

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is an author's version which may differ from the publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/34702>

Please be advised that this information was generated on 2019-02-17 and may be subject to change.

Haplotype Inference in General Pedigrees using the Cluster Variation Method

Cornelis A Albers^{†1} Tom Heskes[‡] Hilbert J Kappen[†]

[†] Department of Cognitive Neuroscience/Biophysics
Institute for Computing and Information Sciences
Radboud University, Nijmegen, The Netherlands, 6525 EZ
[‡] Department of Information and Knowledge Systems
Institute for Computing and Information Sciences
Radboud University, Nijmegen, The Netherlands, 6525 ED

¹ Geert Grooteplein 21/126, 6525 EZ Nijmegen, The Netherlands

Running head: Haplotype Inference

Keywords: Haplotype Inference, Pedigree, the Cluster Variation Method

Corresponding author:

Cornelis A Albers

Department of Cognitive Neuroscience/Biophysics/126, Institute for Computing and Information Sciences, Radboud University

Postal address: Geert Grooteplein 21, 6525 EZ Nijmegen, The Netherlands

Telephone: +31-24-361-4230

Fax: +31-24-354-1435

E-mail: k.albers@science.ru.nl

Abstract

We present CVMHAPLO, a probabilistic method for haplotyping in general pedigrees with many markers. CVMHAPLO reconstructs the haplotypes by assigning in every iteration a fixed number of the ordered genotypes with the highest marginal probability, conditioned on the marker data and ordered genotypes assigned in previous iterations. CVMHAPLO makes use of the Cluster Variation Method (CVM) to efficiently estimate the marginal probabilities. In simulated data sets where exact computation was feasible, we found that the accuracy of CVMHAPLO was high and similar to that of maximum likelihood methods. In simulated data sets where exact computation of the maximum likelihood haplotype configuration was not feasible, the accuracy of CVMHAPLO was similar to that of state of the art Markov chain Monte Carlo (MCMC) maximum likelihood approximations when all ordered genotypes were assigned, and higher when only a subset of the ordered genotypes was assigned. CVMHAPLO was faster than the MCMC approach and provided more detailed information about the uncertainty in the inferred haplotypes. We conclude that CVMHAPLO is a practical tool for the inference of haplotypes in large complex pedigrees.

INTRODUCTION

The problem of haplotyping is to infer for each individual the paternally inherited alleles and the maternally inherited alleles from the unordered genotype data. Haplotyping is an important tool for mapping disease-susceptibility genes, especially of complex diseases. It is an essential step in the analyses used for the mapping of quantitative trait loci (QTL) in animal pedigrees. As genotyping methods become increasingly cheaper, efficient and accurate algorithms for inferring haplotypes are desirable.

Since the marker data are generally not informative enough to unambiguously infer the ordered genotypes, a probabilistic modelling approach can be used to deal with the uncertainties. The computer programs MERLIN (Abecasis et al., 2002), GENEHUNTER (Kruglyak et al., 1996) and SUPERLINK (Fishelson and Geiger, 2002; Fishelson et al., 2005) reconstruct exact maximum likelihood haplotype configurations in general pedigrees. Due to the exponential increase of computation time and memory usage with pedigree size (MERLIN, GENEHUNTER) or the tree-width of the graphical model associated with the likelihood function (SUPERLINK), application of these programs to large pedigrees and many markers typical of QTL-mapping studies may not be feasible, especially when some of the individuals have missing genotypes or no genotype information at all. Approximate statistical approaches based on MCMC sampling such as SIMWALK2 (Lange and Sobel, 1996) use the same likelihood function as the exact probabilistic approaches and consequently may achieve very high accuracy. MCMC methods can be generally applied and have modest memory requirements. Although in theory computation time does not scale exponentially with the problem size, in practice it can be very long and convergence

of the Markov chain can be difficult to assess. An efficient statistical approach based on a heuristic approximation of conditional probabilities was proposed by Gao et al. (2004), however it has been tested only on data sets with no missing genotypes.

To overcome problems of efficiency several non-statistical approaches have been developed. Wijnsman (1987) proposed a zero-recombinant haplotyping method that is linear in the number of markers and individuals. Recently, efficient algorithms were described by Zhang et al. (2005); Baruch et al. (2005). Application of these approaches is limited to data sets without forced recombination events. Qian and Beckmann (2002) presented a six-rule algorithm to reconstruct minimum recombinant haplotypes. Since computation time is quadratic in pedigree size and cubic in the number of markers, application to large data sets may not be practical. Li and Jiang (2004) proposed an expectation maximization (EM) approach that approximately minimizes the number of recombination events. They also proposed an integer linear programming approach that minimizes the number of recombination events. Although computation time of the latter scales linearly with the rate of missing genotypes, it scales exponentially with the number of individuals. Windig and Meuwissen (2004) described an efficient haplotype reconstruction algorithm for general pedigrees. In spite of the improved efficiency of these methods, imputation of missing genotypes can be problematic due to the lack of a statistical treatment of missing data.

We present a statistical approach, implemented in the computer program CVMHAPLO, that combines the general applicability and accuracy of MCMC approaches with high efficiency. Our haplotype inference algorithm is an iterative procedure where each iteration consists of the following two operations:

1. Estimation of the marginal probabilities of all *unassigned* ordered genotypes conditioned on the *assigned* ordered genotypes and the marker data;
2. assignment of a number of the ordered genotypes with the highest conditional marginal probabilities.

Like SIMWALK2, it can be applied to any pedigree and any number of markers. It provides detailed information about the uncertainty in the inferred haplotypes. Computation time of CVMHAPLO scales approximately linearly with the number of markers and individuals.

We use the Cluster Variation Method (CVM) (Kikuchi, 1951; Morita, 1990; Yedidia et al., 2005) to approximately compute the marginal probability distributions of ordered genotypes. The CVM is a variational approximation designed for efficient estimation of marginal probabilities in complex probability models for which exact computation is not feasible. The CVM estimates marginal probabilities by optimizing marginal distributions on overlapping subsets of variables for which exact probability calculus is feasible. The Cluster Variation Method is closely related to the Belief Propagation algorithm (Pearl, 1988; Yedidia et al., 2005) and has recently gained a lot of interest in the machine learning and computer science community. It has been successfully applied in computer vision

(Weiss, 1997), channel decoding (McEliece et al., 1998) and parametric linkage analysis on extended pedigrees (Albers et al., 2006).

We evaluate our approach in simulated and real data sets. We restrict the evaluation to SNPs and discuss extension to markers with more than two alleles. We compare CVMHAPLO with exact maximum likelihood approaches and the state of the art MCMC maximum likelihood approximation of SIMWALK2.

METHODS

Notation and definitions: We explain the notation that we use with the small pedigree example in fig. 1. For each person i and marker l there is a pair of ordered genotype variables $\{G_i^{l,p}, G_i^{l,m}\}$, which are the paternal and maternal alleles. For each non-founder i in the pedigree and each marker l there are the paternal and maternal segregation indicators $\{v_i^{l,p}, v_i^{l,m}\}$. The founder and non-founder individuals are denoted by F and NF, respectively. We denote the vector of all ordered genotype variables by \mathbf{G} , and the vector of all segregation indicators by \mathbf{v} . Both \mathbf{G} and \mathbf{v} are unobserved experimentally. Instead, the observed genotypes consist of *unordered* pairs of alleles M_i^l for a subset of persons and markers. We denote by \mathbf{M} the vector of all observed allele pairs. The marker map is assumed to be known; the recombination frequency between marker l and $l-1$ is denoted by $\theta_{l,l-1}$ and \mathbf{m}^l denotes the prior allele frequencies for marker l .

Given the marker data \mathbf{M} , one can compute the probability distribution over the ordered genotype variables \mathbf{G} (and the segregation indicators \mathbf{v}). If the pedigree and the number of markers is large, such a computation is intractable and cannot be done in a practical amount of time. When we can perform an exact computation, we will denote the resulting marginal probabilities as $P(\cdot|\cdot)$ and when we are not specific about whether the marginals are exact or approximate, we will denote the resulting marginal probabilities as $Q(\cdot|\cdot)$.

The algorithm CVMHAPLO: Algorithm 1 shows CVMHAPLO in pseudo-code. The ordered genotypes and segregation indicators are assigned to a specific value in a number of iterations, labeled by n . Lines 1–3 represent the initialization of the algorithm. Lines 4–17 represent the iterative assignment procedure.

In iteration n of the algorithm, we use the CVM to compute the approximate marginal probability of all unassigned ordered genotypes, conditioned on all observed genotypes \mathbf{M} and conditioned on all ordered genotypes that have been assigned in all previous iterations, which we denote by $\mathbf{G}_{\text{assigned}}^{(n-1)}$. The resulting conditional probability is denoted by $Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)})$ (line 5 in algorithm 1). This is the computationally intensive step of the algorithm. It can be either performed exactly or approximately using the CVM. The latter approach is explained in the appendix.

Subsequently, a number of ordered genotypes will be assigned. All ordered genotypes for which $Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}) = 1$ for some value of $G_i^{l,p}, G_i^{l,m}$ will be assigned, as well as an additional number of ordered genotypes in the following way. For each marker and person we compute

$$q_{\text{map}}^{i,l} = \max_{G_i^{l,p}, G_i^{l,m}} Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)})$$

and record the ordered genotype that yields the maximum (line 7 in algorithm 1). This is a trivial computation. For instance, if

$$\begin{aligned} Q(G_1^{1,p} = A, G_1^{1,m} = A | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}) &= 0.9, \\ Q(G_1^{1,p} = A, G_1^{1,m} = B | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}) &= 0.1, \\ Q(G_1^{1,p} = B, G_1^{1,m} = A | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}) &= 0.0, \\ Q(G_1^{1,p} = B, G_1^{1,m} = B | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}) &= 0.0, \end{aligned}$$

then $q_{\text{map}}^{1,1} = 0.9$ and it is obtained when $\{G_1^{1,p}, G_1^{1,m}\}_{\text{map}} = \{A, A\}$. We sort the $q_{\text{map}}^{i,l}$ in descending order (line 8) and select the pNL ordered genotypes with the highest value (line 9)¹.

In line 6 the partial haplotype configuration $\mathbf{G}_{\text{assigned}}^{(n-1)}$ of the previous iteration is checked for consistency as described in the appendix. The consistency check verifies that the partial haplotype configuration has a non-zero likelihood under the probabilistic model. When an inconsistency is detected, it is assumed that too many ordered genotypes have been assigned per iteration, and the algorithm is re-initialized in lines 12–14 with a lower value of p .

In line 10, $\mathbf{G}_{\text{assigned}}^{(n)}$ is updated so that it contains all assigned ordered genotypes. The procedure of estimating marginal distributions and assigning ordered genotypes is repeated either until all ordered genotypes have been assigned or until a stopping criterion has been reached.

