ABSTRACT

Motivation: Information about haplotype structures gives a more detailed picture of genetic variation between individuals than single-locus analyses. Databases that contain the most frequent haplotypes of certain populations are developing rapidly (e.g. the HapMap database for single-nucleotide polymorphisms in humans). Utilization of such prior information about the prevailing haplotype structures makes it possible to estimate the haplotype frequencies also from large DNA pools. When genetic material from dozens of individuals is pooled together and analysed in a single genotyping, the overall number of genotyped pairs and the costs of the genetic studies are reduced.

Results: A Bayesian model for estimating the haplotypes and their frequencies from pooled allelic observations is introduced. The model combines an idea of using database information for haplotype estimation with a computationally efficient multinormal approximation. In addition, the model treats the number and structures of the unknown haplotypes as random variables whose joint posterior distribution is estimated. The results on real human data from the HapMap database show that the proposed method provides significant improvements over the existing methods.

Availability: A reversible-jump Markov chain Monte Carlo algorithm for analysing the model is implemented in a program called Hippo (haplotype estimation under incomplete prior information using pooled observations). For comparisons, an approximate expectation-maximization algorithm (EM-algorithm) that utilizes database information about the existing haplotypes is implemented in a program called AEML. The source codes written in C (using GNU Scientific Library) are available at www.iki.fi/~mprinen.

Contact: matti.pirinen@iki.fi

1 INTRODUCTION

In diploid species each individual inherits one copy of the genome from each of the two parents. The chromosomes originating from the same parent form a haplotype. Common genotyping techniques produce only unphased genotypes at each marker locus, thus, leaving the haplotype configurations undetermined between any pair of heterozygous loci. However, knowledge of the haplotype structures would be valuable because it provides more information than single locus analyses, for example, for understanding the functional effects of genetic variation and for studying the genetic relatedness between the individuals (Clark, 2004). There exist specific genotyping techniques such as somatic cell hybrids typing, sperm typing, allele-specific PCR, MALDI-TOF genotyping on DNA dilutions and polony PCR that directly yield haplotypes [see Ragoussis (2009) for references]. However, the usual approach is to complement the standard techniques for high-throughput genotyping that do not yield phase information, with statistical methods to infer the missing haplotypes. Several methods for haplotyping individually genotyped diploid genotypes are reviewed and compared by Niu (2004) and Marchini et al. (2006).

The idea behind DNA pooling techniques is to combine equal amounts of genetic material from several individuals and to determine the allele frequencies at certain marker loci in a single genotyping from the combined DNA (Sham et al., 2002). DNA pooling has been used to reduce the number of genotyped pairs in association analyses by making separate pools for the cases and the controls and then comparing the allele frequency differences between the groups. Macgregor et al. (2008) used modern SNP arrays (e.g. Illumina HumanHap300) to genotype DNA pools that contained almost 400 individuals and reported that concordance of the results with individual-level genotyping was excellent. Macgregor et al. (2008) also estimated that by using pooled data one-stage whole-genome association analyses (that are based only on the allele frequency differences between the groups) could be performed with more than 100-fold reduced costs compared with similar undertakings by individual-level genotyping. Thus, a strategy for a two-stage association analysis could be first to narrow down the interesting regions by using pooled data and then to confirm the results using individual-level genotyping within those regions.

Naturally, pooling techniques are more prone to errors and offer less possibilities for assessing quality of the data than individual-level genotyping. For example, deviations from Hardy-Weinberg equilibrium or individual-specific problems in genotyping cannot be identified. Also matching between cases and controls with respect to covariates becomes more difficult with pooled data. The most important source of errors in absolute allele frequency estimates from pooled data is the differential allele amplification, but by carefully adjusting the measurement equipment, allele frequencies can be estimated accurately (Jawaid and Sham, 2009).

When allele frequency data are available from large DNA pools, there may be an astronomical number of different haplotype combinations that are compatible with the data. For this reason, several methods that estimate the haplotype frequencies from DNA pools are restricted to small pool sizes. Such methods include an expectation-maximization algorithm.
Haplotype estimation from DNA pools using incomplete database information

In the future, there is room for studies on more biologically motivated priors for haplotype configurations, which might bring still more accuracy and robustness to the method.

