycle, the method updates the ordered genotype $G_{i,j}$ of each individual i at each locus j exactly once by Gibbs sampling and then attempts to switch by Metropolis jumping from the current haplotype configuration to a different configuration that could not be reached by Gibbs steps alone. The method can handle large pedigrees with large numbers of loci, but can be slow to converge.

Block Samplers Using Local Reverse Peeling

For large pedigrees with multiple linked loci, single-variable MCMC updating methods (methods which do not update (sub)sets of latent variables jointly) do not ensure proper mixing of the samplers [30]. Transition rules permitting passage through inconsistent descent graphs [2, 18] and the Gibbs-Jump algorithm of Lin and Speed [13] can improve mixing to some extent, but joint-updating schemes [e.g., 30, 59] may be more efficient approaches to improve mixing. However, updating all latent variables of $\mathbf{G} = (G_{i,\cdot}, i = 1, ..., n)$ or $\mathbf{S} = (S_{\cdot,j}, j = 1, ..., L)$ jointly by reverse peeling based on exact calculation of conditional probabilities can be computationally prohibitive for large pedigrees with large number of loci as discussed earlier. Approaches (block samplers) that jointly update a small subset of latent variables (e.g., the $S_{i,j}$ s or $G_{i,j}$ s of a subset of individuals at several loci) based on local exact calculations and reverse peeling in each cycle were developed [30, 59]. We refer to these approaches as local reverse peeling.

A sequential imputation method. Irwin et al. [60] developed a Monte Carlo method to sample haplotype configurations $\mathbf{G} = (G_{\cdot,1}, ..., G_{\cdot,L})$ based on sequential imputation of $G_{\cdot,1}, ..., G_{\cdot,L}$, which is a form of importance sampling. At each step, the ordered genotypes at locus j (the components of $G_{\cdot,j}$) are sampled jointly by using local reverse peeling from the distribution of $G_{\cdot,j}$ conditional on a set of sampled ordered genotype vectors at the first $j - 1$ loci $(G_{\cdot,1}, ..., G_{\cdot,j-1})$ and the observed marker data of all pedigree members at locus j $(M_{\cdot,j})$, $\Pr(G_{\cdot,j} \mid G_{\cdot,1}, ..., G_{\cdot,j-1}, M_{\cdot,j})$. The resulting samples of **G** are not samples from the correct conditional distribution, but by using importance sampling they can be re-weighted to the correct distribution. The chosen order of the loci does not necessarily correspond to their physical order, as it affects the variance of the weights [60]. The computing effort of the sequential imputation of $(G_{\cdot,1}, ..., G_{\cdot,L})$ increases linearly with the number of loci.

The sequential imputation method was extended by incorporating genetic interference using the χ^2 model and was applied to haplotyping in pedigrees [20, 61]. The method of Lin et al. [20] and Skrivanek et al. [61] independently samples a set of configurations, calculates a weight for each configuration, and identifies a set of configurations with the highest conditional probabilities. Generating independent samples may be more computationally efficient in some cases where MCMC methods converge very slowly due to strong dependencies among realizations of the Markov chain [61].

The sequential imputation method was implemented in the software SIMPLE, which can analyze large pedigrees and is more suitable for small to moderate numbers of loci. Because the method uses exact (pedigree) peeling at each locus, it can be computationally demanding for complex pedigrees.

LM-sampler. Heath [62] developed an MCMC method, the L-sampler, to update jointly the meiosis indicators in $S_{\cdot,j}$ at a single locus j, instead of updating the components of $G_{\cdot,j}$ as in the sequential imputation of Irwin et al. [60]. Generally, the space of latent variables is smaller for the $S_{\cdot,j}$ than for the $G_{\cdot,j}$ [30]. The L-sampler uses local reverse (pedigree) peeling based on the distribution of $S_{\cdot,j}$ conditional on the indicators at two immediate flanking markers ($S_{\cdot,j-1}$ and $S_{\cdot,j+1}$) and the marker data at locus j, $P(S_{\cdot,j} \mid S_{\cdot,j-1}\, S_{\cdot,j+1}\, M_{\cdot,j})$.

Thompson and Heath [63] presented another MCMC method, the M-sampler, to update jointly the components of $S_{i,\cdot}$, the meiosis indicators for all loci at a single meiosis i, by local reverse (chromosome) peeling (similar to example 2 in the section on reverse peeling), based on the distribution of $S_{i,\cdot}$ conditional on the indicators for all other meioses ($S_{k,\cdot}$, $k \neq i$) and the observed marker data **M** [see also 30].

The L-sampler works well on extended pedigrees but mixes poorly with multiple linked loci and becomes computationally demanding or infeasible when applied to large, very complex pedigrees; conversely, the M-sampler does not suffer from poor mixing due to tightly linked loci but it can mix poorly where there are extended ancestral paths of descent in a pedigree [30]. The LM-sampler combining these two samplers, for example with an L-sampler to M-sampler proportion of 20 to 80%, can achieve more robust and reliable performance [30, 64]. These samplers have been implemented in the well-known MORGAN package (www.stat.washington.edu/thompson/Genepi/MORGAN/Morgan.shtml.) and in the Loki package [62] for linkage analysis. The LM-sampler can be an efficient approach for haplotyping (via identifying a set of optimal **S**) in large pedigrees with large number of loci, but the current versions of the packages do not provide output for haplotyping. The M-sampler can be further improved by updating jointly the indicator vectors $S_{i,\cdot}$ for several meioses i. This strategy is effective and easily carried out for large pedigrees with several untyped top generations; likewise the L-sampler can be improved by updating jointly $S_{\cdot,j}$ for several loci j [30]. For the L-sampler, on a complex pedigree, usually no more than two or three loci can be updated jointly due to computational limitations [30].