Confidence in the assignment: When there are many missing values, there is a large uncertainty about the value of the ordered genotypes. In this case, maybe some ordered genotypes can be assigned with a relatively high confidence but others not. This is signalled by the conditional marginal probabilities computed above. For instance, in the above example it is clear that if the four probabilities are all 0.25, no reliable assignment can be made. In this case, it is clear that a full reconstruction of all ordered genotypes is likely to produce many errors and it is important to monitor the quality of the iterative assignment procedure. We suggest to use the values of the $q_{\text{map}}^{i,l}$ as an indication of the

¹ N denotes the number of individuals in the pedigree and L the number of markers and p is a percentage that is specified by the user.

reliability of the assignment procedure, in the following way. Denote by $\{i, l\}$ the set of all ordered genotypes that have been assigned up to iteration n , and $q_{\text{map}}^{i, l, n_{i, l}}$ is the probability of the assignment at the time that was made. We define the confidence in the total assignment up to iteration n as the average of these assignment probabilities:

$$\text{Confidence}(n) = 100\% \times \frac{1}{|\{i, l\}|} \sum_{\{i, l\}} q_{\text{map}}^{i, l, n_{i, l}}, \quad (1)$$

We demonstrate numerically, that this confidence measure is a good indicator of the accuracy of the assigned ordered genotypes. Therefore, one can use this measure to monitor the quality of the assignment procedure and stop when it reaches a prespecified value.

Application of the Cluster Variation Method: Exact inference of marginal probabilities requires a summation over an exponential number of configurations of ordered genotypes and segregation indicators compatible with the marker data, which may not be feasible in practice for complex pedigrees and a large number of markers. The idea of the Cluster Variation Method is to avoid the exponential sum by optimizing marginal distributions of overlapping subsets of variables, i.e. the clusters. The subsets of variables must be chosen such that exact probability calculus on the corresponding cluster marginal distributions $Q_\alpha(\mathbf{x}_\alpha|\cdot)$, where α labels a cluster, is feasible. In essence, the CVM exactly models correlations between variables that are contained in the same cluster and approximates correlations between variables that are contained in different clusters. In the appendix we provide mathematical details of the CVM; here we will focus on the practical aspects of applying the CVM.

Obtaining approximations of the marginal distributions of the ordered genotypes with the CVM proceeds along the following lines. First the probabilistic model must be defined. We make use of the standard pedigree likelihood assuming Hardy-Weinberg equilibrium and linkage equilibrium; the specific form of the distribution is given in the appendix. As a preprocessing step we eliminate a number of symmetries from the model, such as the unknown phase in the ordered genotypes of the founders (see appendix for details). Second, the CVM requires specification of the set of clusters $B = \{\alpha_1, \alpha_2, \dots\}$ that determines the approximation. Below we describe our choice of clusters for the problem of haplotype inference; this is the default cluster choice of CVMHAPLO.

Third, given the set of clusters and the probabilistic model, the Cluster Variation Method prescribes that the so called *free energy* $F_{\text{CVM}}(\{\tilde{Q}_\alpha\})$ must be minimized with respect to the cluster marginal distributions to obtain the optimal approximation, i.e. $\{Q_\alpha\} = \arg \min_{\{\tilde{Q}_\alpha\}} F_{\text{CVM}}(\{\tilde{Q}_\alpha\})$. The minimization must be performed under the constraint that the clusters have identical marginal distributions on variables that are contained in more than one cluster. The CVM does not prescribe how this minimization must be performed, it only provides the analytic form of the functional $F_{\text{CVM}}(\{\tilde{Q}_\alpha\})$ in terms

of the marginal distributions $\{\tilde{Q}_\alpha\}$, the parameters of the probabilistic model, the marker data and the assigned ordered genotypes. The assumption is that the specific form, which is given in the appendix, yields accurate approximations. We apply the provably convergent double loop algorithm described by Heskes et al. (2003) to perform the numerical minimization of the CVM free energy.

Finally, after the numerical minimization procedure has converged, the marginal distribution of an ordered genotype can be obtained by straightforward marginalization of the marginal probability distribution of one of the clusters, e.g. α , that contains the ordered genotype of interest:

$$Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}) = \sum_{\mathbf{x}_\alpha \setminus \{G_i^{l,p}, G_i^{l,m}\}} Q_\alpha(\mathbf{x}_\alpha | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}) \approx P(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}).$$

Specification of the clusters for CVMHAPLO: For the purpose of haplotype inference we have chosen the clusters such that the corresponding CVM approximation can be applied to any pedigree, regardless of inbreeding and size, and the numerical minimization can be performed within reasonable time and a reasonable amount of memory usage for large problems. Computation time and memory usage of the CVM increase exponentially with cluster size, but approximately linearly with the number of clusters. The accuracy of the CVM approximation generally increases with cluster size, resulting in a trade-off between accuracy and efficiency.

For every non-founder individual i and each pair of adjacent markers l and $l + 1$, we define the cluster

$$B_i^{l,l+1} = \left\{ G_i^{l,p}, G_i^{l,m}, v_i^{l,p}, v_i^{l,m}, G_{\text{fa}(i)}^{l,p}, G_{\text{fa}(i)}^{l,m}, G_{\text{mo}(i)}^{l,p}, G_{\text{mo}(i)}^{l,m}, \right. \\ \left. G_i^{l+1,p}, G_i^{l+1,m}, v_i^{l+1,p}, v_i^{l+1,m}, G_{\text{fa}(i)}^{l+1,p}, G_{\text{fa}(i)}^{l+1,m}, G_{\text{mo}(i)}^{l+1,p}, G_{\text{mo}(i)}^{l+1,m} \right\}. \quad (2)$$

This basic cluster is illustrated in figure 1. Each cluster contains the genotype variables of the child and both its parents for two adjacent markers. It also contains the paternal and maternal segregation indicators of the child for these two adjacent markers. As a result, the CVM treats the inheritance of the child from its parents for two adjacent markers with exact probability calculus². With this choice the number of clusters scales linearly with both the number of individuals and the number of markers, irrespective of the pedigree structure. As we will show, computation time and memory usage for this choice of clusters are acceptable, while the accuracy of the approximation is high.

²The observed genotypes M_i^l are not explicitly included in the cluster. Because their value depends only on the unobserved genotype variables through the conditional probability tables $P(M_i^l | G_i^{l,p}, G_i^{l,m})$ (see appendix for details), as a pre-processing step we integrate over M_i^l before applying the CVM.

Illustration of CVMHAPLO: In this section we demonstrate the procedures with a simple example. We consider a family consisting of a father, a mother, a daughter, and a son. Genotype data is simulated for three markers at 5 cM intervals. The true ordered genotypes are

marker	father	mother	daughter	son
1	BA	BB	BB	BB
2	AA	BB	AB	AB
3	AB	BA	AB	AA

We now apply CVMHAPLO to this data set. With the choice of clusters given by (2), we have four clusters for this example:

$$\begin{aligned}
B_{\text{da}}^{1,2} &= \{G_{\text{da}}^{1,p}, G_{\text{da}}^{1,m}, G_{\text{fa}}^{1,p}, G_{\text{fa}}^{1,m}, G_{\text{mo}}^{1,p}, G_{\text{mo}}^{1,m}, v_{\text{da}}^{1,p}, v_{\text{da}}^{1,m}, \\
&\quad G_{\text{da}}^{2,p}, G_{\text{da}}^{2,m}, G_{\text{fa}}^{2,p}, G_{\text{fa}}^{2,m}, G_{\text{mo}}^{2,p}, G_{\text{mo}}^{2,m}, v_{\text{da}}^{2,p}, v_{\text{da}}^{2,m}\}, \\
B_{\text{so}}^{1,2} &= \{G_{\text{so}}^{1,p}, G_{\text{so}}^{1,m}, G_{\text{fa}}^{1,p}, G_{\text{fa}}^{1,m}, G_{\text{mo}}^{1,p}, G_{\text{mo}}^{1,m}, v_{\text{so}}^{1,p}, v_{\text{so}}^{1,m}, \\
&\quad G_{\text{so}}^{2,p}, G_{\text{so}}^{2,m}, G_{\text{fa}}^{2,p}, G_{\text{fa}}^{2,m}, G_{\text{mo}}^{2,p}, G_{\text{mo}}^{2,m}, v_{\text{so}}^{2,p}, v_{\text{so}}^{2,m}\}, \\
B_{\text{da}}^{2,3} &= \{G_{\text{da}}^{2,p}, G_{\text{da}}^{2,m}, G_{\text{fa}}^{2,p}, G_{\text{fa}}^{2,m}, G_{\text{mo}}^{2,p}, G_{\text{mo}}^{2,m}, v_{\text{da}}^{2,p}, v_{\text{da}}^{2,m}, \\
&\quad G_{\text{da}}^{3,p}, G_{\text{da}}^{3,m}, G_{\text{fa}}^{3,p}, G_{\text{fa}}^{3,m}, G_{\text{mo}}^{3,p}, G_{\text{mo}}^{3,m}, v_{\text{da}}^{3,p}, v_{\text{da}}^{3,m}\}, \\
B_{\text{so}}^{2,3} &= \{G_{\text{so}}^{2,p}, G_{\text{so}}^{2,m}, G_{\text{fa}}^{2,p}, G_{\text{fa}}^{2,m}, G_{\text{mo}}^{2,p}, G_{\text{mo}}^{2,m}, v_{\text{so}}^{2,p}, v_{\text{so}}^{2,m}, \\
&\quad G_{\text{so}}^{3,p}, G_{\text{so}}^{3,m}, G_{\text{fa}}^{3,p}, G_{\text{fa}}^{3,m}, G_{\text{mo}}^{3,p}, G_{\text{mo}}^{3,m}, v_{\text{so}}^{3,p}, v_{\text{so}}^{3,m}\}.
\end{aligned}$$

Here **da** denotes the daughter, **so** denotes the son, **fa** denotes the father, and **mo** denotes the mother. For every child there are two clusters, one for markers 1 and 2, and one for markers 2 and 3. A cluster contains the paternal and maternal genotype variables (e.g. $G_{\text{da}}^{1,p}$) and segregation indicators (e.g. $v_{\text{da}}^{1,p}$) of the child, and the genotype variables of both parents (e.g. $G_{\text{fa}}^{2,m}$) for the two markers in the cluster. Thus the genotype variables and segregation indicators of the children defined for marker 2 are contained in two clusters; the genotype variables of the parents defined for marker 2 are contained in all four clusters. With this set of clusters the CVM will yield approximate probabilities.

With $p = 0.5\%$, CVMHAPLO requires four iterations to reconstruct the haplotypes. In table 1 the marginal distributions of the ordered genotypes as computed with the CVM, and the ordered genotypes that are assigned from these marginals, are shown for all four iterations. In the first iteration all homozygous genotypes can be assigned, since the corresponding marginal distributions indicate that one configuration has probability one. Also the ordered genotypes of the daughter and son for marker 2 can be assigned as these are unambiguously determined by the homozygous genotypes of the parents. The heterozygous ordered genotype of the father for marker 1 is assigned in the first iteration since it has the highest q_{map} . Note that the symmetry due to the unknown phase of founder genotypes was removed beforehand (see appendix for details). In the second iteration the

marginal distributions are re-estimated conditioned on the marker data and the ordered genotypes assigned in the first iteration. The ordered genotype of the father for marker 3 now has the highest probability: $q_{\text{map}} = 0.9$ and is assigned: the MAP-configuration is $\{G_{\text{fa}}^{3,p}, G_{\text{fa}}^{3,m}\}_{\text{map}} = \{G_{\text{fa}}^{3,p} = A, G_{\text{fa}}^{3,m} = B\}$. In the third iteration the ordered genotype of the daughter for marker 3 is assigned and finally in the fourth iteration the ordered genotype of the mother for marker 3 is assigned.

