

Fast Bayesian Haplotype Inference via Context Tree Weighting

Pasi Rastas, Jussi Kollin, and Mikko Koivisto*

Department of Computer Science & HIIT Basic Research Unit
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FIN-00014 University of Helsinki, Finland
`firstname.lastname@cs.helsinki.fi`

Abstract. We present a new, Bayesian method for inferring haplotypes for unphased genotypes. The method can be viewed as a unification of some ideas of variable-order Markov chain modelling and ensemble learning that so far have been implemented only separately in some of the state-of-the-art methods. Specifically, we make use of the Context Tree Weighting algorithm to efficiently compute the posterior probability of any given haplotype assignment; we employ a simulated annealing scheme to rapidly find several local optima of the posterior; and we sketch a full Bayesian analogue, in which a weighted sample of haplotype assignments is drawn to summarize the posterior distribution. We also show that one can minimize in linear time the average switch distance, a popular measure of phasing accuracy, to a given (weighted) sample of haplotype assignments. We demonstrate empirically that the presented method typically performs as well as the leading fast haplotype inference methods, and sometimes better. The methods are freely available in a computer program BACH (Bayesian Context-based Haplotyping)

1 Introduction

Large-scale genotyping – that is, measuring the genomic variation at hundreds to millions of marker loci for tens to thousands of subjects – has become a common approach to the genetic mapping of complex traits and to the discovery of the genomic structure and variation in general. The abundance of single-nucleotide polymorphisms (SNPs) in the human genome, in particular, may enable powerful association analyses as well as detecting larger chromosomal rearrangements, such as deletion [1–3] and inversion [4] polymorphisms, using efficient statistical and computational methods.

Crucial to methods that utilize multilocus genotypes is the modelling of *haplotypes*, the maternal and paternal material that constitute a genotype. As haplotypes tend to be inherited as large blocks, broken only occasionally by recombinations, they carry important information about ancient single-point mutations and larger-scale structural changes. Unfortunately, the usual genotyping technologies cannot reliably measure the haplotypes *per se*, but only the unphased

* Supported by the Academy of Finland under Grant 109101.

genotypes, that is, for any two loci it cannot be determined which of the measured alleles belong to the same haplotype, be it maternal or paternal. Thus, haplotypes appear as central latent variables in genotype analysis methods.

While the precise use of haplotypes should depend on the particularities of the data analysis problem at hand, it is plausible to test a model of haplotypes in the somewhat isolated *problem of haplotype inference*: given a set of multilocus genotypes, find good estimates of the underlying haplotype pairs, called a *haplotype assignment*. Indeed, this problem has attracted much attention in the recent years, and a variety of methods have been proposed.

The state-of-the-art methods for haplotype inference are based on various ideas. Since Clark’s [5] rule-based approach, several likelihood based methods that estimate “independent” haplotype frequencies over a short window of markers using an expectation–maximization (EM) algorithm or using related Gibbs sampling have been presented [6–9]; a sort of divide and conquer technique, called partition ligation, is commonly employed to handle larger numbers of markers. A more sophisticated method is implemented in HAP [10], based on fitting a perfect or almost perfect phylogeny model to short overlapping marker windows and solving conflicts to optimality using dynamic programming. Of the existing methods, the most accurate is perhaps PHASE [8]. It is a Bayesian Markov chain Monte Carlo method that samples haplotypes from a posterior distribution defined by a mosaic model, in which every haplotype is modelled as a concatenation of fragments extracted (fairly independently) from the other haplotypes. A drawback of PHASE is that it is impractically slow on large genotype samples. Three faster methods based on Hidden Markov models and sophisticated EM algorithms, HIT [11], GERBIL [12], and fastPHASE [13], were developed independently. Of these methods, fastPHASE, which uses an ensemble learning technique, seems the most accurate on average. In another direction, fast and accurate phasing methods have been built on variable-order Markov models that can be efficient in capturing rare higher-order dependencies. Such models are implemented in HaploRec [14] based on fast search for frequently occurring haplotype fragments, and more recently in Beagle [15] based on efficient estimation of related probabilistic automata and an EM-type search.