A block Gibbs sampler. As an extension of the LM sampler, Thomas et al. [19] described a block Gibbs sampler using a mixed updating scheme that updates jointly a subset of latent variables at each step. The mixed updating scheme works with the following updating components:

(1) The elements of $S_{\cdot,j}$ and $G_{\cdot,j}$ at each single locus j are updated jointly, conditional on the meiosis indicators $S_{\cdot,j-1}$ and $S_{\cdot,j+1}$ and the marker data at locus j ($M_{\cdot,j}$), similar to the L-sampler. On very complex pedigrees for which reverse pedigree peeling at locus j is very time consuming or infeasible, the method updates several overlapping subsets of latent variables whose union is the set of all the variables at the locus [see also 42].

(2) The meiosis indicators at all loci for a small subset of k meioses (k rows of **S**) are updated jointly, conditional on the values of all other meioses and the observed marker data **M**, similar to the M-sampler, and all ordered genotypes (**G**) are updated.

(3) A specific subset of meiosis indicators at a locus is updated according to rules which are extensions of the transition rules of Sobel and Lange [18]. For example, in each nuclear family in the pedigree, all paternal (or maternal) meiosis indicators at locus j of all children are flipped. This updating step is important for obtaining a chain with good mixing properties [19].

This block sampler was implemented in the software MCLINK for linkage analysis [19]. Although it is efficient for haplotyping in large pedigrees with large numbers of loci, MCLINK does not provide output for haplotyping.

A Conditional Enumeration Method

Although the stochastic sampling methods described above can be applied to complex pedigrees, their computational requirements can still be high or even prohibitive for some large pedigrees and high marker densities. For haplotyping efficiently in large pedigrees with hundreds or thousands of members and with large numbers of (dense) markers, Gao et al. [14] and Gao and Hoeschele [21] developed a deterministic approximation, the conditional enumeration haplotyping method (CEHM), to identify a set of haplotype configurations with the highest likelihoods.

Let $\mathbf{U} = (M_1, ..., M_t)$ denote the set of individual-markers that can not be ordered directly by the observed data (e.g., parental genotypes) in a pedigree, where an individual-marker M_i is a combination of a specific individual and a specific marker locus. Let G_i denote one of the possible ordered genotypes at individual-marker M_i. The joint probability of a set of assignments (G_1, ..., G_t) to the individual-markers in U conditional on the observed data (**M**) is

$$P(G_1,\ldots,G_t \mid \mathbf{M}) = \prod_{i=1}^{t} P(G_i \mid G_1,\ldots,G_{i-1}, \mathbf{M}) = \prod_{i=1}^{t} p_i, \qquad (4)$$

where equation (4) is a variant of equation (1), and $p_i = P(G_i \mid G_1, ..., G_{i-1}, \mathbf{M})$ is the probability of an ordered genotype G_i at individual-marker M_i, conditional on a set of assignments (G_1, G_2, ..., G_{i-1}) at the first $i - 1$ individual-markers, and the observed data **M**.

In the CEHM, the conditional probabilities p_i are calculated by an approximation approach that uses only the closest, informative flanking markers of the individual under consideration and its parents and offspring [14]. An optimal reconstruction order of the individual-markers in **U** is determined based on probabilities p_i such that the individual-markers M_i with more information (as measured by p_i) are reconstructed earlier. Following this optimal order the method successively assigns one or more most likely ordered genotype(s) to each individual-marker M_i by using two user-determined threshold parameters: a threshold for the conditional probabilities p_i and a threshold for the ratio of the conditional probability of a haplotype configuration to the (unknown) largest conditional probability of all possible configurations [21].

The CEHM was tested on published and simulated data sets and was shown to be much faster than SimWalk2 and to identify configurations with much higher likelihood values than the likelihood value of the single configuration identified by SimWalk2 [14, 21]. However, the current version of the CEHM can only handle pedigrees with complete marker data for all individuals at all loci under the assumption of linkage equilibrium among markers (this assumption can lead to biased results).

Accounting for Genetic Interference

Likelihood-based haplotyping methods always adopt Haldane's model of recombination, which is known to be a simplification that ignores the phenomenon of genetic interference as it assumes that recombination occurs independently on disjoint intervals of a chromosome [2]. Ignoring genetic interference may lead to obligatory multiple recombinants in a haplotype [20]. Broman and Weber [65] and Lin et al. [66] showed that a χ^2 (interference) recombination model can fit human data adequately. This χ^2 model can be incorporated into some of the likelihood-based methods [20]. Skrivanek et al. [61] and Lin et al. [20] used the χ^2 model in their sequential imputation haplotyping method, which is implemented in the software SIMPLE. By accounting for genetic interference, SIMPLE can greatly reduce the chance of inferring multiple recombinations in a short segment [20].

Rule-Based Methods for Short Chromosomal Regions

Genetic rule-based haplotyping algorithms [6, 37, 38, 67–71] reconstruct haplotype configurations by minimizing the total number of recombinants in a pedigree. These methods often do not (fully) utilize the information about inter-marker distances and are more appropriate for tightly linked markers in a small chromosomal region. Some of these methods can account for LD among markers by estimating haplotype frequencies in founders.

Haplotyping Assuming No Recombination among Markers

Wijsman [67] proposed a set of twenty genetic rules and implemented them in a computer program for haplotype reconstruction in family data with tightly linked markers in a small region, under the assumption of no recombination among markers. These rules form a basis for many other rule-based algorithms [e.g., 38].

To estimate haplotype frequencies and identify the most common haplotypes for high-density (SNP) markers in LD in a chromosomal region in pedigrees, O'Connell [37] presented a haplotyping algorithm under the assumption of no recombination among markers. The algorithm was implemented in the software, ZAPLO, which recodes the haplotypes at a set of tightly linked markers as alleles at a single locus. After recoding, genotype elimination [56, 57] is performed using the recoded alleles to delete inconsistent genotypes and to find all possible (zero-recombinant) haplotype configurations of the pedigree data. Haplotype frequencies (for founders) are estimated based on these configurations by an expectation-maximization (EM) algorithm [72]; this step accounts for LD among markers. To handle a large number of linked loci, the divide-and-conquer approach is employed to split the region into smaller pieces, and a pruning method is used to delete genotypes with low probabilities. Although this method is computationally demanding for a large pedigree with a large number of loci, the ideas underlying this method are very useful for developing new methods for the analysis of high-density (SNP) markers.