2 MODEL

We are interested in estimating the population haplotype frequencies on a set of diallelic loci residing on a relatively narrow chromosomal interval (up to 100 kb in the examples of this article). We proceed by sampling DNA from \( n \) individuals (2n haplotypes) and by dividing the DNA into \( N \) separate pools in such a way that pool \( i \) contains DNA from \( n_i \) individuals (2n haplotypes). Thus, \( n = \sum_{i=1}^{N} n_i \). The two alleles at each locus are represented by 0 and 1. By genotyping each pool separately, we get the data vectors \( \mathbf{a} = (a_1, \ldots, a_L) \), where \( a_l \in \{0, \ldots, 2n\} \) is the observed frequency of allele 1 at locus \( l \) in pool \( i \).

We denote by \( \nu \) the unknown number of different types of haplotypes that are present in the considered sample (\( 2 \leq \nu \leq 2^L \)). Conditionally on \( \nu \), we let the columns of the (unknown) \( L \times \nu \) matrix \( \mathbf{A} = (a_{i,j}) \) contain the \( \nu \) haplotypes that are assumed to exist in the sample. If we have some information about the existing haplotypes in the population, for example, from a database such as HapMap, we denote by \( m \) the number of the known haplotypes and we let those haplotypes occupy the first \( m \) columns of the matrix \( \mathbf{A} \) in the alphabetical ordering. If the haplotype information is not complete then \( \nu > m \), and the remaining columns \( m+1, \ldots, \nu \) of \( \mathbf{A} \) contain the estimated additional haplotypes, again in the alphabetical ordering. The ordering among haplotypes is introduced because it is important that there is a one-to-one correspondence between the possible \( \nu \) and the possible sets of haplotypes when the probabilistic model is specified below. Conditionally on \( \nu \), we denote by \( \theta = (\theta_1, \ldots, \theta_\nu) \) the \( \nu \)-dimensional vector of relative population frequencies of the haplotypes.

Our goal is to study the joint posterior distribution of \( \nu, \mathbf{A} \) and \( \theta \) given the observations \( \mathbf{a} = (a_{i,j}) \), and the prior information about the haplotypes under a Bayesian model that is specified next.

2.1 Priors for \( \nu, \mathbf{A} \) and \( \theta \)

We use the prior

\[
p(\nu, \mathbf{A}, \theta) \propto \nu^{\nu-1} \prod_{i=1}^{\nu} \theta_i^{a_i-1} \prod_{i=1}^\nu \mathbf{1}(\nu, \mathbf{A}, \theta) \text{ consistent},
\]

where \( \nu \in \mathbb{R} \) and \( \nu > 0 \) are fixed parameters and the indicator restricts the considerations to the consistent configurations, i.e., \( m \leq \nu \leq 2^L \), the first \( m \) columns of \( \mathbf{A} \) contain the known haplotypes (in alphabetical ordering) and the remaining \( \nu - m \) columns contain the additional haplotypes (in alphabetical ordering), and

\[
\theta \in \left\{ (\theta_1, \ldots, \theta_\nu) : \sum_{i=1}^{\nu} \theta_i = 1, 0 < \theta_i < 1, \nu > 0 \right\}.
\]

Let us consider what this prior says marginally about different variables. Denote by \( \mathbf{A}^* \) the submatrix formed by columns \( m+1, \ldots, \nu \) of \( \mathbf{A} \), i.e. \( \mathbf{A}^* \) contains the additional haplotypes. For fixed values of \( \nu \) and \( \mathbf{A} \) the prior is constant for any choice of \( \mathbf{A}^* \) meaning that a prior \((\mathbf{A}^*)^{\nu} \Rightarrow \text{Uniform distribution on the sets of } \nu - m \text{ additional haplotypes.}

For fixed values of \( \nu \) and \( \mathbf{A} \) the prior for \( \theta \) is proportional to \( \prod_{i=1}^\nu \theta_i^{a_i-1} \), which is the (unnormalized) density function of a Dirichlet distribution. Thus, a priori,

\[
(\theta|\nu, \mathbf{A}) \sim \text{Dirichlet}(\mathbf{1}, \ldots, \mathbf{1}).
\]