In this paper, we present a new haplotype inference method that can be viewed as a principled and unified treatment of certain key ideas underlying some of the state-of-the-art methods. Like in PHASE, we take a Bayesian approach in which haplotype inference relies on the posterior distribution of haplotype assignments. However, we model haplotypes using a variable order Markov model similar to those used in HaploRec and Beagle. Instead of fitting the model to given haplotypes, we show that the Bayesian approach of averaging over all the model parameters, including the context trees, can be done efficiently by using the celebrated Context Tree Weighting algorithm (CTW), originally developed for efficient data compression [16]. The remaining challenge is to efficiently sample haplotype assignments proportionally to the posterior, or, alternatively, to maximize the posterior probability. To this end, we consider two kinds of local moves in the space of haplotype assignments: a forward sampling of a haplotype

pair according to the given genotype and a simpler phase switch operation. Then, in the spirit of the ensemble EM technique of fastPHASE, we employ a simulated annealing procedure to rapidly find several local optima of the posterior distribution. We also sketch an analogous full-Bayesian method based on annealed importance sampling [17], which outputs a sample of independently drawn haplotype assignments, each associated with a real-valued weight. (An efficient implementation of this full-Bayesian method is work-in-progress.) Finally, we show that one can find in linear time a Bayes-optimal haplotype assignment, that is, one that minimizes the average switch distance, a popular measure of phasing accuracy, to a given (weighted) sample of haplotype assignments. This useful observation is technically rather simple and has appeared in other guises earlier: an unweighted, non-Bayesian setting is implemented in fastPHASE [13] and treated more formally recently [18]. An implementation of the methods is freely available in a computer program BACH (**B**ayesian **C**ontext-based **H**aplotyping).

Using the HapMap data [19] and simulated data, we have empirically compared the phasing accuracy of BACH against PHASE, fastPHASE, HAP, HIT, HaploRec, and Beagle. The results agree with earlier observations that PHASE, when computationally feasible, is generally the most accurate. Of the fast methods, HaploRec and BACH performed the best on average. Our study also contributes to mutual comparison of the aforementioned current leading methods, many of which were not included in some earlier comparison studies [20].

2 A Bayesian Variable Order Markov Model

Consider m SNPs labelled in their physical order by the numbers $1, 2, \dots, m$. We assume that at each SNP in the population there occur two alleles, which we denote by 0 and 1. A haplotype is a sequence $h_1 \cdots h_m$ where $h_j \in \{0, 1\}$ is an allele at SNP j . Two haplotypes h and h' determine an unphased genotype $g(h, h') = g = g_1 \cdots g_m$, where each single-SNP genotype g_j takes value 0 if $h_j = h'_j = 0$, value 1 if $h_j = h'_j = 1$, and value 2 otherwise.

We aim at a model that given n genotypes $\mathbf{g} = (g^1, \dots, g^n)$ yields good estimates of the underlying $2n$ haplotypes $\mathbf{h} = (h^1, h^2, \dots, h^{2n-1}, h^{2n})$, called a *haplotype assignment*; we denote the two haplotypes underlying g^i by h^{2i-1} and h^{2i} . We take a Bayesian approach and define a joint probability distribution of \mathbf{g} and \mathbf{h} , from which the posterior distribution of the haplotypes is obtained by conditioning on \mathbf{g} . Based on the posterior distribution, “optimal” estimates of haplotype frequencies as well as individual haplotype pairs can be determined; see Sect. 5.

We begin with a basic decomposition:

$$p(\mathbf{h}|\mathbf{g}) \propto p(\mathbf{h}, \mathbf{g}) = p(\mathbf{g}|\mathbf{h})p(\mathbf{h}).$$

Here the first term is easily specified: it evaluates to 1 if for every i , the genotype g^i is the unique genotype determined by h^{2i-1} and h^{2i} , and to 0 otherwise.

The crucial part is the modelling of the latter term, the joint distribution of the $2n$ haplotypes. We next derive a variable-order Markov model, starting with

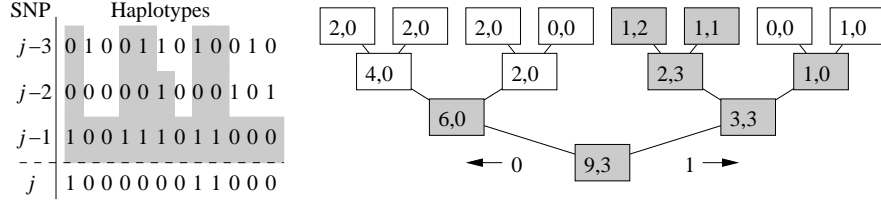


Fig. 1. The full context tree of marker j with maximum depth $D = 3$ for a sample of 12 haplotypes. Each node in the tree corresponds to a suffix. The numbers in each node are the empirical counts of 0s and 1s for marker j following the corresponding suffix. The gray nodes form a smaller context tree. The corresponding suffixes in the sample are shown on gray background.

the chain rule that holds for any probability distribution:

$$p(\mathbf{h}) = \prod_{j=1}^m p(\mathbf{h}_j | \mathbf{h}_1, \dots, \mathbf{h}_{j-1});$$

here and henceforth \mathbf{h}_j denotes the alleles of the haplotypes at SNP j , that is, $\mathbf{h}_j = (h_j^1, h_j^2, \dots, h_j^{2n-1}, h_j^{2n})$. The variable-order Markov model makes the restrictive, yet useful and biologically plausible assumption that \mathbf{h}_j depends only on a relatively small context, i.e., the preceding markers. This assumption is reflected with linkage disequilibrium decreasing with marker distances.