As an extension of the algorithm of O'Connell [37], Zhang et al. [38] developed the computer program HAPLORE, which is based on a generalization of the genetic rules of Wijsman [67] under the assumption of no recombination among markers. HAPLORE consists of three steps: (1) All possible haplotypes are inferred for each individual by use of the genetic rules. (2) Inconsistent haplotypes are deleted with a genotype elimination algorithm as

in O'Connell [37]. (3) Haplotype frequencies are estimated based on the identified configurations with a partition-ligation-expectation-maximization algorithm [73]. HAPLORE is more efficient than ZAPLO, although it is still limited to pedigrees of small to moderate size. Another advantage is that HAPLORE can estimate haplotype frequencies by using pedigree data and data from unrelated individuals together and modeling marker-marker LD.

Minimum-Recombinant Haplotyping Algorithms

The assumption of no recombination among markers in a pedigree in the previous section can be violated even for some dense markers. Haines [6] proposed an algorithm to infer hapotypes in families by minimizing the number of recombinants and applied the algorithm to genotyping error detection. An error was identified if an inferred haplotype included double recombinations in a short region. Based on this work, Qian and Beckmann [69] presented a six-rule algorithm to identify a set of haplotype configurations with the minimum number of recombinants for a pedigree by an approximation that minimizes the number of recombinants in each nuclear family. For each nuclear family in the pedigree, the method assigns ordered genotypes to offspring (parents) conditional on the haplotypes and/or genotypes in parents (offspring) and the criterion of minimum number of recombinants in the family. If the ordered genotypes of some individuals at a locus can not be determined by this process, then method performs exhaustive enumeration of all ordered genotypes of these individuals at the locus. Therefore, it can only analyze pedigrees of small to moderate size with small numbers of markers, which do not have substantial amounts of missing marker data. This method does not use information on distances between markers. Distances between adjacent markers may not be constant; consequently the configuration with minimum number of recombinants may not have the highest likelihood.

To accommodate large pedigrees with substantial amounts of missing data, Li and Jiang [70] extended the method of Qian and Beckmann [69] by using an integer linear programming (ILP) formulation to identify configurations with the minimum number of recombinants in a pedigree. If desired, a configuration with the highest likelihood among these identified configurations can be selected. The ILP method was implemented in the software PedPhase which can also incorporate information on inter-marker distances. Simulation studies of Li and Jiang [70] showed that the ILP method is faster than the software SimWalk2 and generates results comparable to those from the second run of SimWalk2.

Methods Extended from Population-Based Approaches for High-Density (SNP) Markers Using Trio or Nuclear Family Data

As described earlier, the haplotyping methods assuming linkage equilibrium among markers can produce inaccurate results, in particular for tightly linked (SNP) markers in LD [7, 22–24], and use of this haplotype information in linkage or association studies can adversely affect mapping accuracy [7].

To account for LD among tightly linked markers, population-based haplotyping methods using unrelated individuals were developed [10, 11]. Because family data provide substantially more information for inferring haplotypes than samples of unrelated individuals [1], several of these population-based methods were extended to use nuclear families, father-mother-child trios, or sibships [1, 5, 23, 24, 74].

The extended methods using nuclear families or trios [e.g., 1, 5, 23] assume that all parents in the nuclear families or trios are sampled independently from a population in HWE, and they often assume that no recombination occurs in the transmission of haplotypes from the parents to children. These methods infer haplotypes for the independent parents by using the population-based approaches and by excluding the parental haplotype pairs that are not consistent with the children's genotype data. This idea is similar to that of the rule-based method of O'Connell [37]. The extended methods account for LD among markers and can jointly use population data and nuclear family or trio data.

We note that the purpose of these extended methods is to infer haplotypes and estimate haplotype frequencies for parents (founders), rather than to infer haplotype configurations for entire families (or pedigrees) as done by the methods described in previous sections. The extended methods using trios can be applied to inferring haplotypes for genome-wide SNP markers (say 1 million SNPs [1]), whileas the rule-based methods assuming no recombination and using founder population haplotyping frequencies, such as ZAPLO [37] and HAPLORE [38], are more appropriate for tightly linked markers in short chromosomal regions (e.g., candidate genes). This is so because these rule-based methods are designed for larger pedigrees, where haplotyping becomes computationally intensive. Below we review the extended methods.

Maximum likelihood methods implemented via an EM algorithm have been widely used to estimate haplotype frequencies in population data under the assumptions of HWE and random mating [e.g., 75, 76]. Rohde

Table 1. Summary of primary haplotyping methods and related computer programs for pedigrees of small to moderate size (say <40)

No.	Algorithms used	References	Modeling LD (assuming no recombination)	Applications and limitations[1]	Computer programs	Description of identified configuration(s)
1	Elim-mpe, Bayesian network	29	no (no)	small or moderate N_m (with a few markers, pedigree size can be up to a few hundred)	SUPERLINK	highest likelihood
2	Lander-Green	15	no (no)	large N_m	GENEHUNTER	highest likelihood
3	Lander-Green	8	no (no)	very large N_m (thousands)	Merlin	highest likelihood
4	Lander-Green, MTBDD[2]	16, 17	no (no)	very large N_m (thousands)	Allegro2	highest likelihood
5	Lander-Green, EM[3]	22	partly (partly)	very large N_m (thousands)	Merlin	highest likelihood
6	Genetic rules, EM	37	yes (yes)	small or moderate N_m (small region)	ZAPLO	zero recombinants
7	Genetic rules, EM	38	yes (yes)	small or moderate N_m (small region)	HAPLORE	zero recombinants
8	Genetic rules	69	no (no)	small N_m	MRH	minimum recombinants
9	EM	23	yes (yes)	small N_m, nuclear families	available	haplotypes for parents
10	Bayesian approach	5	yes (yes)	very large N_m, nuclear families	–	haplotypes for parents
11	Bayesian approach, perfect phylogeny approach, or EM	1	yes (yes)	very large N_m (thousands), trios	HAP2, PHASE (v2.1), HAP, tripleM	haplotypes for parents

[1] N_m denotes the number of markers which can be analyzed. [2] Multiterminal binary decision diagrams. [3] EM algorithm.

and Fuerst [23] applied a maximum likelihood EM algorithm to haplotyping parents in nuclear family data with dense markers and showed that this method had higher haplotyping accuracy than the software GENEHUNTER [15], which assumes linkage equilibrium among markers. However, this method can only be applied to 30 or fewer biallelic loci [23].