In order to calculate the prior for \( \nu \), we need to sum over all \( \binom{L}{m} \) possible \( \mathbf{A}^* \)s that have \( \nu - m \) haplotypes and integrate over \( \theta \) which gives \( \Gamma(\nu')/\Gamma(\nu') \) (the inverse of the normalizing constant of the Dirichlet distribution of \( \theta \)). Thus, we have that a prior

\[
p(\nu|\mathbf{A}) \propto \binom{L}{m} \Gamma(\nu') \nu'^{-\nu'}.
\]

Guidelines for choosing appropriate values for \( \nu \) and \( \mathbf{A} \) are as follows. If it is likely that there are only a few haplotypes with considerable relative...
frequencies (say >5%), then one should choose a relatively small value for \( \alpha \) (e.g. \( \alpha = 10^{-5} \)), since that conveys the idea of sparseness to vector \( \theta \). After choosing \( \alpha \), one should adjust \( \nu \) in a way that reflects one’s conception about comprehensiveness of the list of the known haplotypes. If the list is supposed to include almost all existing haplotypes then \( \nu \) should be large, whereas a smaller \( \nu \) puts more weight on larger numbers of the additional haplotypes. In practice, one can tune the parameters by carrying out short test runs to see whether the model produces reasonably sparse solutions, and whether the number of additional haplotypes is in accordance with one’s prior thoughts.

2.2 Multinomial approximation of the likelihood

We denote by \( \ell \) the (unknown) proportion of haplotype \( A_k \) (i.e. column \( h \) of matrix \( A \)) in pool \( i \). It follows that \( \ell_k = \mu_k \), where \( \ell = (\ell_1, \ldots, \ell_N) \).

Instead of working with the exact algebraic solution sets of these systems of linear equations, as was done by Gasbarra et al. (2009), we turn to the multinomial approximation that was introduced by Zhang et al. (2008). The expected values \( (\mu_k) \) and variance matrices \( (\Sigma_k) \) of the observed poolwise allele counts \( (2n_{ai}) \) can be represented as

\[
\mu_k = E(2n_{ai}) = 2n_i \theta_k \\
\Sigma_k = Var(2n_{ai}) = 2n_i \Sigma_{ai} \theta_k (\theta_k - \theta_k^2) A_k^T A_k.
\]

The central limit theorem states that as \( n_i \) increases the multinomial approximation \( 2n_{ai} \rightarrow N(\mu_k, \Sigma_k) \) becomes more accurate, and therefore it is reasonable to use the corresponding likelihood function for inferences about \( \theta \) and \( \Lambda \). For certain values of \( \Lambda \) and \( \theta \), the variance matrix \( \Sigma_k \) may be singular, and to avoid improper estimates, we will restrict the model to such cases where for all \( i = 1, \ldots, N_p \) the column space of \( \Sigma_k \), denoted by \( \text{span}(\Sigma_k) \), contains vector \( 2n_{ai} - \mu_i \). This condition also guarantees that \( \theta \in \text{span}(A) \) for all \( i \). Thus, the likelihood function is

\[
p(2n_{ai} = d | \theta, \Lambda) = \prod_{i=1}^{N_p} \left[ \frac{2n_{ai} - \mu_i}{\text{det}(\Sigma_k)} \right]^{2n_{ai}} \exp \left( -\frac{1}{2} \left( \frac{1}{\Sigma_{ai} \theta_k - \mu_i^2} \right) \right),
\]

where \( d \) is the dimension of span \( \Sigma_k \), and the determinant and the inverse of \( \Sigma_k \) are interpreted as the pseudo-determinant and the pseudo-inverse of \( \Sigma_k \) in cases where \( \Sigma_k \) is singular (Ben-Israel and Greville, 2003).

2.3 Posterior distribution

According to Bayes’ formula, the posterior distribution of the unknown variables given the observed allele frequencies is

\[
p(\nu, \Lambda, \theta, d_1, \ldots, d_N | 2n_{ai}) \propto p(\Lambda, \theta) \prod_{i=1}^{N_p} p(2n_{ai} | \theta, \Lambda, d). \tag{4}
\]

We will study this distribution by using a reversible-jump Markov chain Monte Carlo (MCMC) sampling algorithm (Green, 1995).

3 MCMC ALGORITHM

Markov chain Monte Carlo (MCMC) methods are used for generating sequences of dependent samples that are approximately distributed according to the considered target distribution (see e.g. Gamerman and Lopes, 2006). A widely used and general class of MCMC methods consists of the Metropolis–Hasting algorithms (MH; Hastings, 1970; Metropolis et al., 1953).