We make the usual assumption that these contexts, or suffixes, form the leaves of a *context tree*. A context tree is a rooted tree where a node at depth d is a sequence $u = h_{j-d} \cdots h_{j-1} \in \{0, 1\}^d$; the node is either a leaf or the *parent* of its two *children*, $0u$ and $1u$. Thus, a context tree is uniquely determined by the leaves of the tree. See Fig. 1 for an illustration.

More formally, we specify a context tree by a function c_j that with each partial haplotype $h_1 \cdots h_{j-1}$ associates a suffix haplotype $h_{j-d} \cdots h_{j-1}$, a leaf of the context tree, with some context length d ; we denote the set of these suffixes by $S_j = S_j(c_j)$. Let $c_j(\mathbf{h}_1 \cdots \mathbf{h}_{j-1})$ denote the composition of the suffixes over all the $2n$ partial haplotypes. By marginalizing over the unknown function c_j with respect to its prior distribution, we obtain

$$p(\mathbf{h}_j | \mathbf{h}_1, \dots, \mathbf{h}_{j-1}) = \sum_{c_j} p(c_j) p(\mathbf{h}_j | c_j(\mathbf{h}_1 \cdots \mathbf{h}_{j-1})). \quad (1)$$

We have implemented a simple prior over the context trees by restricting the maximum depth of a tree to D and assigning a probability $(1/2)^N$ to each tree with N nodes at depth less than D . This prior is the same as in the original CTW method and stems from the idea of a recursive construction of a context tree by either splitting or stopping at each node, independently, with equal probabilities. The prior for stopping the tree construction could also have been set for each marker separately to reflect the effect of the marker distances.

To complete the model specification, it remains to specify the term $p(\mathbf{h}_j | c_j(\mathbf{h}_1 \cdots \mathbf{h}_{j-1}))$. To this end, we associate with each leaf s of the context tree c_j a parameter θ_s that gives the probability that a haplotype has a 1 at marker j , given that its suffix up to marker $j - 1$ is s . Considering these parameters independent a priori and assigning each a Beta(1/2, 1/2) prior yields a closed-form expression [21]:

$$p(\mathbf{h}_j | c_j(\mathbf{h}_1 \cdots \mathbf{h}_{j-1})) = \prod_{s \in S_j} \rho(a_s, b_s),$$

where a_s and b_s are the counts of haplotypes up to marker j with suffixes s_0 and s_1 , respectively, and the *leaf score* $\rho(a_s, b_s)$ can be written [21] as

$$\rho(a_s, b_s) = \frac{\Gamma(\frac{1}{2} + a_s) \Gamma(\frac{1}{2} + b_s)}{\Gamma(\frac{1}{2}) \Gamma(\frac{1}{2}) \Gamma(1 + a_s + b_s)} = \frac{\frac{1}{2} \cdot \frac{3}{2} \cdot \frac{5}{2} \cdots (a_s - \frac{1}{2}) \cdot \frac{1}{2} \cdot \frac{3}{2} \cdots (b_s - \frac{1}{2})}{(a_s + b_s)!}.$$

We have also experimented with a variant of the leaf score, defined as

$$\tilde{\rho}(a_s, b_s) = \left(\frac{a_s + \frac{1}{2}}{a_s + b_s + 1} \right)^{a_s} \left(\frac{b_s + \frac{1}{2}}{a_s + b_s + 1} \right)^{b_s}.$$

Like $\rho(a_s, b_s)$, this score can be interpreted as the probability of observing a_s suffixes s_0 and b_s suffixes s_1 , but now with a fixed estimate of the θ_s parameter (rather than integrating θ_s out). Intuitively, $\tilde{\rho}(a_s, b_s)$ may perform better than $\rho(a_s, b_s)$ when the assumed Beta prior fits poorly with the observed data. Somewhat surprisingly, in our preliminary experiments $\tilde{\rho}(a_s, b_s)$ yielded consistently better results than $\rho(a_s, b_s)$, a phenomenon we currently do not fully understand. In Sect. 6 we report results only for $\tilde{\rho}(a_s, b_s)$.