To accommodate a large number of (SNP) markers in nuclear families, Lin et al. [5] proposed a haplotyping method based on a Bayesian MCMC approach incorporating a variant of the partition ligation method of Niu et al. [77]. The method groups (SNP) markers into high LD blocks, reconstructs haplotypes for subgroups of markers within each block, and then reconstructs haplotypes for blocks. It can analyze thousands of dense SNPs and more than 1,000 chromosomes.

The process of using genotype information on children in nuclear families to help reconstructing parental haplotypes in the method of Lin et al. [5] essentially assumes that no recombination occurs in the transmission of haplotypes from the parents to children, but recombination can be accommodated to some extent as follows. For example, if a child has genotype $G_1G_2G_3G_4$ at four loci with a recombination between the 2nd and 3rd markers in one of the child's haplotypes, then the genotype is split into two genotypes G_1G_200 and $00G_3G_4$, where 0 is a missing genotype. This process can not handle the families (rare cases) with multiple children in which every child inherits a recombined haplotype or in which a child inherits two recombined haplotypes.

Marchini et al. [1] described the extension of five of the leading population-based haplotyping methods to use father-mother-child trios. These five extended methods (including the method of Lin et al. [5]) incorporate the partition ligation of Niu et al. [77] and are able to process thousands of high-density markers. The extended methods include Bayesian approaches using coalescent-based models (e.g., the PHASE (v2.1) algorithm) [78, 79], a perfect phylogeny approach using constrained maximum likelihood [80], and a maximum likelihood EM method called tripleM [73]. The coalescent-based models attempt to capture the fact that over short genomic regions, sam-

Table 2. Summary of primary haplotyping methods and related computer programs for large complex pedigrees

No.	Algorithms used	References	Modeling LD	Applications and limitations*	Computer programs	Description of identified configuration(s)
1	Metropolis, simulated annealing	18	no	large N_m (computing time can be substantial)	SimWalk2	highest likelihood
2	Gibbs-Jump	13	no	large N_m (can be slow to converge)	–	highest likelihoods
3	Sequential imputation	20, 60	no	small to moderate N_m (account for genetic interference)	SIMPLE	highest likelihoods
4	LM-sampler	30, 62, 63	no	large N_m	MORGAN, Loki (no output for haplotyping)	samples from posterior distribution
5	Block Gibbs sampler	19	no	large N_m	MCLINK (no output for haplotyping)	samples from posterior distribution
6	Conditional enumeration	14, 21	no	large N_m, very large pedigree size (thousands), current version cannot handle missing data	CeHap	highest likelihoods
7	Integer linear programming, EM	70	yes	large N_m (more appropriate for small region with dense markers)	PedPhase	minimum recombinants

* N_m denotes the number of markers which can be analyzed.

pled chromosomes tend to cluster together into groups of similar haplotypes and the perfect phylogeny approach also accounts for this 'clustering property', while tripleM does not. PHASE (v2.1) can also internally re-estimate a variable population-scaled recombination rate across the region being considered [1].

Marchini et al. [1] comprehensively compared the five extended methods when applied to both trios and unrelated individuals by using data simulated based on the coalescent model as well as data from the HapMap project (http://www.hapmap.org/). All methods provided highly accurate estimates of haplotypes when applied to trio data sets. Overall the PHASE (v2.1) algorithm had the highest accuracy for all data sets considered. Although it is one of the slowest methods, PHASE (v2.1) was used to infer haplotypes for the 1 million-SNP HapMap data set [1, 81].

All methods extended from population-based approaches described above essentially assume that no recombination occurs in the transmission of haplotypes from parents to children in the trio or nuclear family data. This assumption is reasonable for high-density biallelic (SNP) markers in a short chromosomal region (at most several megabases) but it may not be appropriate for a long chromosomal region (tens of centimorgan) in families with multiple children or in pedigrees. Multiple children in a family may provide more information for inferring haplotypes than a single child. In addition, the extended methods cannot deal with (multi-generational) pedigrees. Inferring haplotype configurations in pedigrees by modeling LD and recombinants among markers is useful for fine mapping in linkage analysis and association studies. As stated earlier, the method of Abecasis and Wigginton [22] developed for pedigrees with clustered marker data can account for marker-marker LD within each cluster and recombination between clusters. A possible problem is that ignoring LD among markers from different clusters may generate inaccurate results when analyzing a large number of high-density markers.

Discussion and Future Directions

Haplotyping in pedigrees provides a single or a set of most likely haplotype configurations which are useful for many types of genetic analyses, including linkage analysis and haplotype-based association studies. In the past twenty years, many haplotyping methods for family and pedigree data have been developed. This enhanced our ability to map QTL and complex disease genes in human and animal populations. We have reviewed the haplotyping methods in previous sections. Tables 1 and 2 provide

summaries of the primary haplotyping methods and computer programs reviewed in this article for pedigrees of small to moderate size and for large complex pedigrees, respectively. We hope that this overview enables researchers to quickly identify the haplotyping methods and computer programs most suitable for their research needs.

Most of the haployping methods are based on the assumption of no genotyping errors in the observed pedigree data. Genotyping errors can have a substantial impact on haplotyping and linkage results [e.g., 82]. Several methods have been developed to detect genotyping errors (including Mendelian-consistent and Mendelian-inconsistent errors) and pedigree errors [8, 82, 83]. Some of these methods have been implemented in software packages, which include Simwalk2 and Merlin. These software programs are recommended to detect genotyping errors prior to haplotyping in pedigrees.