If we assumed that the prior knowledge of the haplotypes was complete, then an appropriate MCMC algorithm would only need to modify frequencies \( \theta \) (Section 3.2). However, in our problem we allow the number \( \nu \) and structures \( \Lambda \) of the haplotypes to vary, which makes the model more complicated to analyze. For these kinds of model choice problems, an extension of the MH algorithm called RJMCMC has been introduced by Green (1995). The application of an RJMCMC algorithm requires that the posterior distribution is known up to a single normalizing constant that must be same for each submodel. In our model, this is true since the likelihood function is known exactly, and the joint prior distribution is known up to a single normalizing constant.

3.1 Initial state

In the beginning of the MCMC run, matrix \( A \) contains all \( m \) known haplotypes and, if necessary, it is augmented by additional haplotypes in such a way that \( d \in \text{span}(A) \) for all \( i = 1, \ldots, N_p \). The frequencies are initialized as \( \theta_k = 1/\nu \) for all haplotypes \( h = 1, \ldots, v \).

3.2 Updating \( \theta \)

To generate a proposal \( \hat{\theta} \), we sample two haplotypes \( h \) and \( k \) from current frequencies \( \theta \) and redistribute their joint probability mass. More specifically, we sample \( s \sim \text{Beta}(\nu h, \nu k), \) where \( c, d > 0 \) are tuning parameters of the algorithm, and set \( \theta_k = \nu h / \nu k \) and \( \theta_k = (1 - s) \theta_k + s \theta_k \). For all \( i \neq k, h \), we set \( \theta_i = \theta_i \). This proposal is accepted or rejected according to the MH rule.

3.3 Updating \( \nu \) and \( \Lambda \)

This reversible-jump update allows modifications to the number of haplotypes. With a prespecified probability \( p_{\text{add}} \geq 0 \), we propose to add a new haplotype to \( A^* \), and with probability \( 1 - p_{\text{add}} \) we propose to delete an existing haplotype from \( A^* \). (Here, \( A^* \) denotes those \( v-m \) columns of matrix \( A \) that contain the additional haplotypes that are not included in the prior list of the known haplotypes.) The mechanism that modifies the set of haplotypes borrows ideas from the process of genetic mutation.

3.3.1 Adding a haplotype

A putative additional haplotype \( h \) is generated by modifying a haplotype \( h \) at locus \( l \), where \( h \) and \( l \) are sampled from the uniform distribution over columns of \( A \) and the number of loci, respectively. If \( h \) is already among the haplotypes of \( A \), then this update takes no action on that iteration of the algorithm. Otherwise, the relative population frequency for the additional haplotype \( h \) is proposed to be \( \hat{\theta}_h = (1 - \alpha) \theta_h \), where \( \alpha = \text{Beta}(c, d) \) and \( c, d \geq 0 \) are tuning parameters of the algorithm. Correspondingly, \( \theta_h = \nu \theta_h \).

3.3.2 Deleting a haplotype

A haplotype \( h \) that is proposed to be deleted from \( A \) is sampled from a uniform distribution over the possible values \( m < h \leq v \). If the chosen \( h \) can be modified to match some other haplotype(s) in \( A \) by changing an allelic state at a single locus, then such a locus \( s \) is sampled according to the corresponding population frequencies of the haplotypes. Let \( h^* \) be the haplotype that results when the allele at locus \( l \) of \( h \) is modified. The proposal state is such that \( h = h^* \). If the chosen \( h \) has been deleted, and \( \theta_h = \nu h \), in which, however, no column of \( A \) is within the distance of a change at a single locus of \( h \), then this update step takes no action during that iteration of the algorithm.

3.3.3 MH ratio

For each possible addition step of the update, there exist one and only one deletion step that neutralizes the addition, and vice versa. This makes it easy to calculate the reversible-jump MH ratio of this update step. Note that the choice of locus \( l \) that is to be modified in the process, is different when attempting to add a haplotype from that of attempting a deletion.