3 The Context Tree Weighting Algorithm

We apply the CTW algorithm [16] – originally developed for compressing a single binary string – to efficiently evaluate the sum over context trees for haplotype (1). The idea is to compute for each possible haplotype suffix s the sum of scores of all possible subtrees rooted at s ; denote this sum by ρ_s . If the length of s is the maximum D , then s must be a leaf and ρ_s is set to $\rho(a_s, b_s)$. Otherwise, ρ_s is obtained by averaging over the case that s is a leaf of a context tree and the case that it is has two children, $0s$ and $1s$:

$$\rho_s := \frac{1}{2} \rho(a_s, b_s) + \frac{1}{2} \rho_{0s} \rho_{1s}. \quad (2)$$

It is easy to show [16] that this recurrence yields

$$\rho_\lambda = \sum_{c_j} 2^{-N(c_j)} \prod_{s \in S_j(c_j)} \rho(a_s, b_s) = p(\mathbf{h}_j | \mathbf{h}_1, \dots, \mathbf{h}_{j-1}),$$

where λ denotes the empty sequence, i.e., the root of a context tree, and $N(c_j)$ is the number of nodes at depth less than D in c_j .

The algorithm can be implemented to run in time $O(nD)$ for a single marker j . First, one builds a trie of substrings starting at $j - 1$ backwards, as shown in Fig. 1. For each node s in this trie, the corresponding counts a_s and b_s can be computed according to the values on marker j ; this takes time $O(nD)$. Similarly, the values $\rho(a_s, b_s)$ for all nodes s can be computed in an incremental fashion in time $O(nD)$. Finally, the recurrence (2) can be solved along the trie in time $O(nD)$. Note that only the haplotype suffixes that are present in the data need to be considered (as $\rho(0, 0) = 1$, the subtrees with zero counts will average to 1).

4 Finding Plausible Haplotypes Using Local Search

The posterior probability $p(\mathbf{h}|\mathbf{g})$ of a haplotype assignment \mathbf{h} , obtained by averaging over all possible context trees and the involved parameters, provides a Bayesian measure for the plausibility of \mathbf{h} . Ideally, we would like to explore the entire posterior distribution to fully take into account the uncertainty about each single assignment. Unfortunately, this seems computationally infeasible. What seems more practical is to quickly find several local optima, which then, when combined, provide a reasonable summary of the posterior distribution.

4.1 Local Search and Simulated Annealing

To maximize the posterior $p(\mathbf{h}|\mathbf{g})$, we have implemented a simulated annealing method that starts with an arbitrary haplotype assignment and then proceeds iteratively with simple *phase switch* moves that change the haplotype pair $(h, h') = (h^{2i-1}, h^{2i})$ for a randomly picked genotype g^i .

Phase Switch: Select the pair of haplotypes (h, h') for a random genotype g and a random marker j and switch the phase of the partial haplotypes of h and h' at markers $k \geq j$. That is, the new haplotype pair for g is $(h_1 \cdots h_{j-1} h'_j \cdots h'_m, h'_1 \cdots h'_{j-1} h_j \cdots h_m)$.

A proposed phase switch is accepted with probability $\min\{1, A^{1/\tau}\}$, where A is the ratio of the posterior probabilities of the proposed and the current haplotype assignment, and $\tau \geq 0$ is a decreasing temperature parameter. Each such an iteration can be computed in time $O(D^2)$ by storing the haplotype pair of each genotype as its switch sequence (see Sect. 5) and maintaining the tries used in the CTW algorithm for computing $p(\mathbf{h}|\mathbf{g})$. The posterior is evaluated with a large depth, $D = 40$; temperature τ is set to 1 and the best assignment found in mn steps is taken as the starting assignment in the next batch. Three more such batches are run with $\tau = 1/2, 1/4, 0$, each batch started from the best assignment found so far. The procedure is repeated for some number of times, 20 in our experiments, to obtain several local optima of the posterior, with the genotypes reversed in every other iteration as in Beagle [15].

We have found that the above described local search converges rapidly to a local optimum, but also that the quality of the optimum is highly sensitive to the initial haplotype assignment. We therefore designed a slower but otherwise more effective *forward sampling* heuristic for finding good initial assignments.

Forward Sampling:

1. Set the haplotypes \mathbf{h} arbitrarily such that they match the genotypes \mathbf{g} .
2. For each genotype g^i , draw a new haplotype pair (h^{2i-1}, h^{2i}) from $p(h^{2i-1}, h^{2i} | \mathbf{h}^{-i}, \mathbf{g}) \propto p(\mathbf{h} | \mathbf{g})$, where \mathbf{h}^{-i} consists of the remaining haplotype pairs in \mathbf{h} . Repeat this 10 times.
3. Return the centroid (see Sect. 5) of the last 5 samples.