There is still a need for further improvements of existing and the development of novel haplotyping methods. A major challenge is to develop efficient methods for identifying a set of haplotype configurations with the highest likelihoods in large complex pedigrees with large numbers of high-density markers and with substantial amounts of missing marker data, while accounting for marker-marker LD. It is far from trivial to infer haplotype configurations for large pedigrees with several ancestral generations completely lacking genotype data followed by recent generations with marker data. An example is the Hutterite pedigree, which includes 1623 members over 13 generations with about 800 individuals in the most recent four generations having genotype data [84].

With the development of improved haplotyping methods for population data [79, 85], it should be worthwhile to extend these population based approaches to accommodate family structure and genome wide marker data by modeling recombination in families and LD in founders.

Lastly, most of the existing haplotyping methods ignore the different genetic maps between males and females and the phenomenon of genetic interference. Accommodating different values for the recombination frequencies of males and females and incorporating genetic interference models will increase the accuracy of haplotyping in pedigrees [20, 30].

Acknowledgements

We thank Kui Zhang for helpful discussions. This research was supported by grant GM073766 from the National Institute of General Medical Sciences and supported in part by grants ES09912, CA100949, and P30DK056336 from the National Institutes of Health. Partial support from the Virginia Bioinformatics Institute is also appreciated. We thank the editors and reviewers for their constructive suggestions that improved our manuscript.

Appendix I: Viterbi algorithm

Rabiner [31] provided a detailed description of the Viterbi algorithm. Here we present the Viterbi algorithm in the format of the elim-mpe algorithm as follows. Let $\mathbf{Q}^0 = (q_1^0, q_2^0, ..., q_T^0)$ denote the best state sequence such that $P(\mathbf{Q}^0, \mathbf{O})$ is maximized. The maximum value of $P(\mathbf{Q}, \mathbf{O})$ can be calculated as

$$
\begin{aligned}
&\max_{q_1,q_2,\ldots,q_T} P(\mathbf{Q},\mathbf{O}) \\
&= max_{q_2,\ldots,q_T}\, max_{q_1}\, P(q_1)P(O_1 \mid q_1)P(q_2 \mid q_1)P(O_2 \mid q_2)\cdots \\
&\qquad P(q_T \mid q_{T-1})P(O_T \mid q_T) \\
&= max_{q_2,\ldots,q_T}\, max_{q_1}\left[P(q_1)P(O_1 \mid q_1)P(q_2 \mid q_1)\right] \\
&\qquad \cdot \prod_{j=2}^{T-1}\left[P(O_j \mid q_j)P(q_{j+1} \mid q_j)\right]\cdot P(O_T \mid q_T) \\
&= max_{q_2,\ldots,q_T}\, max_{q_1}\, f_1(q_1,q_2) \\
&\qquad \cdot \prod_{j=2}^{T-1}\left[P(O_j \mid q_j)P(q_{j+1} \mid q_j)\right]\cdot P(O_T \mid q_T) \\
&= max_{q_2,\ldots,q_T}\, h_1(q_2)\cdot \prod_{j=2}^{T-1}\left[P(O_j \mid q_j)P(q_{j+1} \mid q_j)\right]\cdot P(O_T \mid q_T) \\
&= max_{q_3,\ldots,q_T}\, max_{q_2}\left[h_1(q_2)P(O_2 \mid q_2)P(q_3 \mid q_2)\right] \\
&\qquad \cdot \prod_{j=3}^{T-1}\left[P(O_j \mid q_j)P(q_{j+1} \mid q_j)\right]\cdot P(O_T \mid q_T) \\
&\cdots \\
&= max_{q_T}\, h_{T-1}(q_T)P(O_T \mid q_T) \\
&= max_{q_T}\, f_T(q_T)
\end{aligned}
\qquad \text{(A1)}
$$

where, for example, $f_1(q_1,q_2) = P(q_1)P(O_1 \mid q_1)P(q_2 \mid q_1)$, which is the product of all factors that depend on q_1, and $h_1(q_2) = \max_{q_1} f_1(q_1,q_2)$ is the maximum value of $f_1(q_1,q_2)$ over q_1.

The first seven lines in equation (A1) show the maximization over q_1 by maximizing $f_1(q_1,q_2)$. For each specific value of q_2 the Viterbi algorithm stores the corresponding maximum value $h_1(q_2)$ and $\psi_1(q_2) = \arg\max_{q_1} f_1(q_1,q_2)$, which is the value of q_1 maximizing $f_1(q_1,q_2)$. Similarly in the following lines in equation (A1), maximization over q_2 is conducted by maximizing $f_2(q_2,q_3) = h_1(q_2)P(O_2 \mid q_2)P(q_3 \mid q_2)$. For each specific value of q_3 the algorithm stores $h_2(q_3) = \max_{q_2} f_2(q_2,q_3)$ and $\psi_2(q_3) = \arg\max_{q_2} f_2(q_2,q_3)$. This process continues by performing a maximization over each of remaining q_t of the state sequence $\mathbf{Q}$ in the order $(q_1,q_2, ..., q_T)$. Then an optimal value is assigned to each element q_t in the reverse order to obtain $\mathbf{Q}^0 = (q_1^0,q_2^0, ..., q_T^0)$ based on the stored values for each element q_t as follows.

First, find $q_T^0 = \arg\max_{q_T} f_T(q_T)$,
second, find $q_{T-1}^0 = \psi_{T-1}(q_T^0) = \arg\max_{q_{T-1}} f_{T-1}(q_{T-1},q_T^0)$, and so on,
and lastly find $q_1^0 = \psi_1(q_2^0)$.