The inferred haplotype configuration is identical to the true haplotype configuration and is an exact maximum likelihood solution. In this example, the absolute error of the CVM approximation of the conditional marginal probabilities of the ordered genotypes is in the order of 10^{-4} . Note that the true ordering of the genotypes was not available to the algorithm.

MATERIALS

Data sets

We evaluated CVMHAPLO on two pedigrees that were taken from experimental linkage studies. Pedigree I is an extended pedigree and concerns an affected/not-affected disease with a complex mode of inheritance. It consists of 53 individuals, of which 13 have been genotyped with an Affymetrix 10K SNP array. The pedigree is shown in figure 2. Pedigree II is a complex pedigree of 368 individuals, of which 262 were genotyped for 8 SNP markers spanning approximately 0.08 cM. It is taken from a QTL fine mapping study in a chicken population. The pedigree is shown in simplified form in figure 3. Pedigree IIsub contains a subset of the individuals in pedigree II and is shown in figure 4. Exact computations are feasible in this pedigree. In all analyses the Haldane mapping function was used. Details of the data sets analyzed are given in table 2.

CVMHAPLO

100 outer loop iterations and 2 inner loop iterations of the double loop algorithm were used for the first iteration of the haplotype inference algorithm; in the subsequent iterations 10 outer loop and 2 inner loop iterations were used, see appendix for details. For all simulations we used $p = 0.5\%$.

Implementation

The implementation of CVMHAPLO was done in C++ and compiled with gcc version 4.1.1.

Hardware

All simulations were performed on a cluster of 5 machines with two dual-core 2.4 GHz AMD64 processors each and 4 GB of physical memory available per processor, running the Linux operating system.

RESULTS

Accuracy of the marginal distributions of ordered genotypes

In this section we assess the accuracy of the CVM approximation of the marginal distributions of the ordered genotypes as computed with CVMHAPLO for problems for which exact likelihood computations are feasible. We compared the CVM approximation of the marginal distributions $Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M})$ with the exact marginal distributions $P(G_i^{l,p}, G_i^{l,m} | \mathbf{M})$ using the absolute difference as error measure. Unfortunately there are no linkage programs that provide exact marginal distributions; therefore we implemented the junction tree algorithm (Jensen, 1996; Jensen and Kong, 1999) to calculate these.

40 replicates of 5 markers were simulated for pedigree I (configuration A in table 2). Figure 5 shows a scatter plot of the exact marginal probabilities versus the approximate CVM marginal probabilities. The mean error was 0.0044 ± 0.012 . Eight of the CVM estimates had an error larger than 0.25; the figure shows that these correspond to exact marginal probabilities that were close to 0.5. The error of the CVM estimates was generally smaller for exact marginal probabilities close to one and zero. This is relevant because the ordered genotypes that correspond to these extreme marginal probabilities are the ordered genotypes with the least uncertainty that will be assigned by CVMHAPLO in every iteration, while the ordered genotypes with the most uncertainty will not be assigned.

We also determined the accuracy of the approximation in the same pedigree and configuration with real marker data. The mean error was 0.0036 ± 0.0073 , with a maximum error of 0.059. We conclude that the CVM estimates of the marginal probabilities of the ordered genotypes are accurate for the purpose of haplotyping.

Accuracy of the inferred haplotypes

In this section we evaluate the accuracy of the reconstructed haplotypes in simulated data sets where the true inheritance was known. We define accuracy as the percentage of assigned ordered genotypes equal to the true simulated ordered genotype.

Comparison with exact maximum likelihood methods: We assessed the performance of CVMHAPLO in two pedigrees for which exact calculation of the maximum likelihood haplotype configuration was feasible. We analyzed pedigree I with 5 markers using our own implementation of the junction tree algorithm and pedigree Isub with 8 markers

using SUPERLINK (configuration B and G in table 2). Replicates were simulated with 30, 40, . . . , 90 % of the individuals genotyped for all markers. The genotyped individuals were chosen at random. For each percentage of genotyped individuals 40 replicates were simulated, resulting in a total number of 280 replicates per pedigree.

Column two and three in table 3 show that the accuracy of respectively the exact maximum likelihood haplotype configuration and the haplotype configuration obtained with CVMHAPLO were similar for all percentages of genotyped individuals, for both pedigrees. The fourth and fifth column show the log-likelihoods $\log P(\mathbf{G}_{\text{assigned}}, \mathbf{v}_{\text{assigned}}, \mathbf{M})$ of the corresponding haplotype configurations. For both pedigrees the log-likelihoods of CVMHAPLO were very close to the exact maximum log-likelihoods when the percentage of genotyped individuals was high, and slightly lower when the percentage of genotyped individuals was low. Although in theory higher likelihoods should result in higher accuracies, we find that the differences in the likelihoods did not significantly affect the accuracy.

We also performed a partial haplotype reconstruction where we used the confidence measure (1) as a stopping criterion. The sixth column in the table shows the accuracy of the haplotype configuration obtained from the iteration n of CVMHAPLO where the confidence in the assignment was 99 %. Indeed, independently of the percentage of genotyped individuals the accuracy of this partial haplotype configuration was approximately 99 %. The last column shows that the percentage of assigned ordered genotypes in the partial haplotype configuration decreased when fewer individuals had genotype information. In pedigree I the percentage of assigned ordered genotypes was lower than the percentage of genotyped individuals, while it was higher in pedigree Isub, indicating a non-trivial dependence of the accuracy on the structure of the pedigree and distribution of the genotyped individuals over the pedigree. These results show that (1) provides a useful stopping criterion to obtain partial assignments of high accuracy.

To assess the performance of CVMHAPLO in the real data sets of pedigree I and Isub, we performed the simulations of table 3, however simulating genotype data for the same individuals as in the real data set (configuration A and H in table 2) rather than for individuals selected at random. For pedigree I we found that the log-likelihoods of CVMHAPLO were on average 2.03% lower than the exact maximum log-likelihood, but that the accuracy was higher (75.95 % and 73.02 %, respectively). For pedigree Isub we found that the log-likelihoods of CVMHAPLO were on average 2.02% lower than the exact maximum log-likelihood, but that the accuracies were comparable (91.56 % and 92.05 %, respectively). These results are compatible with the results of table 3: when the pedigree contains untyped individuals the accuracy of CVMHAPLO is comparable to the accuracy of the exact maximum likelihood approach, while the log-likelihoods are slightly lower.

When applied to the real data sets of pedigree I and pedigree Isub, CVMHAPLO yielded a haplotype configuration with a log-likelihood that was 2.3 % lower than the exact maximum log-likelihood for pedigree I, and a log-likelihood that was equal to the exact maximum log-likelihood for pedigree Isub. These results suggest that CVMHAPLO will also accurately infer haplotypes in the real data sets.

We conclude that the accuracy of the haplotype configurations inferred with CVMHAPLO was high and similar to the accuracy of exact maximum likelihood haplotype configurations.

Comparison with SIMWALK2: For pedigree I 40 replicate data sets with 20 markers were simulated and for pedigree II 8 replicate data sets with 8 markers were simulated (respectively configuration C and F in table 2). The number of replicates for pedigree II was relatively small due to the long computation times of SIMWALK2. Exact computation of the maximum likelihood haplotype configuration was not feasible in these data sets. Figure 6 shows the accuracy as a function of the percentage of assigned ordered genotypes of CVMHAPLO and SIMWALK2 for both pedigrees.

By default, SIMWALK2 assigns only a subset of the alleles from the unordered marker data, the size of which depends on the informativeness of the marker data (the two leftmost data points ‘SIMWALK2 (all)’ and ‘SIMWALK2 (genotyped)’ in figure 6). The subset consists of those alleles that are transmitted to an observed genotype given the inheritance vector of the (approximate) maximum likelihood configuration. SIMWALK2 can also be forced to assign all alleles in the haplotypes (the two rightmost data points ‘SIMWALK2 (all)’ and ‘SIMWALK2 (genotyped)’ in figure 6). Depending on the number of iterations, CVMHAPLO infers anywhere between zero and all of the ordered genotypes.

When all alleles in the haplotypes are assigned (100 % on the horizontal axis), the accuracy of CVMHAPLO was not significantly different from the accuracy of SIMWALK2, both on the subset of genotyped individuals and the full pedigree. As expected, the accuracy of SIMWALK2 and CVMHAPLO was higher in the subset of genotyped individuals. The likelihoods of the haplotype configurations inferred with CVMHAPLO were slightly lower than the likelihoods of the haplotype configurations inferred with SIMWALK2: in pedigree I the mean difference in the log-likelihood was $-1.6 \% \pm 1.2 \%$; in pedigree II the mean difference was $-3.3 \% \pm 2.3 \%$. Apparently the lower likelihoods did not significantly affect the overall accuracy, in agreement with our previous results.

In the case of partial assignments, we infer from figure 6 that the accuracy of SIMWALK2 and CVMHAPLO are similar for the genotyped individuals, and that the accuracy of CVMHAPLO is significantly higher for the individuals without genotype information. The accuracy of CVMHAPLO was very high when only the ordered genotypes with high confidence (eq. 1) were assigned, and decreased as more ordered genotypes were assigned. In contrast, the criterion used by SIMWALK2 to flag the alleles that could be assigned with certainty was more coarse. This difference was more pronounced in pedigree I than in pedigree II. We attribute this difference to a large extent to the larger percentage of missing data in pedigree I. We conclude that CVMHAPLO gives more accurate partial assignments than SIMWALK2 when the percentage of missing values is high.

Scaling with the number of markers: To assess the scaling of the accuracy of CVMHAPLO with the number of markers, we analyzed 10 replicates with 20 markers and 10 replicates with 200 markers for pedigree I (configuration C and D in table 2, respectively). In order to obtain replicates with comparable marker informativeness, the replicates with 200 markers consisted of 10 adjacent blocks of the markers in the replicates with 20 markers. Analysis of the replicates with 200 markers with CVMHAPLO was feasible, whereas SIMWALK2 did not converge in reasonable time (one week for a single replicate).

Figure 7 shows that the average accuracy for the replicates with 20 markers and 200 markers was similar. We conclude that the accuracy of CVMHAPLO does not degrade with the number of markers.