3.4 Updating \( A \) (for fixed \( \nu \))

The previously described update Step 3.3 results in situations where each haplotype in \( A^* \) tends to have a closely related haplotype in \( A \) which differs only at a single locus. To facilitate the generation of more diverse haplotypes, we also include an update step where only haplotype matrix \( A \) is modified.
We choose a haplotype \( h \) and a locus \( l \) from uniform distributions over \( \Lambda^* \) and the number of loci, respectively. Then we propose to replace haplotype \( h \) with haplotype \( \tilde{h} \) at locus \( l \). The population frequency that was previously assigned to \( h \) is transferred to \( \tilde{h} \). If the proposed haplotype \( \tilde{h} \) already exists in \( \Lambda \), this update step takes no action.

### 3.5 Other methods

The two previous approaches for haplotype frequency estimation from large DNA pools were based either on approximate maximum likelihood estimation (Kuk et al., 2009; Zhang et al., 2008) or a Bayesian model which was constrained by a system of linear equations arising from the pooled observations and known haplotype structures (Gasbarra et al., 2009). A notable difference between these approaches and the method introduced in this article is that the previous methods are not able to treat the list of haplotypes as a random variable.

#### 3.5.1 AEM with list

Kuk et al. (2009) published an AEM for finding a maximum likelihood estimate for \( \theta \) under the multinomial sampling model of the haplotypes. Their algorithm is based on the work of Zhang et al. (2008) who introduced the multinomial approximation (formula 1) for pooled allele counts as an essential part of the implementation of an EM algorithm. However, these algorithms considered only the case where \( v = 2^d \) and \( A \) contained all the possible haplotypes on \( L \) diallelic loci. In this article, such an extension of the AEM algorithm was implemented that restricts the considerations only to the given haplotype matrix \( A \). The method is called AEML (Approximate EM algorithm with List) and is publicly available from the webpage www.iki.fi/~mpninen.

#### 3.5.2 Linear equations approach

Recently Gasbarra et al. (2009) proposed a Bayesian model for estimating population haplotype frequencies \( \theta \) from pooled DNA data by using a known list \( \Lambda \). Their model builds on the exact solutions \( P_k \) of the linear equations \( \Lambda^T = \theta \) that were mentioned in Section 2.2. By finding the extremal points of the linear solution spaces of the equations, all solutions can be characterized as convex combinations of the extremal points. The MCMC algorithm of Gasbarra et al. explores that solution space under a model that combines a continuous approximation of the multinomial distribution with a sparsity-producing prior on the haplotype frequencies.

### 4 RESULTS

The above described RJMCMC algorithm was implemented in a publicly available software Hippo. The method was tested using real human data from the HapMap database (International HapMap Consortium, 2007), and comparisons were carried out with AEML as well as with an earlier Bayesian method of Gasbarra et al. (2009).

In the examples, the distance between two haplotype distributions \( \theta = (\theta_1, \ldots, \theta_2^2) \) and \( \pi = (\pi_1, \ldots, \pi_2^2) \) was measured by the total variation distance (TVD):

\[
\Delta(\theta, \pi) = \sum_{k=1}^{2^n} |\theta_k - \pi_k|.
\]

\( \Delta(\theta, \pi) \) tells how much of the probability mass should be relocated in order to turn \( \theta \) into \( \pi \) and thus varies between 0 (for identical distributions) and 1 (for completely non-overlapping distributions).

#### 4.1 Example 1: Tag-SNPs

The purpose of this example is to determine whether the proposed Bayesian model is able to identify such haplotypes that are not known a priori, but which have a considerable frequency (at least ~10%) among the sampled individuals. Such settings were created by choosing one 155 kb region from the human genome (chr7: 1678 kb . . . 1833 kb) which, according to Phase II HapMap data release 21 (International HapMap Consortium, 2007), is located between two adjacent recombination hotspots. For that region, there are 148 SNPs genotyped in the HapMap database (release 21) for the CEU population (Utah residents with ancestry from northern and western Europe). In this example, the Tagger algorithm of de Bakker et al. (2005) was used for picking five tag-SNPs for the considered region in the CEU population, and the resulting tag-SNPs cover physically a 78 kb region (chr7: 1685 kb . . . 1763 kb), which originally contained 90 SNPs in the HapMap database. To create a dataset, 200 haplotypes were chosen with replacement from among the 120 CEU haplotypes available in the HapMap database. Nine different haplotypes are present on the five tag-SNPs in the final sample (Table 1).