Appendix II: Elim-mpe algorithm

In a Bayesian network, for the set of variables $\mathbf{X} = (X_1, ..., X_m)$ with the subset of unknown variables $\mathbf{X}^* = (X_1^*, ..., X_n^*)$, let $\mathbf{X}_{-n,ch}$ denote the set of remaining variables in $\mathbf{X}$ after removing X_n^* and its child variables $ch(X_n^*)$. For the given ordering $X_1, ..., X_m$, the maximum value of $P(\mathbf{X})$ over all unknown variables $(X_1^*, ..., X_n^*)$ can be calculated as

$$\begin{aligned}
&max_{X_1^*,\ldots,X_n^*} P(\mathbf{X}) = max_{X_1^*,\ldots,X_{n-1}^*} max_{X_n^*} \prod_{i=1}^{m} P\left(X_i \mid pa\left(X_i\right)\right) \\
&= max_{X_1^*,\ldots,X_{n-1}^*} \left[\prod_{X_i \in \mathbf{X}_{-n,ch}} P\left(X_i \mid pa\left(X_i\right)\right) \cdot max_{X_n^*} P\left(X_n^* \mid pa\left(X_n^*\right)\right) \right. \\
&\qquad \left. \prod_{X_i \in ch\left(X_n^*\right)} P\left(X_i \mid pa\left(X_i\right)\right) \right] \\
&= max_{X_1^*,\ldots,X_{n-1}^*} \left[\prod_{X_i \in \mathbf{X}_{-n,ch}} P\left(X_i \mid pa\left(X_i\right)\right) \cdot max_{X_n^*} f\left(X_n^*, x_n^1, \cdots, x_n^{k_n}\right) \right] \\
&= max_{X_1^*,\ldots,X_{n-1}^*} \left[\prod_{X_i \in \mathbf{X}_{-n,ch}} P\left(X_i \mid pa\left(X_i\right)\right) \cdot h_n\left(x_n^1, \cdots, x_n^{k_n}\right) \right], \qquad \text{(A2)}
\end{aligned}$$

where

$$f\left(X_n^*, x_n^1, \cdots, x_n^{k_n}\right) = P\left(X_n^* \mid pa\left(X_n^*\right)\right) \prod_{X_i \in ch\left(X_n^*\right)} P\left(X_i \mid pa\left(X_i\right)\right)$$

is the product of all factors that depend on X_n^*, and $h_n(x_n^1, ..., x_n^{k_n}) = \max_{x_n^*} f(X_n^*, x_n^1, ..., x_n^{k_n})$ is the maximum values of the $f(\cdot)$ function over X_n. Variables $(x_n^1, ..., x_n^{k_n})$ are a subset of $(X_1^*, ..., X_{n-1}^*)$ i.e., $(x_n^1, ..., x_n^{k_n}) \subseteq (X_1^*, ..., X_{n-1}^*)$. For each assignment of $(x_n^1, ..., x_n^{k_n})$, the value of $h_n(x_n^1, ..., x_n^{k_n})$ and the value of X_n^* maximizing $f(X_n^*, x_n^1, ..., x_n^{k_n})$, or X_n^{*0} $(x_n^1, ..., x_n^{k_n}) = \arg\max_{x_n} f(X_n^*, x_n^1, ..., x_n^{k_n})$ are stored, and then the variable X_n^* is eliminated from the variables for maximization. Similarly, in the following step, the $f(\cdot)$ and $h(\cdot)$ functions for X_{n-1}^* are calculated, and so on. This process is repeated until all variables are eliminated in the order $X_n^*, ..., X_1^*$, and then optimal values are assigned to the variables in the reverse order to obtain $\mathbf{X}^{*0} = (X_1^{*0}, ..., X_n^{*0})$ based on the stored values for each variable, where $P(\mathbf{X})$ is maximized at $\mathbf{X}^* = \mathbf{X}^{*0}$. In this process, the $f(\cdot)$ function for variable X_i^* $(1 \leq i < n)$ is the product of all remaining factors (including some $h(\cdot)$ functions) depending on X_i^* in equation (A2) after $X_{i+1}^*, ..., X_n^*$ have been eliminated.

References

1 Marchini J, Cutler D, Patterson N, Stephens M, Eskin E, Halperin E, Lin S, Qin ZS, Munro HM, Abecasis GR, Donnelly PA: Comparison of phasing algorithms for trios and unrelated individuals. Am J Hum Genet 2006; 78:437–450.

2 Sobel E, Lange K, O'Connell JR, Weeks DE: Haplotyping algorithm; in Speed T, Waterman MS (eds): IMA volumes in mathematics and its applications. Genetic mapping and DNA sequencing. New York, Springer-Verlag, 1996, Vol 81, pp 89–110.

3 Akey J, Jin L, Xong M: Haplotypes vs. single marker linkage disequilibrium tests: What do we gain? Eur J Hum Genet 2001;9:291–300.

4 Hugot JP, Chamaillard M, Zuoali H, Lesage S, Cezard JP, Belaiche J, Almer S, Tysk C, O'Morain CA, Gassull M, Binder V, Finkel Y, Cortot A, Modigliani R, Laurent-Puig P, Gower-Rousseau C, Macry J, Colombel JF, Sahbatou M, Thomas G: Association of NOD2 leucine-rich repeat variants with susceptibility to Crohn's disease. Nature 2001; 411:599–603.

5 Lin S, Chakravarti A, Cutler DJ: Haplotype and missing data inference in nuclear familes. Genome Res 2004;14:1624–1632.

6 Haines JL: Chromlook: an interactive program for error detection and mapping in reference linkage data. Genomics 1992;14:517–519.

7 Schaid DJ, McDonnell SK, Wang L, Cunningham JM, Thibodeau SN: Caution on pedigree haplotype inference with software that assumes linkage equilibrium. Am J Hum Genet 2002;71:992–995.

8 Abecasis GR, Cherny SS, Cookson WO, Cardon LR: Merlin-rapid analysis of dense genetic maps using sparse gene flow trees. Nat Genet 2002;30:97–101.

9 Burdick JT, Chen WM, Abecasis GR, Cheung VG: In silico method for inferring genotypes in pedigrees. Nat Genet 2006;38:1002–1004.

10 Niu T: Algorithms for inferring haplotypes. Genet Epidemiol 2004;27:334–347.

11 Salem RM, Wessel J, Schork NJ: A comprehensive literature review of haplotyping software and methods for use with unrelated individuals. Hum Genomics 2005;2:39–66.

12 Clerget-Darpoux F, Elston RC: Are linkage analysis and the collection of family data dead? Prospects for family studies in the age of genome-wide association. Hum Hered 2007;64:91–96.