The effect of marker-marker linkage disequilibrium: We investigated the effect of marker-marker linkage disequilibrium (LD) on the accuracy of the haplotype reconstruction of CVMHAPLO and SIMWALK2. For pedigree I, LD was generated as follows. Five haplotype blocks each containing four markers were defined. Next, for each block a pool of four haplotypes with randomly chosen frequencies was created; the resulting mean pair-wise LD coefficient $|D'|$ was 0.85 ± 0.28 for markers within a haplotype block. For pedigree II LD was generated by assuming a single haplotype block with a pool of 25 haplotypes with randomly chosen frequencies. This resulted in a mean pair-wise LD coefficient $|D'|$ was 0.59 ± 0.36 . For each block the haplotypes of the founders were assigned by sampling a haplotype from the corresponding pool, and the haplotypes for the non-founders were obtained by gene-dropping, whereby recombination between markers within a block was allowed. Thus, the alleles of markers in different blocks were assumed to be in equilibrium. For pedigree I and II respectively 40 and 8 replicates were simulated. These were analyzed using the correct marginal allele frequencies (obtained by marginalizing the true haplotype frequencies), however under the assumption of linkage equilibrium between the markers. Thus, the replicates were analyzed using an incorrect model. The results were compared to results obtained with an equal number of replicates simulated under the assumption of linkage equilibrium and the same marginal allele frequencies. Both LD and non-LD replicates are shown as configuration E and F in table 2.

We did not find a significant effect of LD on the accuracy of the inferred haplotypes for pedigree I and II, neither for the genotyped individuals nor for the individuals without genotype information (results not shown). We found this to be the case for both CVMHAPLO and SIMWALK2. In the presence of LD we also observed that the log-likelihood of the fully reconstructed CVMHAPLO haplotype configurations were slightly lower than the log-likelihood of the haplotype configurations of SIMWALK2, similar to what we found in data sets simulated without LD. Thus we expect the effect of marker-marker LD, which may be assumed to be present in real data sets, on the accuracy of the inferred haplotypes to be small.

Evaluation of the confidence measure: Figure 6 demonstrates that it would be useful to have an indication of the reliability of the (partial) haplotype configuration, as the accuracy decreased significantly when a larger subset of ordered genotypes was assigned. For these replicates we therefore compared the confidence level from eq. 1 to the accuracy of the (partial) haplotype configuration inferred in iteration n of CVMHAPLO, for all n . Figure 8 shows for a given confidence level the mean accuracy of the corresponding haplotype configurations, where the leftmost circle and square data point correspond to the haplotype configurations where all the ordered genotypes were assigned (the haplotypes obtained in the final iteration). We see that for pedigree I the confidence was lower than the accuracy, but still highly correlated with it. For pedigree II the confidence of the haplotype configurations gave a very good indication of the accuracy. The difference is most likely due to the fact that the marker data in pedigree II were more informative than in pedigree I. We conclude that the confidence measure (1) gives a useful indication of the accuracy (assuming absence of genotype errors) and may be used to control the accuracy of the inferred haplotypes.

Comparison of computation time and memory usage

Finally we report the computation time and memory usage for all the experiments that were performed. For CVMHAPLO we report the computation time of the marginal posterior distributions (computed in the first iteration) and the computation time of the full reconstruction (computation time of all subsequent iterations) separately. When applicable we report the values for SIMWALK2 and for the exact computation with the junction tree algorithm.

For all analyses performed with CVMHAPLO we used a fixed value of $p = 0.5\%$ for the percentage of ordered genotypes assigned in every iteration, independent of the number of markers and individuals. Theoretically, for a fixed percentage p computation time of CVMHAPLO is expected to scale linearly with the number of markers and approximately linearly with the number of individuals depending on the pedigree structure, which is confirmed by the results shown in table 4. Although CVMHAPLO required more memory, it was significantly more efficient than SIMWALK2 for the complex pedigree II, and scaled more favorably with the number of markers.

DISCUSSION

In order to obtain useful results with maximum likelihood methods, it must be assumed that the distribution of the parameters that are being estimated (for the purpose of haplotype inference these are the ordered genotypes) is peaked around the maximum likelihood solution. If genotype information is available for all individuals and markers, this assumption is generally valid, however, if there are many missing genotypes this assumption may not be valid. Although a maximum likelihood estimate will yield the most likely value of the parameters given the observations, it is not guaranteed that all parameters can be inferred with certainty. Indeed, as we have shown in our experiments, the accuracy of a partial haplotype configuration can be significantly different from a full haplotype reconstruction. Therefore in the case of missing marker data there is a clear need to monitor the quality of the haplotype inference and we suggest to use equation 1 for this purpose. Our results indicate that it can be used to obtain partial haplotype configurations of high accuracy.

In this light it is interesting to note that although our approach does not explicitly maximize the likelihood, it inferred haplotype configurations with nearly optimal likelihood when full genotype information was available. In case of missing genotypes, the log-likelihood of the inferred haplotype configurations was ~ 2 percent lower than the exact maximum log-likelihood; however the accuracy was not significantly different. Our results suggest that the assignments that are suboptimal in the sense of the likelihood are limited to the ordered genotypes that cannot be inferred with high certainty from the marker data.

An important parameter of CVMHAPLO is p , the percentage of ordered genotypes assigned in every iteration. In general, for smaller values of p the accuracy will be higher and the computation times longer. In our experience values of p smaller than 0.5% did not yield significantly higher accuracies. With $p = 0.5\%$ only 4 replicates of the ~ 700 replicates analyzed the algorithm required a restart with $p = 0.25\%$. For higher values of p the number of replicates where CVMHAPLO made an inconsistent assignment increased somewhat. Therefore we recommend to use the default value of $p = 0.5\%$, but it can be adjusted by the user. When an inconsistent assignment has been detected we expect that it is not necessary to completely restart the algorithm as we do now, but that it may also suffice to reset the algorithm only a few iterations backwards. In principle genotype elimination methods (Lange and Goradia, 1987; O'Connell and Weeks, 1998) can be used to detect and prevent incompatible assignments. We plan to investigate whether more computationally inexpensive heuristics can be devised to prevent inconsistent assignments.

We compared our approach to the approximate maximum likelihood haplotyping algorithm of SIMWALK2, since like our approach, SIMWALK2 is a statistical approach that does not require absence of recombinations or tightly linked markers and does not assume the number of recombinations to be minimal. Furthermore, SIMWALK2 is commonly used by practitioners. In previous work on the estimation of parametric LOD scores in pedi-

rees without inbreeding (Albers et al., 2006), we showed that our approach based on the CVM was more efficient than the MCMC-sampler implemented in the computer program MORGAN (Thompson, 1994; Thompson and Heath, 1999; George and Thompson, 2003). Since MORGAN has no option for haplotyping and cannot be applied to general pedigrees, we believe a comparison here would not be of added value. We also considered the Integer Linear Programming algorithm implemented in PEDPHASE (Li and Jiang, 2004), since this algorithm does not require absence of recombinations. On the simulated data sets for pedigree I (configuration C in table 2), the accuracy was on average 10 % lower than that of CVMHAPLO, and the log-likelihoods were on average 50 % lower than those of SIMWALK2. It could not analyze the real and simulated data sets (configuration F) for pedigree II within one week. The Block-Extension algorithm in PEDPHASE produced inconsistent output for the data sets simulated for pedigree I and terminated with error status for the data sets simulated for pedigree II. Finally, on simulated data (400 SNPs covering 100 cM) in a pedigree of 400 outbred mice, where a small number of parents had many offspring, we found that our approach was more accurate than the approach described by Windig and Meuwissen (2004), although it was less efficient (results not shown). This approach requires that sufficient genotyped offspring are available for each parent and therefore we expect that it will be less accurate than SIMWALK2 and CVMHAPLO on the data sets considered in this article, especially for pedigree I. For these reasons we have not included this approach in our comparison.

Like SIMWALK2, our haplotype inference algorithm currently does not explicitly account for linkage disequilibrium between the markers. In dense SNP panels, such as the Affymetrix 10K and Illumina 6K panels, significant marker-marker LD has been shown to be present (Peralta et al., 2005). LD can lead to a bias in LOD scores when the marker alleles are assumed to be in equilibrium, especially when parental genotypes are missing (Huang et al., 2004; Abecasis and Wigginton, 2005). Our finding that there was no significant effect of LD on the accuracy appears to be in contradiction, but we believe the difference can be explained by the fact that we evaluated the accuracy of a single haplotype configuration, while Huang et. al investigated LOD scores, which may be more sensitive to violations of the assumption of linkage equilibrium. The issue of LD is a modeling issue and therefore in principle unrelated to the issue of the quality of the CVM approximation, although, of course, the quality of the CVM approximation may depend on the model. The CVM approximation can be applied to any probabilistic model and in particular to a pedigree likelihood model that includes LD. Pair-wise modeling of LD between markers would not require a different choice of the clusters in the CVM approximation. This is a direction for further research.

We applied our algorithm to data sets consisting of SNP markers only. Currently our software does accept multi-allelic markers. Due to the increased state space in case of multi-allelic markers, the efficiency of our implementation is not as high as in the case of SNPs. Work is in progress to improve the efficiency for multi-allelic markers by applying additional preprocessing techniques and using clusters with fewer variables in the CVM

approximation.

In this paper we assumed absence of genotyping errors. In practice this will rarely be the case. A simple heuristic for error detection is to locate unlikely double recombinants. These can be inferred from the marginal distributions over segregation indicators of adjacent loci, which can be trivially obtained from the cluster marginal distributions. However, it is preferable to use an error model as proposed by Sobel et al. (2002). Such an error model can be relatively easily incorporated into our approach, at the expense of a larger state space. Although the efficiency of CVMHAPLO will be reduced, we believe that the additional computational expense may be well justified. In a preliminary analysis of a real data set of 1600 animals genotyped for 14 closely linked markers, we found that inclusion of an error model significantly reduced the number of unlikely recombinant haplotypes, which suggests that haplotype effects can be estimated with better power. These results will be presented in a separate publication. We plan to incorporate the modelling of genotyping errors in future versions of CVMHAPLO.

Software

CVMHAPLO can be obtained by contacting the first author.

Acknowledgments

We are grateful to Addie Vereijken at Hendrix Genetics, Boxmeer, the Netherlands and Jim Stankovich at the Walter and Eliza Hall Institute of Medical Research, Melbourne, Australia for providing the data sets. We would like to thank the anonymous reviewers for their suggestions. This work was sponsored by the Dutch Technology Foundation (STW). We acknowledge the support of the Pascal Network of Excellence.

References

- Abecasis, G. R., Cherny, S. S., Cookson, W. O., and Cardon, L. R. 2002. Merlin-rapid analysis of dense genetic maps using sparse gene flow trees. *Nat Genet*, 30:97–101.
- Abecasis, G. R. and Wigginton, J. E. 2005. Handling marker-marker linkage disequilibrium pedigree analysis with clustered markers. *Am J Hum Genet*, 77:754–767.
- Albers, C. A., Leisink, M. A. R., and Kappin, H. J. 2006. The Cluster Variation Method for efficient linkage analysis on extended pedigrees. *BMC Bioinformatics*, 7:S1.