One hundred datasets were created by randomly dividing the sampled haplotypes into 10 pools each containing 20 haplotypes (corresponding to 10 diploid individuals). Then each of these datasets was analysed by using Hippo and AEML with six different prior lists of known haplotypes. In case A, the prior list contained only the five most frequent haplotypes in the population. In cases B–E, the prior list contained eight haplotypes; the haplotype that was left out from the list in each case can be seen in Figure 1. In case F, the list contained all nine haplotypes that were actually present in the datasets. Finally, in case G, the list contained all 32 possible haplotypes on five loci. Because under incomplete prior information the posterior distribution of Hippo may be multimodal, each of the 100 datasets was actually analysed five times in each of the six cases, and from those five replicates the one whose posterior expectation had the highest posterior density according to the model was reported. (For AEML, all five separate runs produced similar results.) Some statistics of the TVDs between the estimated haplotype frequencies and the true ones are given in Table 2.

From Table 2, it can be seen that Hippo and AEML produced similar accuracy in case A, where the prior list of haplotypes contained the five most common haplotypes of the population corresponding together to 97.5% of all haplotypes, as well as in case F, where the prior knowledge of the haplotypes was complete. In case G, where no prior list was used, AEML yielded slightly more accurate estimates than Hippo. The method A+H in case G was a combination of AEML and Hippo, where AEML was first run for the datasets by including all 32 haplotypes in the list and then those haplotypes whose AEML estimates were over 5% were chosen to form the initial list for Hippo. This improved the results compared with the standard version of Hippo, but the results were not consistently better than those of the standard AEML runs.

The advantage gained from the varying haplotype list of Hippo becomes clear in the cases B–E. There AEML with a constant list is deemed to have a TVD of at least of the frequency of the missing haplotype, but Hippo achieves almost consistently as good accuracy as with the complete prior information (case F). The accuracy

| Table 1. Haplotypes and their frequencies in the dataset of Example 1 |
|------------------|------------------|
|                   | 0.0011 | 11100  | 00110  | 01110  | 00010  | 11110  | 10110  | 00001  | 11111  |
|                   | 0.4    | 0.31   | 0.125  | 0.09   | 0.04   | 0.015  | 0.01   | 0.005  | 0.005  |

3299
of Hippo in cases A–E is better than that of AEML in case G, thus, showing that if most frequent haplotypes are available in the database then the use of them as prior information is advantageous compared with the strategy which includes all possible haplotypes in the haplotype list. The method of Kuk et al. (2009) corresponds to AEML in case G.

### 4.2 Example 2: fine mapping

In this example, a 36.8 kb region (chr2: 136,642 … 136,678 kb) from the HapMap database (phase 3, release 2) is considered. For CEU (Utah residents with ancestry from northern and western Europe) and CHB (Han Chinese in Beijing) populations, HapMap contains 234 and 168 haplotypes, respectively. Fifteen SNPs are polymorphic in the considered region when the data of these two populations are combined. In case A of this example, 1000 haplotypes from CHB population were sampled (with replacement) (Table 3). One hundred pooled datasets were created from the sampled haplotypes by randomly combining the haplotypes into 20 pools each containing 50 haplotypes (corresponding to 25 diploid individuals). Then the pooled data were analysed by Hippo and AEML by using the four most common CHB haplotypes as the prior list of haplotypes (the first four rows of Table 3). Each of the 100 datasets were analysed 10 times by both methods and the run whose estimates had the highest posterior probability (Hippo) or likelihood (AEML) was chosen for records. The quantiles of TVD to the true distribution given in Table 4 show that Hippo was able to find also a priori unknown rare haplotypes since in 95% of the cases TVD was less than the sum of the frequencies of the unknown haplotypes (11.5%). AEML performed as well as possible with the constant list as TVD to true distribution was consistently equal to the proportion of the unknown haplotypes. This is not surprising since each of the four listed haplotypes has a certain SNP that characterizes it among the listed haplotypes and therefore the listed haplotypes cannot be mixed up with each other. The interesting observation is that Hippo is able to do better than the deterministic inference of AEML in almost all cases.