13 Lin S, Speed TP: An algorithm for haplotype analysis. J Comput Biol 1997;4:535–546.

14 Gao G, Hoeschele I, Sorensen P, Du FX: Conditional probability methods for haplotyping in pedigrees. Genetics 2004;167:2055–2065.

15 Kruglyak L, Daly MJ, Reeve-Daly MP, Lander ES: Parametric and nonparametric linkage analysis: a unified multipoint approach. Am J Hum Genet 1996;58:1347–1363.

16 Gudbjartsson DF, Jonasson K, Frigge ML, Kong A: Allegro, a new computer program for multipoint linkage analysis. Nat Genet 2000;25:12–13.

17 Gudbjartsson DF, Thorvaldsson T, Kong A, Gunnarsson G, Ingolfsdottir A: Allegro version 2. Nat Genet 2005;37:1015–1016.

18 Sobel E, Lange K: Descent graphs in pedigree analysis: applications to haplotyping, location scores, and marker sharing statistics. Am J Hum Genet 1996;58:1323–1337.

19 Thomas A, Gutin A, Abkevich V, Bansal A: Multilocus linkage analysis by blocked Gibbs sampling. Stat Comput 2000;10:259–269.

20 Lin S, Skrivanek Z, Irwin M: Haplotyping using SIMPLE: caution on ignoring interference. Genet Epidemiol 2003;25:384–387.

21 Gao G, Hoeschele I: A rapid conditional enumeration haplotyping method in pedigrees. Genet Sel Evol 2008;40:25–36.

22 Abecasis GR, Wigginton JE: Handling marker-marker linkage disequilibrium: pedigree analysis with clustered markers. Am J Hum Genet 2005;77:754–767.

23 Rohde K, Fuerst R: Haplotyping and estimation of haplotype frequencies for closely linked biallelic multilocus genetic phenotypes including nuclear family information. Hum Mutation 2001;17:289–295.

24 Ding X, Zhang Q, Flury C, Simianer H: Haplotype reconstruction and estimation of haplotype frequencies from nuclear families with only one parent available. Hum Hered 2006;62:12–19.