- Baruch, E., Weller, J. I., Cohen, M., Ron, M., and Seroussi, E. 2005. Efficient inference of haplotypes from genotypes on a large animal pedigree. *Genetics*. doi:10.1534/genetics.105.047134.
- Fishelson, J., Dovgolevsky, N., and Geiger, D. 2005. Maximum likelihood haplotyping for general pedigrees. *Human Heredity*, 59:41–60.
- Fishelson, J. and Geiger, D. 2002. Exact genetic linkage computations for general pedigrees. *Bioinformatics*, 18:S189–S198.
- Gao, G., Hoeschele, I., Sorensen, P., and Du, F. 2004. Conditional probability methods for haplotyping in pedigrees. *Genetics*, 167:2055–2065.
- George, A. W. and Thompson, E. A. 2003. Discovering disease genes multipoint linkage analyses via a new Markov Chain Monte Carlo approach. *Statistical Science*, 18:515–535.
- Heskes, T., Albers, C. A., and Kappen, H. J. 2003. Approximate inference and constrained optimization. In *Proceedings of Uncertainty in AI*, pages 313–320.
- Huang, Q., Shete, S., and Amos, C. I. 2004. Ignoring linkage disequilibrium among tightly linked markers induces false-positive evidence of linkage for affected sib pair analysis. *Am J Hum Genet*, 75:1106–1112.
- Jensen, C. S. and Kong, A. 1999. Blocking-Gibbs sampling for linkage analysis in large pedigrees with many loops. *Am J Hum Genet*, 65:885–902.
- Jensen, F. V. 1996. *An introduction to Bayesian networks*. UCL Press.
- Kikuchi, R. 1951. A theory of cooperative phenomena. *Physical Review*, 81:988.
- Kruglyak, L., Daly, M. J., Reeve-Daly, M. P., and Lander, E. S. 1996. Parametric and non-parametric linkage analysis: A unified multipoints approach. *Am J Hum Genet*, 58:1347–1363.
- Lange, K. and Goradia, T. M. 1987. An algorithm for automatic genotype elimination. *Am J Hum Genet*, 40:250–256.
- Lange, K. and Sobel, E. 1996. Descent graphs in pedigree analysis application to haplotyping, location scores, and marker-sharing statistics. *Am J Hum Genet*, 58:1323–1337.
- Lauritzen, S. L. 1996. *Graphical models*. Oxford University Press.
- Lauritzen, S. L. and Sheehan, N. A. 2003. Graphical models for genetic analyses. *Statistical Science*, 4:489–514.

- Li, J. and Jiang, T. 2004. An exact solution for finding minimum recombinant haplotype configurations on pedigrees with missing data by integer linear programming. In *Proc. RECOMB04*, pages 101–110.
- McEliece, R. J., MacKay, D. J. C., and Cheng, J. F. 1998. Turbo decoding as an instance of Pearl’s ‘Belief Propagation’ algorithm. *IEEE Journal on Selected Areas in Communication*, 16:140–152.
- Morita, T. 1990. Cluster Variation Method and Moebius inversion formula. *J Stat Phys*, 59:819–825.
- Mukhopadhyay, N., Almasy, L., Schroeder, M., Mulvihill, W. P., and Weeks, D. E. 2005. Mega2: Data-handling for facilitating genetic linkage and association analyses. *Bioinformatics*, 21:2556–7.
- Murphy, K. P., Weiss, Y., and Jordan, M. I. 1999. Loopy Belief Propagation for approximate inference an empirical study. In *Proceedings of Uncertainty in AI*, pages 467–475.
- O’Connell, J. R. and Weeks, D. E. 1998. PedCheck: A program for identification of genotype incompatibilities in linkage analysis. *Am J Hum Genet*, 63:259–266.
- Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc.
- Peralta, J. M., Dyer, T. D., Warren, D. M., Blangero, J., and Almasy, L. 2005. Linkage disequilibrium across two different single-nucleotide polymorphism genome scans. In *Genetic Analysis Workshop 14: Microsatellite and single-nucleotide polymorphism*, volume 6(Suppl 1), page S86.
- Qian, D. and Beckmann, L. 2002. Minimum-recombinant haplotyping in pedigrees. *Am J Hum Genet*, 70:1434–1445.
- Sobel, E., Papp, J. C., and Lange, K. 2002. Detection and integration of genotyping errors in statistical genetics. *Am J Hum Genet*, 70:496–508.
- Thomas, A., Abkevich, V., and Bansai, A. 2000. Multilocus linkage analysis by Blocked Gibbs Sampling. *Statistics and Computing*, 10:259–269.
- Thompson, E. A. 1994. Monte Carlo likelihood in genetic mapping. *Stat Sci*, 9:355–366.
- Thompson, E. A. and Heath, S. C. 1999. Estimation of conditional multilocus gene identity among relatives. 33:95–113.

- Weiss, Y. 1997. Interpreting images by propagating bayesian beliefs. In *Advances in Neural Information Processing Systems 9*, pages 908–915.
- Wijsman, E. 1987. A deductive method of haplotype analysis in pedigrees. *Am J Hum Genet*, 41:356–373.
- Windig, J. J. and Meuwissen, T. H. E. 2004. Rapid haplotype reconstruction in pedigrees with dense marker maps. *J. Anim. Breed. Genet.*, 121:26–39.
- Yedidia, J. S., Freeman, W. T., and Weiss, Y. 2005. Constructing free-energy approximations and generalized Belief Propagation algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312.
- Zhang, K., Sun, F., and Zhao, H. 2005. HAPLORE: a program for haplotype reconstruction in general pedigrees without recombination. *Bioinformatics*, 21:90–103.

Appendix

We use the formalism of Bayesian networks (Pearl, 1988; Lauritzen, 1996; Fishelson and Geiger, 2002; Lauritzen and Sheehan, 2003) to represent the probability distribution that describes the problem of linkage analysis. The probability distribution is given by the product of conditional probability tables defined on subsets of a low, tractable number of variables. This facilitates the application of the Cluster Variation Method, which requires a set of tractable potential functions as input for the approximation.

A Bayesian network representation: The full probability distribution is given by

$$\begin{aligned}
P(\mathbf{M}, \mathbf{v}, \mathbf{G} | \mathbf{m}, \theta) = & \\
& \prod_{i \in \mathbf{F}, \mathbf{NF}} \prod_l P\left(M_i^l | G_i^{l,p}, G_i^{l,m}\right) \times \\
& \prod_{i \in \mathbf{NF}} \prod_l P\left(v_i^{l,p} | v_i^{l-1,p}, \theta_{l,l-1}\right) P\left(v_i^{l,m} | v_i^{l-1,m}, \theta_{l,l-1}\right) \times \\
& \prod_{i \in \mathbf{NF}} \prod_l P\left(G_i^{l,p} | v_i^{l,p}, G_{\text{fa}(i)}^{l,p}, G_{\text{fa}(i)}^{l,m}\right) P\left(G_i^{l,m} | v_i^{l,m}, G_{\text{mo}(i)}^{l,p}, G_{\text{mo}(i)}^{l,m}\right) \times \\
& \prod_{i \in \mathbf{F}} \prod_l P\left(G_i^{l,p} | \mathbf{m}^l\right) P\left(G_i^{l,m} | \mathbf{m}^l\right). \tag{3}
\end{aligned}$$

Here $\text{fa}(i)$ and $\text{mo}(i)$ represent the father and mother of individual i , respectively. The second line in this equation represents the observation model, i.e. the probability of the observed genotype conditional on the true ordered genotype, and incorporates the

unknown phase of the observed genotype. The third line represents the recombination model parameterized by the recombination frequencies between the adjacent markers $l - 1$ and l , for $l > 1$. The fourth line represents the paternal and maternal allele transmission from parents to children. The last line represents the prior allele distributions. Here, Hardy-Weinberg equilibrium of the alleles within a marker and linkage equilibrium between alleles of different markers is assumed. We assume that \mathbf{m} , θ are given as well as the M_i^l of the person-markers with genotype information.

The Cluster Variation Method: In this section we describe the CVM for the case that no ordered genotypes have been assigned and only unordered genotype observations are available. The case in which ordered genotype assignments are available does not require a different treatment, as assignments of ordered genotypes can be modelled as observed genotypes for which both the value of the two alleles and the ordering of the alleles is known.

The idea of the Cluster Variation Method (Kikuchi, 1951; Morita, 1990; Yedidia et al., 2005) is to approximate the intractable probability distribution

$$P(\mathbf{v}, \mathbf{G} | \mathbf{M}, \mathbf{m}, \theta) = \frac{P(\mathbf{M}, \mathbf{v}, \mathbf{G} | \mathbf{m}, \theta)}{P(\mathbf{M} | \mathbf{m}, \theta)}$$

in terms of marginal distributions on overlapping subsets of variables, i.e. the clusters. It requires specification of the set of clusters $B = \{\alpha_1, \alpha_2, \dots\}$: a collection of overlapping subsets of variables, chosen such that the corresponding approximate marginal distributions $Q_\alpha(\mathbf{x}_\alpha)$ are feasible for exact probability calculus³. We define I as the set of clusters that consists of *all* clusters that can be formed by the intersection of clusters in B and in I . Thus any intersection of clusters is contained in I . The choice of B determines the approximation and fully determines I . The following restrictions on the choice of the set of clusters B hold:

1. For every conditional probability table in the definition of equation 3, there must exist at least one cluster $\alpha \in B$ that contains all variables of the conditional probability table.
2. No cluster $\alpha_1 \in B$ is a subset of another cluster $\alpha_2 \in B$.

To motivate the formalism of the CVM, we first observe that the exact posterior distribution $P(\mathbf{v}, \mathbf{G} | \mathbf{M}, \mathbf{m}, \theta)$ can be obtained by minimizing the exact free energy defined as

$$F_{\text{exact}}(Q) \equiv \sum_{\mathbf{x}} Q(\mathbf{x}) \log \frac{Q(\mathbf{x})}{\Psi(\mathbf{x})}, \quad \text{subj. to } \sum_{\mathbf{x}} Q(\mathbf{x}) = 1,$$

³For ease of notation we do not explicitly state that $Q_\alpha(\mathbf{x}_\alpha)$ is conditioned on the marker data \mathbf{M} and assigned ordered genotypes $\mathbf{G}_{\text{assigned}}$, and that $\mathbf{x} = (\mathbf{v}, \mathbf{G})$.

with respect to $Q(\mathbf{x})$, where $\Psi(\mathbf{x})$ is the right hand side of eq. 3. This can be verified by simple differentiation with respect to Q . Since the functional $F_{\text{exact}}(Q)$ itself is generally intractable to evaluate, the Cluster Variation Method proposes to minimize the approximate free energy

$$F_{\text{CVM}}(\mathcal{Q}) = \sum_{\gamma \in B \cup I} a_{\gamma} \sum_{\mathbf{x}_{\gamma}} Q_{\gamma}(\mathbf{x}_{\gamma}) \log \frac{Q_{\gamma}(\mathbf{x}_{\gamma})}{\Psi_{\gamma}(\mathbf{x}_{\gamma})} \quad \text{subj. to consistency constraints} \quad (4)$$

with respect to the approximate marginal distributions $Q_{\gamma}(\mathbf{x}_{\gamma})$. Here \mathcal{Q} defines the collection of all approximate cluster marginals $\{Q_{\gamma}(\mathbf{x}_{\gamma}) : \gamma \in B \cup I\}$.