There is one particular haplotype (the bottom row of Table 3) that is present in CEU population with frequency of 20%, but which is not
The parameter values for Hippo are given in Table 5. The burn-in period was one-tenth of the whole number of iterations in both the examples. AEML was run until an increase in the likelihood was <10<sup>-7</sup> or 10,000 iterations were executed. The method of Gasbarra et al. (2009) was run with the same parameters that were used in their paper. The elapsed times of single runs with Hippo were about 25 s and 5 min in the datasets of Examples 1 and 2, respectively (Pentium-4 2.80 GHz, 2 GB RAM). AEML performed each run in a couple of seconds. Technically, for both Hippo and AEML, the asymptotic running time per iteration is linear with respect to the number of pools and cubic with respect to the number of loci and does not depend on the pool sizes. In practice, however, the running times of the programs grow also with pool size because with larger pools more iterations are required before convergence is reached.

### Table 4. Quantiles of distances to the true distribution in Example 2

<table>
<thead>
<tr>
<th>Case</th>
<th>Method</th>
<th>5%</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Hippo</td>
<td>0.016</td>
<td>0.033</td>
<td>0.048</td>
<td>0.056</td>
<td>0.106</td>
</tr>
<tr>
<td></td>
<td>AEML</td>
<td>0.115</td>
<td>0.115</td>
<td>0.115</td>
<td>0.115</td>
<td>0.115</td>
</tr>
<tr>
<td>B</td>
<td>Hippo</td>
<td>0.015</td>
<td>0.032</td>
<td>0.047</td>
<td>0.055</td>
<td>0.092</td>
</tr>
<tr>
<td></td>
<td>AEML</td>
<td>0.294</td>
<td>0.294</td>
<td>0.294</td>
<td>0.294</td>
<td>0.294</td>
</tr>
</tbody>
</table>

### Table 5. Hippo’s parameters in Examples 1 and 2

<table>
<thead>
<tr>
<th>Ex.</th>
<th>u</th>
<th>y</th>
<th>c</th>
<th>d</th>
<th>Pstir</th>
<th>cinit</th>
<th>cnew</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10&lt;sup&gt;-6&lt;/sup&gt;</td>
<td>8</td>
<td>10.0</td>
<td>0.2</td>
<td>0.5</td>
<td>10.0</td>
<td>2.0</td>
<td>2.5 x 10&lt;sup&gt;5&lt;/sup&gt;</td>
</tr>
<tr>
<td>2</td>
<td>10&lt;sup&gt;-5&lt;/sup&gt;</td>
<td>12</td>
<td>10.0</td>
<td>0.2</td>
<td>0.5</td>
<td>10.0</td>
<td>2.0</td>
<td>5.0 x 10&lt;sup&gt;5&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

observed at all among CHB samples. In case B that haplotype was added among the CHB haplotypes in such a way that its frequency was 20%, and 1000 haplotypes were sampled with replacement resulting in the frequencies that are shown in Table 3. Again 100 datasets of 20 pools each containing 50 haplotypes were created and analysed by Hippo and AEML using the same list of the four most common CHB haplotypes as the prior information about the existing haplotypes. AEML produced again the results whose TVD to the true distribution was equal to the sum of the frequencies of the missing haplotypes (29.4%). Hippo was able to achieve the same accuracy as in case A, thus, showing that the missing haplotypes were well estimated. Figure 2 shows the estimates of the CEU haplotype given by Hippo over all 100 datasets showing that in almost all cases the estimates were between 18% and 22% when the true value is 20.2%.

### 4.3 Parameter values

The parameter values for Hippo are given in Table 5. The burn-in period was one-tenth of the whole number of iterations in both the examples. AEML was run until an increase in the likelihood of the a priori missing haplotype in Example 2. True value is marked with X.

This article introduces a Bayesian method for estimating haplotype frequencies from large DNA pools. The method combines multinormal approximation of the pooled allele counts (Kuk et al., 2009; Zhang et al., 2008) with a possibility to use external database information as prior knowledge about the possible major haplotypes in the population (Gasbarra et al., 2009). A completely new aspect of the introduced method is its ability to perform well also in situations where the prior knowledge about the haplotypes is incomplete.

### 5 DISCUSSION

This article introduces a Bayesian method for estimating haplotype frequencies from large DNA pools. The method combines multinormal approximation of the pooled allele counts (Kuk et al., 2009; Zhang et al., 2008) with a possibility to use external database information as prior knowledge about the possible major haplotypes in the population (Gasbarra et al., 2009). A completely new aspect of the introduced method is its ability to perform well also in situations where the prior knowledge about the haplotypes is incomplete.