25 Lindholm E, Zhang J, Hodge SE Greenberg DA: The reliability of haplotyping inference in nuclear families: Misassignment rates for SNPs and microsatellites. Hum Hered 2004; 57:117–127.
26 Gao G, Hoeschele I: Approximating identity-by-descent matrices using multiple haplotype configurations on pedigrees. Genetics 2005;171:365–376.
27 Haldane JBS: The combination of linkage values and the calculation of distances between the loci of linked factors. J Genet 1919; 8:299–309.
28 Lander ES, Green P: Construction of multilocus genetic linkage maps in humans. Proc Natl Acad Sci USA 1987;84:2363–2367.
29 Fishelson M, Dovgolevsky N, Geiger D: Maximum likelihood haplotyping for general pedigrees. Hum Hered 2005;59:41–60.
30 Thompson EA: Statistical inference from genetic data on pedigrees. NSF-CBMS Regional Conference Series in Probability and Statistics, Vol 6. Beachwood, OH: Institute of Mathematical Statistics, 2000.
31 Rabiner LR: A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE 1989;77:257–286.
32 Baum LE: An inequality and associated maximization technique in statistical estimation for probabilistic functions on Markov processes; in Shisha O (ed): Inequalities-III; Proceedings of the Third Symposium on Inequalities. University of California Los Angeles, 1969, Academic Press, New York, 1972, pp 1–8.
33 Fujita M, McGeer PC, Yang JC: Multi-terminal binary decision diagrams: An efficient data structure for matrix representation. Formal methods in system design 1997;10: 149–169.
34 Ingolfsdottir A, Gudbjartsson D: Genetic linkage analysis algorithms and their implementation; in Priami C et al (ed): Transact on Computat Systems Biol III, LNBI 3737. Berlin/Heidelberg, Springer-Verlag, 2005, pp 123–144.
35 Lafferty J, Vardy A: Ordered Binary decision diagrams and minimal trellises. IEEE Trans Computers 1999;48:971–986.
36 Hermanns H, Meyer-Kayser J, Siegle M: Multi terminal binary decision diagrams to represent and analyse continuous time Markov chains. In 3rd Int. Workshop on the Numerical Solution of Markov Chains. Prensas Univesitaris de Zaragoza, 1999, pp 188–207.
37 O'Connell JR: Zero-recombinant haplotyping: applications to fine mapping using SNPs. Genet Epidemiol 2000;19(suppl 1): S64–S70.
38 Zhang K, Sun F, Zhao H: HAPLORE: a program for haplotype reconstruction in general pedigrees without recombination. Bioinformatics 2005;21:90–103.
39 Elston RC, Stewart J: A general model for the genetic analysis of pedigree data. Hum Hered 1971;21:523–542.
40 Ott J: Analysis of human genetic linkage. Baltimore and London. The Johns Hopkins University Press, 1999.
41 Cannings C, Thompson E, Skolnick M: Probability functions on complex pedigrees. Adv Appl Prob 1978;10:26–61.
42 Lange K, Elston RC: Extensions to pedigree analysis. I. Likelihood calculations for simple and complex pedigrees. Hum Hered 1975;25:95–105.
43 Lange K, Boehnke M: Extensions to pedigree analysis. V. Optimal calculation of Mendelian likelihoods. Hum Hered 1983;33:291–301.
44 Ploughman LM, Boehnke M: Estimating the power of a proposed linkage study for a complex genetic trait. Am J Hum Genet 1989;44: 543–551.
45 Janss LLG, Van Arendonk JAM, Van der Werf JHJ: Computing approximate monogenic model likelihoods in large pedigrees with loops. Genet Sel Evol 1995;27:567–579.
46 Wang T, Fernando RL, Stricker C, Elston RC: An approximation to the likelihood for a pedigree with loops. Theor Appl Genet 1996; 93:1299–1309.
47 Fernandez SA, Fernando RL, Guldbrandtsen B, Totir LR, Carriquiry AL: Sampling genotypes in large pedigrees with loops. Genet Sel Evol 2001;33:337–367.
48 Lauritzen SL, Spiegelhalter DL: Local computations with probabilities on graphical structures and their application to expert systems. J R Stat Soc B 1988;50:157–224.
49 Lauritzen SL, Sheehan NA: Graphical models for genetic analyses. Stat Sci 2003;18:489–514.
50 Heath SC: Genetic linkage analysis using Markov chain Monte Carlo techniques; in Green PJ, Hjort NL, Richardson S (ed): Highly Structured Stochastic System. London/New York/Oxford, Oxford University Press, 2003, pp 363–381.
51 Fernandez SA, Fernando RL: Determining peeling order using sparse matrix algorithms. J Dairy Sci 2002;85:1623–1629.
52 Pearl J: Probabilistic Reasoning in Intelligent Systems. San Francisco, Morgan Kaufmann, 1988.
53 Lauritzen SL: Graphical Models. Oxford University Press, 1996.
54 Dechter R: Bucket elimination: a unifying framework for probabilistic inference; in J M I (ed): Learning in Graphical Models. Kluwer Academic Press, 1998, pp 75–104.
55 Murphy KP: A brief introduction to graphical models and Bayesian networks. Http://www.cs.ubc.ca/~murphyk/Bayes/bayes-tutorial.pdf. 2001.
56 O'Connell JR, Weeks DE: An optimal algorithm for automatic genotype elimination. Am J Hum Genet 1999;65:1733–1740.
57 Lange K, Goradia TM: An algorithm for automatic genotype elimination. Am J Hum Genet 1987;40:250–256.
58 O'Connell JR, Weeks DE: The VITESSE algorithm for rapid exact multilocus linkage analysis via genotype set-recoding and fuzzy inheritance. Nat Genet 1995;11:402–408.
59 Jensen CS, Kjaerulff U, Kong A: Blocking Gibbs sampling in very large probabilistic expert systems. Int J Hum Comp Stud 1995; 42:647–666.
60 Irwin M, Cox N, Kong A: Sequential imputation for multilocus linkage analysis. Proc Natl Acad Sci USA 1994;91:11684–11688.
61 Skrivanek Z, Lin S, Irwin M: Linkage analysis with sequential imputation. Genet Epidemiol 2003;25:25–35.
62 Heath SC: Markov chain Monte Carlo segregation and linkage analysis for oligogenic models. Am J Hum Genet 1997;61:8748–760.
63 Thompson EA, Heath SC: Estimation of conditional multilocus gene identity among relatives; in Seillier-Moiseiwitsch F (ed): Statistics in Molecular Biology and Genetics: Selected Proceedings of a 1997 Joint AMS-IMS-SIAM Summer Conference on Statistics in Molecular Biology, IMS Lecture Note-Monograph Series Volume 33, Hayward, CA: Institute of Mathematical Statistics, 1999, pp 95–113.
64 Heath SC, Thompson EA: MCMC samplers for multilocus analyses on complex pedigrees. Am J Hum Genet 1997;61:A278.
65 Broman KW, Weber JL: Characterization of human crossover interference. Am J Hum Genet 2000;66:1911–1926.
66 Lin S, Cheng R, Wright FA: Genetic crossover interference in the human genome. Ann Hum Genet 2001;65:79–93.
67 Wijsman E: A deductive method of haplotype analysis in pedigrees. Am J Hum Genet 1987;41:356–373.
68 Tapadar P, Ghosh S, Majumder PP: Haplotyping in pedigrees via a genetic algorithm. Hum Hered 2000;50:43–56.
69 Qian D, Beckmann L: Minimum-recombinant haplotyping in pedigrees. Am J Hum Genet 2002;70:1434–1445.
70 Li J, Jiang T: Computing the minimum recombinant haplotype configuration from incomplete genotype data on a pedigree by integer linear programming. J Comp Biol 2005;12:719–739. [http://www.cs.ucr.edu/~jili/haplotyping.html]
71 Baruch E, Weller JI, Cohen M, Ron M, Seroussi E: Efficient inference of haplotypes from genotypes on a large animal pedigree. Genetics 2006;172:1757–1765.
72 Dempster A, Laird N, Rubin D: Maximum likelihood from incomplete data via the E-M algorithm. J R Stat Soc Ser B 1977;39:1–38.
73 Qin ZS, Niu T, Liu JS: Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. Am J Hum Genet 2002;71: 1242–1247.

74 Liu PY, Lu Y, Deng HW: Accurate haplotype inference for multiple linked single-nucleotide polymorphisms using sibship data. Genetics 2006;174:499–509.
75 Excoffier L, Slatkin M: Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. Mol Biol Evol 1995;12:921–927.
76 Hawley ME, Kidd KK: HAPLO: a program using the EM algorithm to estimate frequencies of multi-site haplotypes. J Hered 1995; 86:409–411.
77 Niu T, Qin ZS, Xu X, Liu JS: Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. Am J Hum Genet 2002;70:157–169.
78 Stephens M, Smith NJ, Donnelly P: A new statistical method for haplotype reconstruction from population data. Am J Hum Genet 2001;68:978–989.
79 Stephens M, Scheet P: Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. Am J Hum Genet 2005;76:449–462.
80 Halperin E, Eskin E: Haplotype reconstruction from genotype data using imperfect phylogeny. Bioinformatics 2004;20:1842–1849.
81 International HapMap Consortium: A haplotype map of the human genome. Nature 2005;437:1299–1320.
82 Sobel E, Papp JC, Lang K: Detection and integration of genotyping errors in statistical genetics. Am J Hum Genet 2002;70:496–508.
83 Douglas JA Skol AD, Boehnke M: Probability of detection of genotyping errors and mutations as inheritance inconsistencies in nuclear-family data. Am J Hum Genet 2002;70: 487–495.
84 Weiss LA, Pan L, Abeny M, Ober C: The sex-specific genetic architecture of quantitative traits in humans. Nat Genet 2006;38:218–222.
85 Zhang Y, Niu T, Liu JS: A coalescence-guided hierarchical Bayesian method for haplotype inference. Am J Hum Genet 2006;79:313–22.