The *cluster potential* functions Ψ_{γ} are defined by the conditional probability tables of the Bayesian network in equation 3:

$$\Psi_{\gamma}(\mathbf{x}_{\gamma}) = \prod_{n: \mathbf{x}_{\{n, \pi(n)\}} \subseteq \mathbf{x}_{\gamma}} P(x_n | \mathbf{x}_{\pi(n)}). \quad (5)$$

In this equation x_n refers to a variable in the Bayesian network and $\mathbf{x}_{\pi(n)}$ denotes the set of variables on which variable x_n is conditioned. Thus, the product of conditional probability tables that defines a potential function Ψ_{γ} may contain the tables associated with the allele transmissions, $P(G_i^{l,m} | v_i^{l,m}, G_{\text{mo}(i)}^{l,p}, G_{\text{mo}(i)}^{l,m})$ as well as unordered genotype observations $P(M_i^l | G_i^{l,p}, G_i^{l,m})$ ⁴.

The Moebius coefficients a_{γ} satisfy

$$\sum_{\delta \in B \cup I \supseteq \gamma} a_{\delta} = 1, \quad \forall \gamma \in B \cup I,$$

and have the effect that for instance the evidence in the form of observed genotypes is not over counted. For details we refer to (Heskes et al., 2003).

The constraints in equation 4 are consistency and normalization constraints. The consistency constraints are

$$\sum_{\mathbf{x}_{\alpha \setminus \mathbf{x}_{\beta}}} Q_{\alpha}(\mathbf{x}_{\alpha}) = Q_{\beta}(\mathbf{x}_{\beta}), \quad \forall \alpha \in B, \beta \in I \subset \alpha,$$

which require any pair of clusters to have identical marginal distributions over the subset of variables contained in both clusters. The normalization constraints are

$$\sum_{\mathbf{x}_{\gamma}} Q_{\gamma}(\mathbf{x}_{\gamma}) = 1, \quad \forall \gamma \in B \cup I,$$

⁴Note again that the variables M_i^l are not explicitly included in the clusters included since they are observed.

which require the cluster marginal distributions to sum to one.

Thus, the approximate free energy $F_{\text{CVM}}(\mathcal{Q})$ is a sum of the free energies associated with each cluster $\gamma \in B \cup I$ multiplied by the Moebius coefficient a_γ . The Generalized Belief Propagation (GBP) algorithm (Yedidia et al., 2005; Pearl, 1988; Murphy et al., 1999) is a fixed point iteration scheme that finds extrema of $F_{\text{CVM}}(\mathcal{Q})$. However, GBP does not always converge. Therefore we use the convergent double loop algorithm described by Heskes et al. (2003) to minimize $F_{\text{CVM}}(\mathcal{Q})$. The idea of the double loop algorithm is to iteratively minimize convex upper bounds on the free energy. At each iteration of the algorithm a convex upper bound is calculated (the outer loop) which is minimized in the inner loop. The algorithm always converges to a (local) minimum of $F_{\text{CVM}}(\mathcal{Q})$, provided the inner loop has converged.

Consistency of the assignment: When p of the ordered genotypes are assigned simultaneously in one iteration, it is possible that the resulting $\mathbf{G}_{\text{assigned}}^{(n)}$ is inconsistent in the sense that this configuration has zero probability under the probability model of equation 3. This is not likely to happen if p is chosen sufficiently small. Our implementation of CVMHAPLO automatically detects inconsistent assignments as follows. When assignments are inconsistent, for one or more of the CVM cluster marginal distributions as a result $Q_\gamma(\mathbf{x}_\gamma) = 0, \forall \mathbf{x}_\gamma$, i.e. all states have zero probability. When all alleles in the pedigree are assigned and no inconsistency is detected like this, the inferred haplotype configuration is consistent.

Removal of symmetries: The probabilistic model defined by eq. 3 contains a number of symmetries. The first symmetry concerns the haplotypes of the founders: since by definition the founders do not have parents that are included in the pedigree, it is impossible to determine which haplotype is paternal and which is maternal. The second symmetry occurs when a father and mother are founders and also do not have genotype information. In this case a haplotype that is found in one of the children could be inherited from either parent with equal probability (assuming no recombination). More symmetries may be present, but in general it is hard to enumerate them all.

We find experimentally that CVMHAPLO yields better results when these symmetries are removed before application of the CVM. We remove the first symmetry by fixing for one child of every founder the parental source of the allele inherited from the founder for one marker. The second symmetry is removed by fixing in one grandchild of every untyped pair of founder parents the parental source of the inherited allele for one marker.

Algorithm 1 CVMHAPLO

- 1: $\mathbf{G}_{\text{assigned}}^{(0)} \Leftarrow \emptyset$
- 2: $n \Leftarrow 1$
- 3: Choose p
- 4: **repeat**
- 5: Run the double loop algorithm to compute the CVM approximate marginal distributions $Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)})$ for all individuals i and loci l .
- 6: **if** $\mathbf{G}_{\text{assigned}}^{(n-1)}$ is consistent **then**
- 7: For all individuals i and loci l , compute

$$q_{\text{map}}^{i,l} = \max Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}), \quad (6)$$

$$\{G_i^{l,p}, G_i^{l,m}\}_{\text{map}} = \arg \max Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}}^{(n-1)}). \quad (7)$$

- 8: Order the genotypes $\{q_{\text{map}}^{i_1, l_1}, q_{\text{map}}^{i_2, l_2}, \dots\}$ such that $q_{\text{map}}^{i_1, l_1} \geq q_{\text{map}}^{i_2, l_2} \geq q_{\text{map}}^{i_3, l_3}, \dots$
 - 9: Select the ordered genotypes with $q_{\text{map}} = 1$ and at most pNL ordered genotypes with $q_{\text{map}} < 1$, and assign the value of the corresponding genotype variables $\{G_i^{l,p}, G_i^{l,m}\}_{\text{map}}$ to $\{G_i^{l,p}, G_i^{l,m}\}$
 - 10: Update $\mathbf{G}_{\text{assigned}}^{(n)}$
 - 11: **else**
 - 12: $\mathbf{G}_{\text{assigned}}^{(n)} \Leftarrow \emptyset$
 - 13: $n \Leftarrow 0$
 - 14: $p \Leftarrow \frac{1}{2}p$
 - 15: **end if**
 - 16: $n \Leftarrow n + 1$
 - 17: **until** all ordered genotypes have been assigned
-

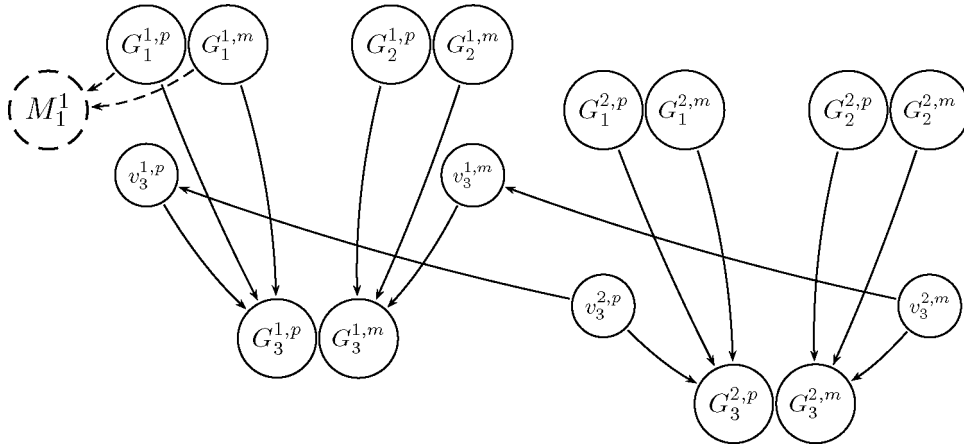


Figure 1: Illustration of the basic variables in the probabilistic model and the cluster choice. $G_i^{l,p}$ is the paternal allele of individual i and marker l , $G_i^{l,m}$ is the maternal allele. Paternal and maternal segregation indicators are respectively denoted by $v_i^{l,p}$ and $v_i^{l,m}$. The variables for the observed genotype variables M_i^l (dashed lines) are shown only for individual 1, but apply for every individual/marker for which a genotype was observed. A basic cluster consists of the genotype variables of the parents ($i = 1, i = 2$ in the figure) of one child ($i = 3$), as well as the genotype variables and the paternal and maternal segregation indicators of the child, for a pair of adjacent markers ($l = 1, 2$). These variables are shown as solid circles in the figure. For every individual that is not a founder and every pair of adjacent markers such a cluster is defined. The observed genotype variables M_i^l (dashed lines) are not explicitly included in the clusters since their value is observed (see running text).

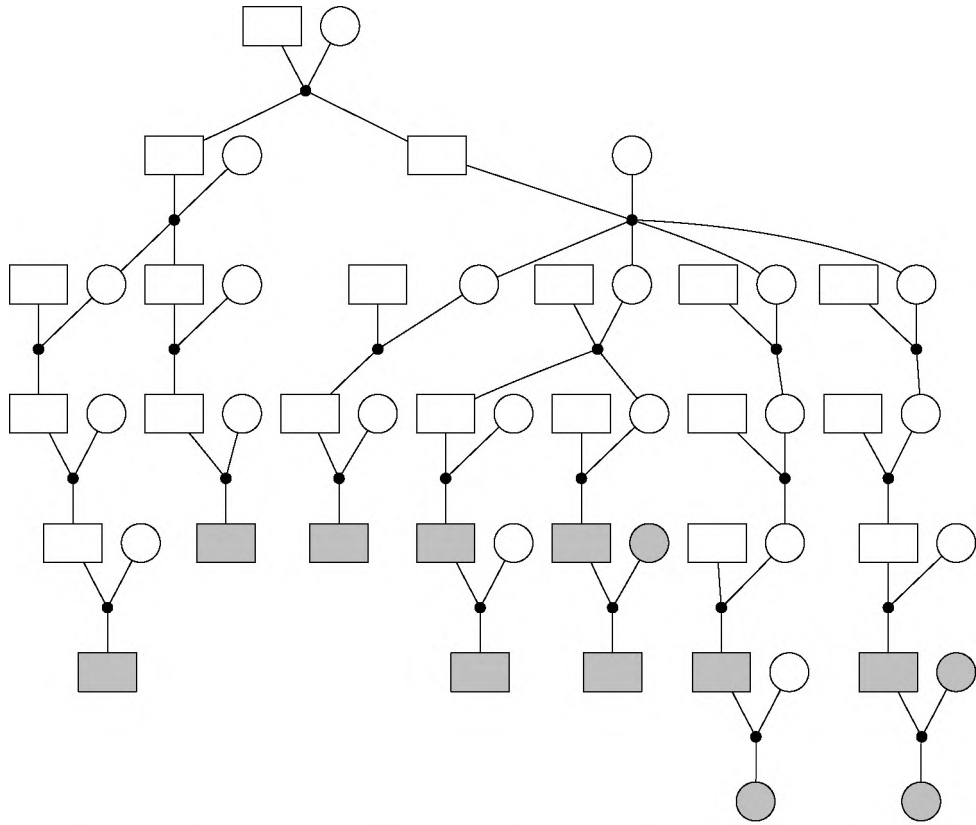


Figure 2: Pedigree I. Individuals marked gray have genotype information.

