Maximum Likelihood Resolution of Multi-block Genotypes

Gad Kimmel
School of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel
kgad@tau.ac.il

Ron Shamir
School of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel
rshamir@tau.ac.il

ABSTRACT

We present a new algorithm for the problems of genotype phasing and block partitioning. Our algorithm is based on a new stochastic model, and on the novel concept of probabilistic common haplotypes. We formulate the goals of genotype resolving and block partitioning as a maximum likelihood problem, and solve it by an EM algorithm. When applied to real biological SNP data, our algorithm outperforms two state of the art phasing algorithms. Our algorithm is also considerably more sensitive and accurate than a previous method in predicting and identifying disease association.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical Sciences—*Biology and genetics*; G.3 [**Probability and Statistics**]: Probabilistic algorithms

General Terms

algorithms, haplotyping

Keywords

haplotype, haplotype block, genotype, SNP, algorithm, maximum likelihood, genotype phasing, haplotype resolution, disease association

1. INTRODUCTION

A major challenge after the completion of the human genome project is to learn about DNA differences among individuals. This knowledge can lead to better understanding of human genetics, and to finding the genetic causes for complex and multi-factorial diseases. Most DNA differences among individuals are single base sites, in which more than one nucleic acid can be observed across the population. Such differences and their sites are called *single nucleotide polymorphisms* (SNPs) [19, 7]. Usually only two alternative bases occur at a SNP site. Millions of SNPs have already been detected [20, 22], out of an estimated total of 10 millions common SNPs [13].

When studying polymorphism in the population, one looks only at SNP sites and disregards the long stretches of bases between them that are the same in the population. The sequence variants at each site are called the *alleles* at that site. The sequence of alleles in contiguous SNP sites along a chromosomal region is called a *haplotype*. Recent evidence indicates that haplotypes tend to be preserved along relatively long genomic stretches, with recombination occurring primarily in narrow regions called *hot spots* [7, 19]. The regions between two neighboring hot spots are called *blocks*. The number of distinct haplotypes within each block that are observed in a population is very limited: typically, some 70-90% of the haplotypes within a block are identical (or almost identical) to very few (2-5) distinct *common haplotypes* [19]. This finding is very important for disease association studies, since once the blocks and the common haplotypes are identified, one can, in principle, obtain a much stronger association between a haplotype and a disease phenotype.

Several studies have concentrated on the problem of block identification in a given collection of haplotypes: Zhang et al. [27, 28] sought a block partitioning that minimizes the number of tag SNPs (roughly speaking, this is a set of sites with the property that the combination of alleles in it uniquely identifies the alleles at all other sites, or a prescribed fraction of the haplotypes in that block). Koivisto et al. [12] used a minimum description length (MDL) criterion for block definition. Kimmel et al. [11] minimized the total number of common haplotypes, while allowing errors and missing data. All these studies used the same basic dynamic programming approach of [27] to the problem, but differed in the optimization criterion used within the dynamic programming computation.

The block partitioning problem is intertwined with another problem in diploid organisms. Such organisms (including humans) have two near-identical copies of each chromosome. Most techniques for determining SNPs do not provide the haplotype information separately for each of the two copies. Instead, they generate for each site *genotype* information, i.e., an unordered pair of allele readings, one from each copy [20].

Hence, given the genotype data {A,A} {A,C} {C,G} for three SNP sites in a certain individual, there are two possible haplotype pair solutions: (ACC and AAG), or (ACG and AAC). A genotype with two identical bases in a site is

RECOMB'04, March 27–31, 2004, San Diego, California, USA.

called *homozygote*, while a genotype with two different bases is called *heterozygote* in that site. The genotype in the example above is homozygote for the allele A in the first site, and heterozygote in the second and third sites. The process of inferring the haplotypes from the genotypes is called *phasing* or *resolving*.

In the absence of additional information, each genotype with h heterozygote sites can be resolved in 2^{h-1} different ways. Resolving is done simultaneously in all the available genotypes and is based on some assumptions on how the haplotypes were generated. The first approach to haplotype resolution was Clark's parsimony-based algorithm [3]. Likelihood-based EM algorithms [6, 15] gave better results. Stephens et al. [21] and Niu et al. [18] proposed MCMC-based methods which gave promising results. All of those methods assumed that the genotype data correspond to a single block with no recombination events. Hence, for multi-block data the block structure must be determined separately.

A novel combinatorial model was suggested by Gusfield [9]. According to this model, the resolution must produce haplotypes that define a *perfect phylogeny tree*. Gusfield provided an efficient yet complex algorithm for the problem. Simpler, direct efficient algorithms under this model were recently developed [5, 1]. Eskin et al. [5] showed good performance with low error rates on real genotypes.