Forward sampling can be implemented relatively efficiently using the idea of forward-backward sampling algorithm of hidden Markov models (see [15] and references therein). As that HMM algorithm is well-known, we here only mention some key features of the present application and omit most of the technical details. First, observe that given a genotype g^i , it suffices to consider drawing the haplotype h^{2i-1} , since h^{2i} is fully determined by g^i and h^{2i-1} . Second, notice that the j th allele of h^{2i-1} depends on the alleles at markers $j-1, j-2, \dots, j-D$; the state of these D alleles corresponds to a hidden state in an analogous HMM. The time complexity of the sampling procedure essentially depends on the number of possible states, 2^D ; the effect of the number of markers is linear, and the involved transition probabilities are easily obtained by statistics precomputed by the CTW algorithm for computing $p(\mathbf{h} | \mathbf{g})$. Thus, sampling a haplotype pair can be implemented in time $O(mD2^D)$ for a single genotype. We also note that the term 2^D can be improved to 2^k , where k is the maximum number of heterozygous sites of g in any D consecutive markers.

4.2 Sampling Haplotypes from the Posterior Distribution

We next sketch an annealed importance sampling (AIS) method [17] to draw haplotype assignments from the posterior distribution $p(\mathbf{h} | \mathbf{g})$ conditional on the given n genotypes, \mathbf{g} . In this simulated annealing type MCMC technique, the idea is to generate a sequence $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \dots, \mathbf{h}^{(T)}$ along a smooth chain of distributions $\pi^{(1)}, \pi^{(2)}, \dots, \pi^{(T)}$. Only the last configuration, $\mathbf{h}^{(T)}$, will be retained with an appropriately defined weight $w(\mathbf{h}^{(T)})$; the procedure is repeated some number of times to obtain a collection \mathcal{H} of independent weighted samples. For the posterior expectation of any function of interest, $\psi(\mathbf{h})$, an unbiased estimate is obtained as $\sum_{\mathbf{h} \in \mathcal{H}} w(\mathbf{h}) \psi(\mathbf{h}) / \sum_{\mathbf{h} \in \mathcal{H}} w(\mathbf{h})$ [17].

For concreteness, consider a simple annealing scheme of the form $\pi^{(t)}(\mathbf{h}) \propto f^{(t)}(\mathbf{h}) = p(\mathbf{h} | \mathbf{g})^{t/T}$, with, say, $T = 1000$. First, $\mathbf{h}^{(1)}$ is drawn from $\pi^{(1)}$ by running a Metropolis algorithm, say, mn steps; suitable proposal distributions can be constructed based on the forward sampling and phase switch moves described above. Then, for $t = 1, \dots, T-1$, an assignment $\mathbf{h}^{(t+1)}$ is drawn based on $\mathbf{h}^{(t)}$ using an analogous Metropolis kernel that now leaves $\pi^{(t+1)}$ invariant. Finally, the last configuration, $\mathbf{h}^{(T)}$, is stored together with a weight $w(\mathbf{h}^{(T)}) = \prod_{t=1}^{T-1} f^{(t+1)}(\mathbf{h}^{(t)}) / f^{(t)}(\mathbf{h}^{(t)}) = (p(\mathbf{h}^{(1)})p(\mathbf{h}^{(2)}) \dots p(\mathbf{h}^{(T)}))^{1/T}$.

5 Minimization of the Expected Switch Distance

In principle, the output of Bayesian inference of haplotypes is the posterior distribution. In practice, and especially for comparing different methods, one however needs to provide a single estimate for the haplotype pair underlying each observed genotype. How such an estimate should be constructed depends not only on the posterior but also on the cost assigned for mistakes in the estimate. The cost can usually be specified by a *loss function*, ℓ , that associates two haplotypes pairs, $\{h, h'\}$ and $\{\hat{h}, \hat{h}'\}$, a real-valued loss $\ell(\{h, h'\}, \{\hat{h}, \hat{h}'\})$; the total cost over all genotypes is assumed to be a sum of the individual losses. Given a loss function, a Bayes-optimal estimate is one that minimizes the posterior expected loss. When a posterior is summarized by a (weighted) sample of haplotype assignments, the expected loss is approximated by a (weighted) average over the sample.

While it is often not possible to reliably infer the correct haplotype pair, except for very short regions, it is usually possible to provide an estimate that is close to the true haplotypes in terms of the switch distance [22], a loss function commonly used in haplotype inference. The *switch distance* of two haplotype pairs $\{h, h'\}$ and $\{\hat{h}, \hat{h}'\}$, denoted by $\text{sd}(\{h, h'\}, \{\hat{h}, \hat{h}'\})$, is the number of phase switches needed to turn $\{h, h'\}$ into $\{\hat{h}, \hat{h}'\}$. For example, if the true pair is $\{000000, 111111\}$ then the switch distance to $\{000111, 111000\}$ is 1 while the switch distance to $\{000110, 111001\}$ is 2.