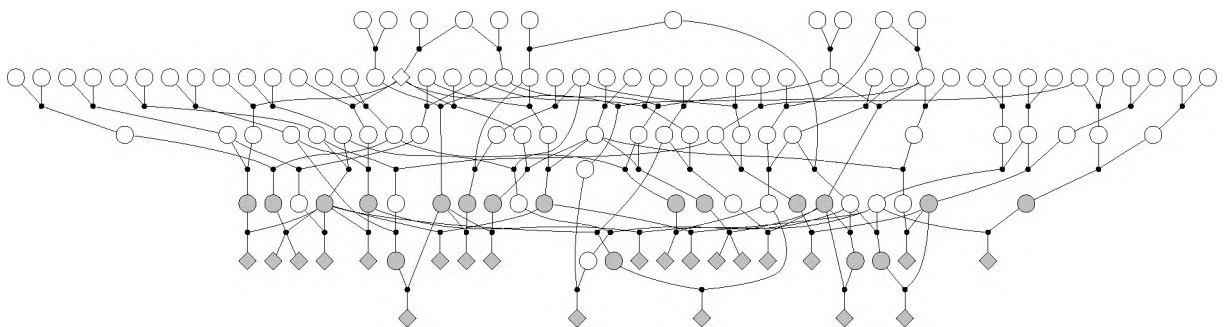


Figure 3: Pedigree II. Schematic representation. Diamonds represent groups of 5–15 individuals. Gray nodes represent groups with genotyped individuals.

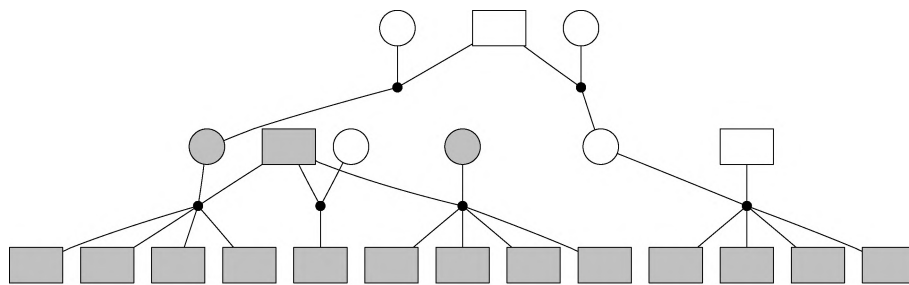


Figure 4: Pedigree Isub, a subpedigree of pedigree II.

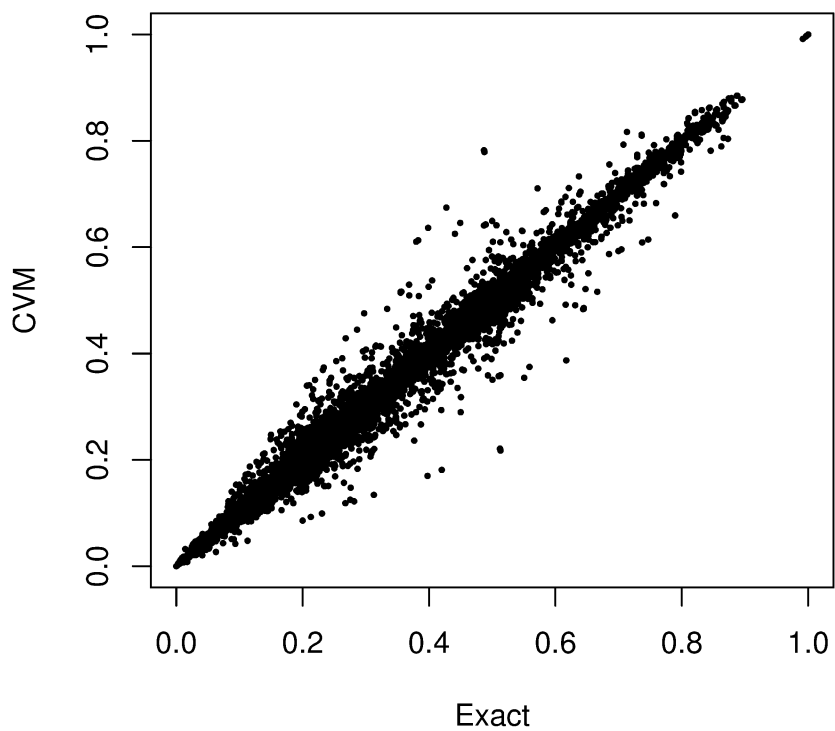


Figure 5: Scatter plot of exact marginal probabilities of the ordered genotypes versus the CVM approximation of the marginal probabilities, computed for pedigree I and 5 markers (configuration A in table 2).

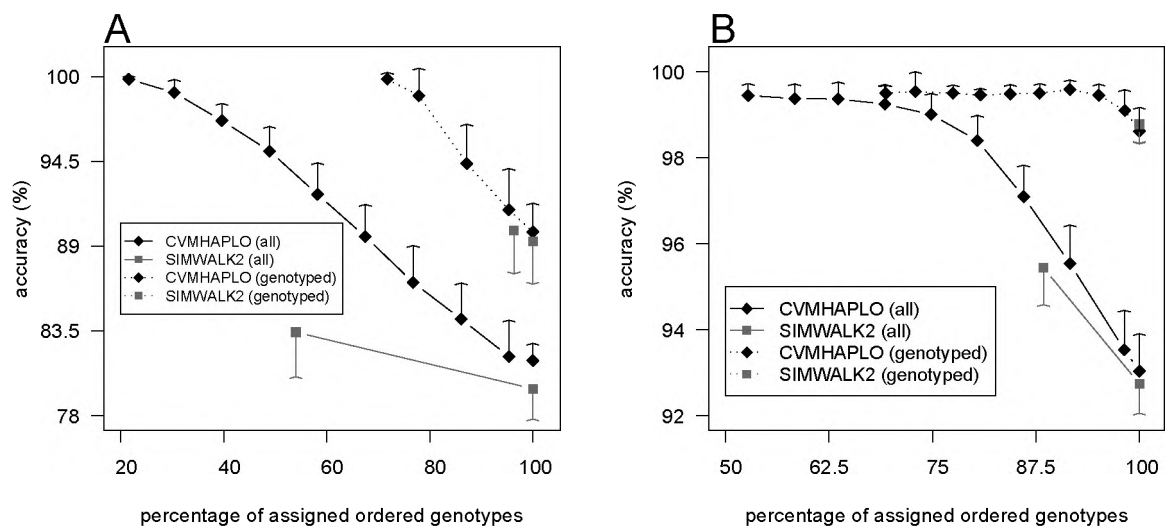


Figure 6: Comparison of the haplotype reconstruction accuracy of CVMHAPLO and SIMWALK2 for pedigree I (panel A, 20 markers, configuration C in table 2) and pedigree II (panel B, 8 markers, configuration F in table 2). Accuracy is defined as the percentage of assigned ordered genotypes identical to the true simulated ordered genotype. Accuracy is shown for all individuals ('all') and for genotyped individuals only ('genotyped'). Standard deviations are over 40 replicates for pedigree I and 8 replicates for pedigree II. For clarity, standard deviations are shown only on one side of the curve. Note the different scales on the horizontal and vertical axis.

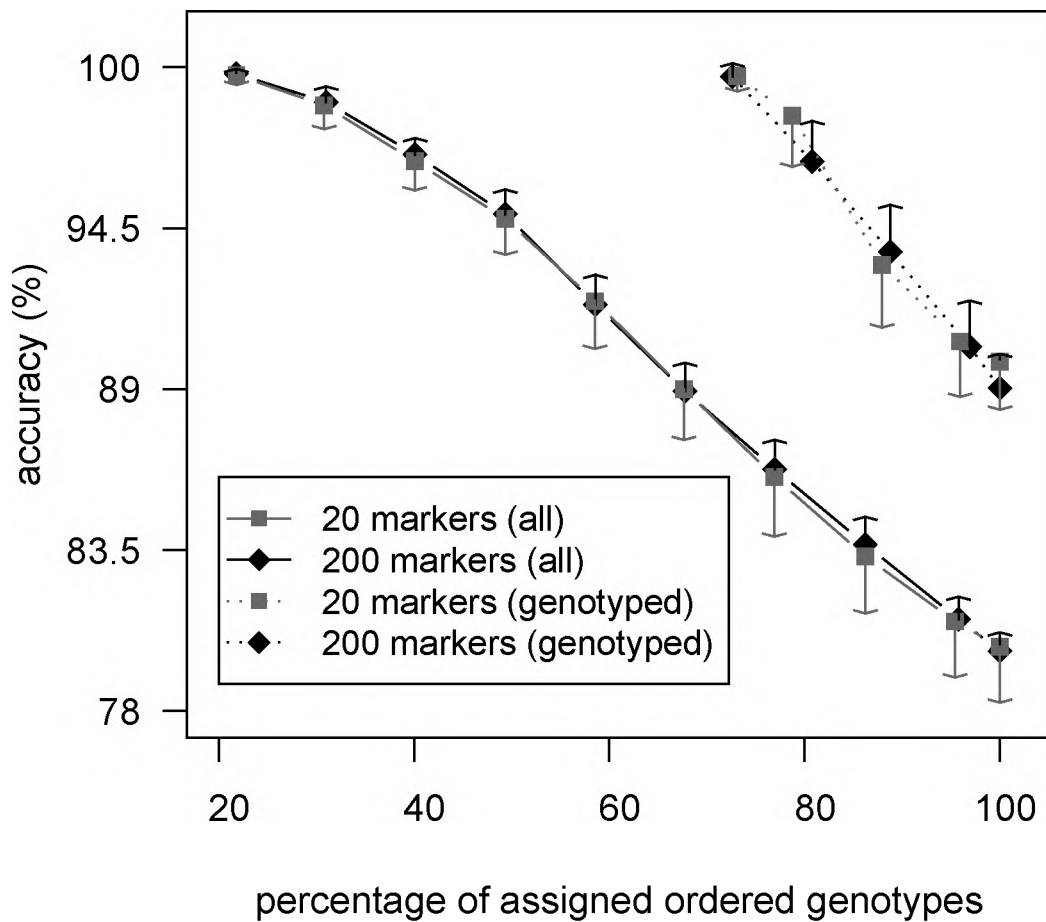


Figure 7: Comparison of the haplotype reconstruction accuracy of CVMHAPLO in pedigree I with 10 replicates of 20 markers (gray, configuration C in table 2) and 10 replicates of 200 markers (black, configuration D in table 2). Accuracy is shown for all individuals ('all') and for genotyped individuals only ('genotyped').