Applicability: the motivation for pooling genetic material comes from the savings that result from the decrease in the total number of genotypings. For single locus analyses, the benefits of pooling techniques are clear, but for more complex analyses the partial loss of the haplotype information during the pooling process poses a problem. However, the results in this article show that under reasonably accurate knowledge of the structures of the major haplotypes in the population also data from large pools can be haplotyped efficiently. For example, in Example 2 the proposed method (Hippo) estimated the haplotype distribution on average within a distance of 0.05 from the true distribution by using only 20 large pools (25 individuals in each). If, instead, the same number of genotypings were carried out at individual level (20 individuals, 40 haplotypes), the corresponding median distance to the true distribution would be as high as 0.28 due to the small sample size. (This figure was calculated by assuming that the haplotypes could be inferred perfectly from the individually genotyped data.) Thus, DNA pooling provides realistic possibilities for reducing the cost of genetic studies also in cases that require haplotype estimation if appropriate statistical methods and an external source of population haplotype data are available.

When haplotypes are estimated from large DNA pools, the considerations need to be restricted to relatively short genomic regions that rarely encounter recombination events. This is because longer regions contain more variation and a considerable proportion of rare haplotypes whose frequencies cannot be accurately estimated from pooled data. Thus, a possible application of the method would be in estimating the haplotype distributions within certain genes or other relatively short regions in one or more populations. Example 2 of this article presented such a situation where two closely related populations (cases A and B) were analysed at a 36.8 kb region and a significant difference in frequency of one particular haplotype was (correctly) identified.

Model: this paper provides a basic formulation of a Bayesian model that is able to treat the haplotype list (A) as a random variable. Several straightforward extensions of the model would be readily available.
In this article, the prior distribution for haplotype frequencies \( \theta \) was a symmetric Dirichlet distribution, which seems to be an appropriate choice when the frequencies among the sampled haplotypes may differ considerably from those in the database. However, if more detailed prior knowledge were available, it would also be possible to use an asymmetric Dirichlet distribution whose parameters would reflect the available prior information.

In this article, the additional haplotypes (\( A^+ \)) were a priori assumed to be uniformly distributed given the number of haplotypes (\( \nu \)). Such a prior implies that the existing haplotypes do not tell anything about the unobserved ones. This is of course only a coarse simplification of the truth since in reality the haplotypes in the population have a common history and they have developed from common ancestors through recombinations and mutations. There exist approximations of the conditional sampling distribution for a haplotype from a population given a group of observed haplotypes (Li and Stephens, 2003; Stephens and Scheet, 2005). By using such a model, it could be possible to define a joint prior for \( \theta \) and \( A \) that would take into account the similarities in structures of the different haplotypes. On the other hand, if the list of known haplotypes is (at least almost) comprehensive, then the uniform prior for \( A^+ \) may already be sufficient for all practical purposes.

Still another way to extend the model is by treating \( \alpha \) and \( \gamma \) as random variables with their own prior distributions. For \( \alpha \) such an approach was taken by Gasbarra et al. (2009).

In the examples of this article, there were considerable variation in the accuracy of the estimates with different pooled configurations. Partially, this may be a consequence of a relatively small number of pools, but a more extensive study on the form of the prior distributions and the way of choosing the best candidate from several modes of the posterior distribution could also bring more robustness to the proposed method.

Algorithm The current RJMCMC algorithm proposes new haplotypes by modifying a single site of an existing haplotype. This procedure mimics the natural process of mutation and thus produces putative haplotypes that are closely related to the existing ones. Another natural candidate for creating putative new haplotypes would mimic the recombination process by sampling existing ones. Another natural candidate for creating putative new haplotypes would be modifying a single site of an existing haplotype. Thus produces putative haplotypes that are closely related to the haplotypes by modifying a single site of an existing haplotype.

ACKNOWLEDGEMENTS

I would like to thank Dr Mikko J. Sillanpää, Dr Sangitula Kalathinal and Dr Dario Gasbarra for discussions related to this work. I am also thankful to three anonymous reviewers whose comments helped me to improve the manuscript.

Funding: The ComBi graduate school of the Academy of Finland.

Conflict of Interest: none declared.

REFERENCES