Even though finding a Bayes-optimal estimate seems hard in general, it turns out that the special case of switch distance can be solved efficiently, in time linear in the sample size. To see this, first observe that any haplotype pair $\{h, h'\}$ compatible with a fixed genotype g is bijectively related to its switch sequence $\xi = (\xi_1, \dots, \xi_{k-1})$ defined by $\xi_s = |\lambda_s - \lambda_{s+1}|$, where k is the number of heterozygous sites of g and $\lambda_s \in \{0, 1\}$ encodes (in an arbitrary way) the two possible phases of the s th heterozygous site of g ; furthermore, the switch sequence can be constructed in linear time. Second, we observe that if $\{h, h'\}$ and $\{\hat{h}, \hat{h}'\}$ are two haplotype pairs compatible with g , and ξ and $\hat{\xi}$ their switch sequences, then $\text{sd}(\{h, h'\}, \{\hat{h}, \hat{h}'\}) = \sum_{s=1}^k |\xi_s - \hat{\xi}_s| = \|\xi - \hat{\xi}\|_1$, the 1-norm of the vector $\xi - \hat{\xi}$. Thus, it remains to find a switch sequence $\hat{\xi}$ that minimizes

$$\sum_{\xi \in \Xi} w(\xi) \|\xi - \hat{\xi}\|_1,$$

where Ξ denotes the multiset of switch sequences determined by the given sample of haplotype pairs, and $w(\xi)$ is the weight of the haplotype assignment that corresponds to ξ . Now, we know that for any set of binary sequences the centroid with respect to the metric induced by the 1-norm is obtained simply by coordinate-wise voting. This suffices for the unweighted case; see [18] for a somewhat more detailed argumentation. Having this, it is not difficult to see that the weighted case can be solved by weighted voting: set $\hat{\xi}_s$ to 1 if and only if

$$\sum_{\xi \in \Xi: \xi_s=1} w(\xi) > \sum_{\xi \in \Xi: \xi_s=0} w(\xi).$$

Such a voting can obviously be carried out in time $O(k|\Xi|)$. Combining these observations gives

Theorem 1. *Let g be a genotype over m SNPs and \mathcal{H} a set of haplotype pairs that are compatible with g , each pair associated with a real-valued weight. Then a centroid, that is, a haplotype pair that minimizes the sum of the weighted switch distances to the haplotype pairs in \mathcal{H} , can be found in time $O(m|\mathcal{H}|)$.*

6 Experimental Results

We have implemented the presented method in a prototype program BACH (Bayesian Context-based Haplotyping) and compared its performance against state-of-the-art software for haplotype inference on both real and simulated data.

6.1 Methods Compared

BACH is written in Java and the implementation is not optimized for speed. Currently, it contains only the simulating annealing method (with forward sampling and phase switch using tuning parameters $D = 8$ and $D = 40$, respectively); the annealed importance sampling analogue sketched in Sect. 4.2 is work-in-progress.

For comparison we included the latest versions of PHASE [23], fastPHASE [13], Beagle [15], HaploRec [14], HIT [11], and HAP [10]. We included PHASE as a reference, even though it took several hours to run on a single small (HapMap) dataset; we did not try to run it for the much larger, simulated datasets. The other tested methods scale well and take only some minutes or a few hours per large dataset. Specifically, the time complexity of BACH (with fixed maximum context length) is only linear, $O(mn)$; relatively large constant factors, however, render BACH the slowest of the fast methods. All methods except HAP were available as stand-alone applications, which we ran on a regular desktop PC. HAP was run on its web server, and due the amount of manual labour involved in sending the data and fetching the results, we tested HAP only on 24 of the the real datasets, selected at random.

We also tested deviating from the default parameters of the compared methods, and report the best results obtained. For example, we set “-nsamples=25” in Beagle for the real data, as suggested by its manual; this improved Beagle’s performance slightly. For HaploRec we used option “-p S”, here referred to as HaploRec-S, to use a segmentation model. By default, HaploRec uses a variable-order Markov model (option “-p VMM_AVG”), here referred to as HaploRec-V. For HIT, we used ten founders ($K = 10$).