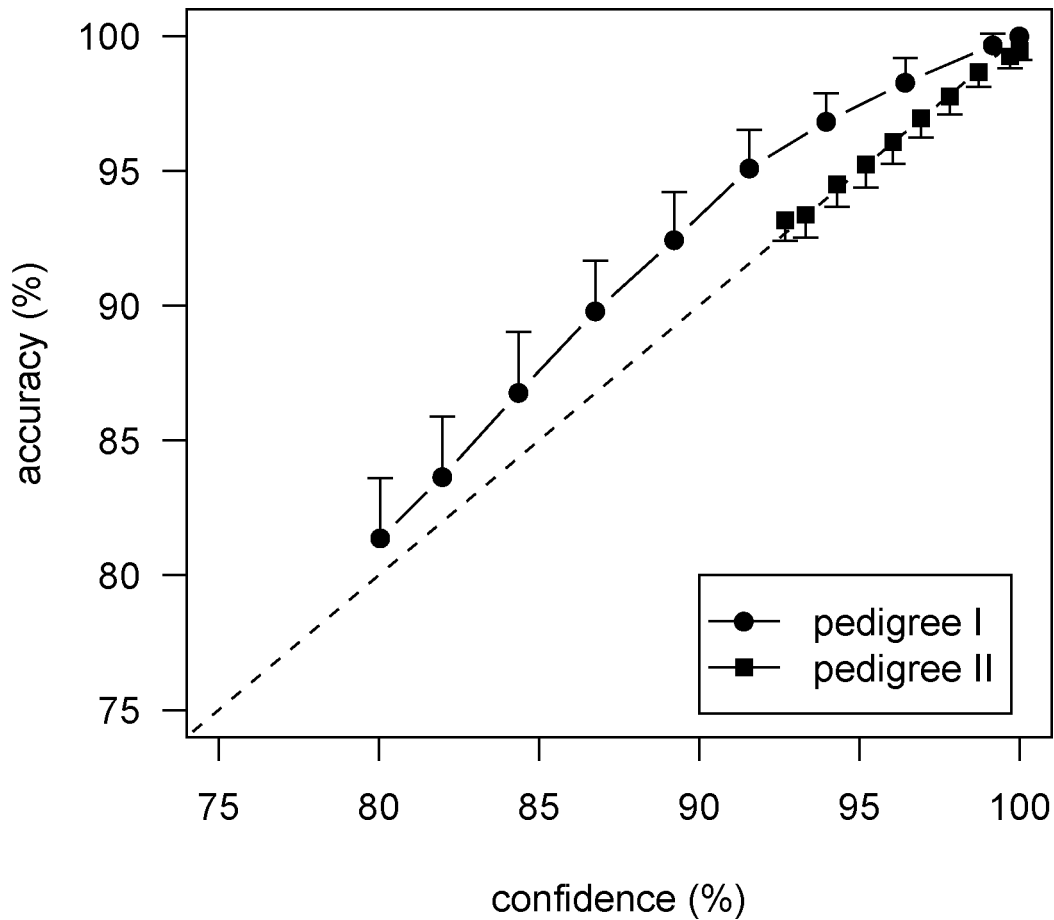


Figure 8: Accuracy vs. confidence of the haplotype configurations inferred with CVMHAPLO. For every iteration of CVMHAPLO the accuracy and the confidence level from (1) of the (partial) haplotype configuration was computed for the replicates analyzed in figure 6. The figure shows for a given confidence level the mean accuracy of the corresponding haplotype configurations. The leftmost circle and square correspond to the haplotype configurations where all ordered genotypes were assigned.

Table 1: Illustration of CVMHAPLO

iteration	marker	father	mother	daughter	son
1	1	(0.00,0.05,0.95,0.00)→ BA	(0.00,0.00,0.00,1.00)→ BB	(0.00,0.00,0.00,1.00)→ BB	(0.00,0.00,0.00,1.00)→ BB
	2	(1.00,0.00,0.00,0.00)→ AA	(0.00,0.00,0.00,1.00)→ BB	(0.00,1.00,0.00,0.00)→ AB	(0.00,1.00,0.00,0.00)→ AB
	3	(0.00,0.86,0.14,0.00)→ -	(0.00,0.20,0.80,0.00)→ -	(0.00,0.83,0.17,0.00)→ -	(1.00,0.00,0.00,0.00)→ AA
2	1	(0.00,0.00,1.00,0.00)→ BA	(0.00,0.00,0.00,1.00)→ BB	(0.00,0.00,0.00,1.00)→ BB	(0.00,0.00,0.00,1.00)→ BB
	2	(1.00,0.00,0.00,0.00)→ AA	(0.00,0.00,0.00,1.00)→ BB	(0.00,1.00,0.00,0.00)→ AB	(0.00,1.00,0.00,0.00)→ AB
	3	(0.00,0.90,0.10,0.00)→ AB	(0.00,0.18,0.82,0.00)→ -	(0.00,0.86,0.14,0.00)→ -	(1.00,0.00,0.00,0.00)→ AA
3	1	(0.00,0.00,1.00,0.00)→ BA	(0.00,0.00,0.00,1.00)→ BB	(0.00,0.00,0.00,1.00)→ BB	(0.00,0.00,0.00,1.00)→ BB
	2	(1.00,0.00,0.00,0.00)→ AA	(0.00,0.00,0.00,1.00)→ BB	(0.00,1.00,0.00,0.00)→ AB	(0.00,1.00,0.00,0.00)→ AB
	3	(0.00,1.00,0.00,0.00)→ AB	(0.00,0.10,0.90,0.00)→ -	(0.00,0.95,0.05,0.00)→ AB	(1.00,0.00,0.00,0.00)→ AA
4	1	(0.00,0.00,1.00,0.00)→ BA	(0.00,0.00,0.00,1.00)→ BB	(0.00,0.00,0.00,1.00)→ BB	(0.00,0.00,0.00,1.00)→ BB
	2	(1.00,0.00,0.00,0.00)→ AA	(0.00,0.00,0.00,1.00)→ BB	(0.00,1.00,0.00,0.00)→ AB	(0.00,1.00,0.00,0.00)→ AB
	3	(0.00,1.00,0.00,0.00)→ AB	(0.00,0.05,0.95,0.00)→ BA	(0.00,1.00,0.00,0.00)→ AB	(1.00,0.00,0.00,0.00)→ AA

For each individual, marker and iteration of CVMHAPLO the CVM approximation of the marginal distribution of ordered genotype $Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}})$ is shown between parentheses, where the probabilities are ordered as $(Q(AA), Q(AB), Q(BA), Q(BB))$. The value assigned to the ordered genotypes are shown next to the marginal probabilities. Assignments of ordered genotypes with $q_{\text{map}} \equiv \max Q(G_i^{l,p}, G_i^{l,m} | \mathbf{M}, \mathbf{G}_{\text{assigned}}) < 1$ are shown in boldface. Ordered genotypes that were not assigned are represented by '-'. The reconstructed ordered genotypes are identical to the true ordered genotypes.

Table 2: Overview of data sets analyzed

	pedigree	markers	individuals	genotyped	dist. ¹	MMAF ²	avg. spacing (cM)
A	I	5	53	13	real	0.31	0.312
B	I	5	53	30 %- 90 %	random	0.31	0.312
C	I	20	53	13	real	0.24	0.601
D	I	200	53	13	real	0.24	0.601
E	I	20	53	13	real	0.34	0.601
F	II	8	368	262	real	0.28	0.012
G	IIsub	8	22	30 %- 90 %	random	0.28	0.012
H	IIsub	8	22	16	real	0.28	0.012

¹ Distribution of genotyped individuals; 'real' indicates as in real data set, 'random' indicates randomly assigned individuals

² MMAF: Mean Minor Allele Frequency

Table 3: Comparison of CVMHAPLO with exact maximum likelihood methods

percentage genotyped individuals	full haplotype reconstruction				partial reconstruction ¹	
	accuracy ²		log-likelihood		accuracy	percentage assigned
	exact ML	CVMHAPLO	exact ML	CVMHAPLO	CVMHAPLO	CVMHAPLO
Pedigree I						
90	97.20	97.53	-79.01	-79.06	99.47	92.25
80	95.69	95.92	-77.51	-77.69	99.51	85.98
70	92.46	92.97	-76.34	-76.45	99.38	74.76
60	89.26	89.95	-74.80	-75.23	99.48	60.72
50	83.99	84.46	-72.18	-73.35	99.16	38.62
40	79.93	81.33	-70.19	-71.64	99.59	31.49
30	77.57	77.80	-66.00	-67.14	99.43	20.31
Pedigree IIsub						
90	98.16	98.44	-33.51	-33.51	99.01	98.05
80	96.25	96.13	-33.50	-33.53	99.35	89.90
70	94.87	94.63	-33.51	-33.51	99.31	84.94
60	93.67	94.00	-32.60	-32.67	99.71	79.55
50	91.06	91.93	-31.97	-32.09	99.60	70.72
40	87.17	87.54	-32.07	-32.13	99.41	55.07
30	83.91	83.35	-31.14	-32.79	99.53	37.61

Note: All values are reported as means over 40 replicates

¹ The partial haplotype configuration $\mathbf{G}_{\text{assigned}}^{(n)}$ obtained from the iteration n where the confidence from eq. (1) was 99 %.

² Accuracy is defined as the percentage of assigned ordered genotypes equal to the true simulated ordered genotype.

Table 4: Comparison of computation time and memory usage

Pedigree markers	Computation time (s)				Memory usage (MB)			
	CVM	CVMHAPLO	SIMWALK2	EXACT	CVMHAPLO	SIMWALK2	EXACT	
I ^{1,2}	5	26	542	NA ⁴	329	18	NA	1230
I ²	5	26 ± 1	530 ± 21	NA	307 ± 29	18 ± 0.3	NA	750 ± 233
IIsub ³	8	6 ± 1	185 ± 27	NA	< 1	6 ± 0.5	NA	< 1
IIsub ^{1,3}	8	9	165	NA	< 1	6	NA	< 1
I	20	187 ± 6	2977 ± 887	≈ 2400	NF ⁵	85 ± 1	10	NF
I	200	2520 ± 43	16.2 h ± 9.6 h	> 280 h	NF	944 ± 2	23	NF
II	8	568 ± 32	2.2 h ± 0.12 h	4.5 d ± 1.7 d	NF	150 ± 3	15	NF
II ¹	8	572	2.9 h	5 d	NF	151	15	NF

¹ Real data set² Exact results computed with junction tree algorithm³ Exact results computed with SUPERLINK⁴ NA: Simulations not performed as exact computation was feasible⁵ NF: Exact computation not feasible