6.2 Datasets

The real data, containing 132 datasets, were obtained from the HapMap database [19]. These datasets contain samples from human populations YRI (Yoruba) and CEU (Utah), from which there were 30 SNP trios available. The HapMap

database contains 120 haplotypes inferred from these trios, which we took as the ground truth and converted them into 60 genotypes. From each chromosome we selected 100 SNPs in such a way that the average distance between adjacent SNPs is close to either 1, 3 or 9 kb. Thus, by CEU-3k (other combinations analogously) we refer to the datasets from the CEU population with an average SNP spacing of 3 kb.

To generate larger datasets, we followed the example of Browning and Browning [15] and used COSI [24] with the best-fit parameters to simulate 2,000 haplotypes of 1 Mb in length from a "European" population. We generated 100 sparse and dense datasets. The sparse datasets were filtered by first removing all SNPs with minimum allele frequency (MAF) ≤ 0.05 , and then selecting at random 250 SNPs. We then sampled a subset of 120 haplotypes, and again eliminated SNPs with MAF ≤ 0.05 . The SNPs were then selected to the final data set by selecting an informative subset by the method of Carlson et al. [25]. In the resulting set of SNPs, either the SNP was genotyped or there was at least one genotyped SNP that had the squared correlation coefficient $r^2 \geq 0.7$ with the omitted SNP. For dense sets the filtering was similar, except that we sampled 1,428 markers instead of 250, and the r^2 threshold for the tagging algorithm was 0.9. These datasets are referred to as Dense and Sparse. The former had median SNP count of 101, and the latter 367 markers.

6.3 Comparison of Phasing Accuracy

We evaluated the phasing accuracy of each method in terms of the switch error, that is, the switch distance between the predicted and the true haplotypes (see Sect. 5) divided by the maximum switch distance. We found that by average switch error (Table 1) BACH is among the most accurate methods on both the HapMap datasets and the synthetic datasets. On the HapMap datasets PHASE is superior, whereas HAP (on the selected 24 datasets) is consistently less accurate than the rest (details not shown).

We also compared the methods pairwise and examined the percentage of the HapMap datasets on which one method was more accurate than another method (Table 2). We observed that, after PHASE, the most accurate methods were fastPHASE, BACH, and HaploRec-S, with no clear order. Note that HAP and the synthetic datasets were not included in this comparison.

7 Concluding Remarks

We have presented a Bayesian implementation of variable-order Markov modeling of haplotypes. The promise of this approach is in its robustness, as it is not based of fitting a single model to the data. As we showed, the required sum over models can be efficiently computed using the context tree weighting algorithm [16]. We believe there is room for improving our heuristics for optimizing or exploring the resulting objective function (the posterior distribution). Yet, the

Table 1. Average switch errors of the tested methods

	PHASE	fastPHASE	BACH	Beagle	HaploRec-S	HaploRec-V	HIT
CEU-1k	0.0299	0.0343	0.0348	0.0405	0.0364	0.0376	0.0375
CEU-3k	0.0652	0.0692	0.0665	0.0764	0.0692	0.073	0.0745
CEU-9k	0.144	0.146	0.147	0.164	0.147	0.153	0.159
YRI-1k	0.0407	0.0579	0.0597	0.0645	0.0540	0.0601	0.0642
YRI-3k	0.0931	0.117	0.113	0.125	0.111	0.122	0.126
YRI-9k	0.189	0.204	0.198	0.223	0.193	0.204	0.220
Sparse	-	0.0398	0.0305	0.0317	0.0288	0.0316	0.0442
Dense	-	0.0169	0.0125	0.0116	0.0133	0.0145	0.0190

Table 2. Percentage of HapMap datasets won by a method on a column vs. a method on a row.

	PHASE	fastPHASE	BACH	Beagle	HaploRec-S	HaploRec-V	HIT
PHASE	-	20.5	19.7	5.3	18.9	9.1	7.6
fastPHASE	78.0	-	48.5	17.4	55.3	34.8	15.2
BACH	79.5	45.5	-	18.2	53.8	28.0	15.2
Beagle	93.9	79.5	79.5	-	87.1	72.7	55.3
HaploRec-S	78.8	42.4	42.4	12.9	-	25.8	16.7
HaploRec-V	90.2	59.8	71.2	23.5	73.5	-	29.5
HIT	92.4	82.6	82.6	37.9	78.8	67.4	-

techniques implemented in BACH were sufficient for demonstrating the potential of the approach: the phasing accuracy of BACH was competitive to the very best of the existing fast haplotype inference methods.

The context tree weighting algorithm has been previously successfully applied to protein classification [26]. In this application, a variable-order Markov model was extended to wild-card symbols that match to every alphabet symbol. Such wild-cards could be incorporated into our haplotype inference method as well.

References

1. Conrad, D.F., Andrews, T.D., Carter, N.P., Hurles, M.E., Pritchard, J.K.: A high-resolution survey of deletion polymorphism in the human genome. *Nat. Genet.* **38** (2006) 75–81
2. Corona, E., Raphael, B.J., Eskin, E.: Identification of deletion polymorphisms from haplotypes. In: *Research in Computational Molecular Biology (RECOMB '07)*, Springer (2007) 354–365
3. Kohler, J.E., Cutler, D.J.: Simultaneous discovery and testing of deletions for disease associations in SNP genotyping studies. *Am. J. Hum. Genet.* **81** (2007) 684–699
4. Bansal, V., Bashir, A., Bafna, V.: Evidence for large inversion polymorphisms in the human genome from HapMap data. *Genome Res.* **17** (2007) 219–230
5. Clark, A.G.: Inference of haplotypes from PCR-amplified samples of diploid populations. *Mol. Biol. Evol.* **7** (1990) 111–122

6. Excoffier, L., Slatkin, M.: Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol. Biol. Evol.* **12** (1995) 921–927
7. Long, J.C., Williams, R.C., Urbanek, M.: An E-M algorithm and testing strategy for multiple-locus haplotypes. *Am. J. Hum. Genet.* **56** (1995) 799–810
8. Stephens, M., Smith, N., Donnelly, P.: A new statistical method for haplotype reconstruction from population data. *Am. J. Hum. Genet.* **68** (2001) 978–989
9. Niu, T., Qin, Z., Xu, X., Liu, J.: Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *Am. J. Hum. Genet.* **70** (2002) 157–169
10. Halperin, E., Eskin, E.: Haplotype reconstruction from genotype data using imperfect phylogeny. *Bioinformatics* **20** (2004) 104–113
11. Rastas, P., Koivisto, M., Mannila, H., Ukkonen, E.: A hidden Markov technique for haplotype reconstruction. In: *Algorithms in Bioinformatics (WABI 05)*, Springer (2005) 140–151
12. Kimmel, G., Shamir, R.: Genotype resolution and block identification using likelihood. *Proceeding of the National Academy of Sciences of the United States of America (PNAS)* **102** (2005) 158–162
13. Scheet, P., Stephens, M.: A fast and flexible statistical model for large-scale population genotype data: Applications to inferring missing genotypes and haplotypic phase. *Am. J. Hum. Genet.* **78** (2006) 629–44
14. Eronen, L., Geerts, F., Toivonen, H.: Haploreco: efficient and accurate large-scale reconstruction of haplotypes. *BMC Bioinformatics* **7** (2006) 542
15. Browning, S., Browning, B.: Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *Am. J. Hum. Genet.* **81** (2007) 1084–97
16. Willems, F.M.J., Shtarkov, Y.M., Tjalkens, T.J.: The context-tree weighting method: Basic properties. *IEEE Trans. Inform. Theory* **41** (1995) 653–664
17. Neal, R.M.: Annealed importance sampling. *Statist. Comput.* **11** (2001) 125–139
18. Kääriäinen, M., Landwehr, N., Lappalainen, S., Mielikäinen, T.: Combining haplotypes. Technical Report C-2007-57, Department of Computer Science, University of Helsinki (2007)
19. The International HapMap Consortium: A haplotype map of the human genome. *Nature* **437** (2005) 1299–1320
20. Marchini, J., Cutler, D., Patterson, N., et al: A comparison of phasing algorithms for trios and unrelated individuals. *Am. J. Hum. Genet.* **78** (2006) 437–450
21. Willems, F.M.J.: The context-tree weighting method : Extensions. *IEEE Trans. Inform. Theory* **44** (1998) 792–798
22. Lin, S., Cutler, D.J., Zwick, M.E., Chakravarti, A.: Haplotype inference in random population samples. *Am. J. Hum. Genet.* **71** (2002) 1129–37
23. Stephens, M., Scheet, P.: Accounting for decay of linkage disequilibrium in haplotype inference and missing-data imputation. *Am. J. Hum. Genet.* **76** (2005) 449–462
24. Schaffner, S.F., Foo, C., Gabriel, S., Reich, D., Daly, M.J., Altshuler, D.: Calibrating a coalescent simulation of human genome sequence variation. *Genome Res.* **15** (2005) 1576–1583
25. Carlson, C.S., Eberle, M.A., Rieder, M.J., Yi, Q., Kruglyak, L., Nickerson, D.A.: Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium. *Am. J. Hum. Genet.* **74** (2004) 105–120
26. Eskin, E., Grundy, W.N., Singer, Y.: Protein family classification using sparse markov transducers. In: *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, AAAI Press (2000) 134–145