While elegant and powerful, the perfect phylogeny approach has certain limitations: first, it assumes that the input data admit a perfect phylogeny tree. This assumption is often violated in practice, due to data errors and rare haplotypes. In fact, Eskin at al. show that in the real data that they analyzed, a block does not necessarily admit a perfect phylogeny tree. Second, the model requires partition of data into blocks by other methods. Third, the solution to the problem may not be unique and there may be several (or many) indistinguishable solutions. (These limitations were addressed heuristically in [5]). Recently, Greenspan and Geiger [8] proposed a new method and algorithm, called *HaploBlock*, which performs resolution while taking into account the blocks structure. The method is based on a Bayesian network model. Very good results were reported.

In this study we provide a new algorithm for block partitioning and phasing. Our algorithm is based on a new model for genotype generation. Our model is based on a haplotype generation model, parts of which were suggested by Koivisto et al. [12]. In our model, common haplotypes are redefined in a probabilistic setting, and we seek a solution that has maximum likelihood, using an EM algorithm. The model allows errors and rare haplotypes, and the algorithm is particularly tailored to the practical situation in which the number of common haplotypes is very small. We applied our algorithm to two genotype data sets: on the data set of Daly et al. [4] our algorithm performed better than HaploBlock [8] and Eskin et al. [5]. On genotype and phenotype data for the μ opioid receptor gene, due to Hoehe et al. [10], our algorithm revealed strong association for disease, by using blocks partitioning and resolving, and improved sharply over the original analysis in [10].

Unlike most former probabilistic approaches [6, 15, 21, 18], our algorithm reconstructs the block partitioning and resolves the haplotypes simultaneously, and assigns a likelihood value to the complete solution. Consequently, it is considerably faster and more accurate. While our approach has some resemblance to HaploBlock, there are also significant differences. First, our approach is not based on a Bayesian network, but rather computes the maximum likelihood directly. Second, our algorithm actually computes the likelihood function of each block, and thus the real maximum likelihood partitioning is optimized, while HaploBlock uses an MDL criterion for block partitioning. Third, once the model parameters are found, we solve the phasing problem directly to optimality, such that the likelihood function is maximized. In constrast, HaploBlock applies a heuristic to find the block partitioning, even though this partitioning is part of the model parameters. Fourth, our stochastic model allows a continuous spectrum of probabilities for each component in each common haplotype, while the HaploBlock software allows only two common probability values for all mutations. HaploBlock has the theoretical advantage of allowing a larger number of common haplotypes, but this is apparently less relevant in practice [7, 4]. HaploBlock's model also incorporates inter-block transitions, while we handle them separately after the main optimization process.

This paper is organized as follows: In Section 2 we present our stochastic model, in Section 3 we show how block partitioning and resolution of haplotypes are performed under our model. Section 4 contains our results on the two real data sets.

2. THE STOCHASTIC MODEL

Consider first the problem of resolving a single block. The input to the problem is presented by a $n \times m$ *genotype matrix* M, in which the rows correspond to samples (individuals genotyped), and columns correspond to SNP sites. Hence, the i^{th} row $M[i,*]$ describes the i^{th} genotype (the vector of readings for all the SNP sites), which is also denoted by $\boldsymbol{g_i}$. We assume that all sites are bi-allelic, and that the two alleles were renamed arbitrarily to 0 and 1. The genotype readings are denoted by $M[i,j] \in \{0,1,2\}$. 0 and 1 stand for the two homozygote types {0,0} and {1,1}, respectively, and 2 stands for a heterozygote. A $2n \times m$ binary matrix M' is an *expansion* of the genotype matrix M if each row $M[i,*]$ expands to two rows denoted by $M'[i,*]$ and $M'[i',*]$, with $i' = n + i$, satisfying the following: for every i, if $M[j,i] \in \{0,1\}$, then $M[i,j] = M'[i,j] = M'[i',j]$; if $M[i,j] = 2$, then $M'[i,j] \neq M'[i',j]$. M' is also called a *haplotype matrix* corresponding to M. Given a genotype matrix, the *phasing problem* is to find its best expansion, i.e., the best n pairs of haplotype vectors that could have generated the genotype vectors. "Best" must be defined with respect to the data model used.

We now describe our stochastic model for how the haplotype matrix of a single block is generated. The model aims to reflect the fact that only few distinct common haplotypes are usually observed in each block [7, 4], and the variability between observed haplotypes originating from the same common haplotype. The model assumes a set of common haplotypes that occur in the population with certain probabilities. Each genotype is created by selecting independently two of the common haplotypes according to their probabilities and forming their confluence. The two (possibly identical) common haplotypes are called the *creators* of that genotype. The key property of our model is the probabilistic formulation of the common haplotypes: Formally, a *probabilistic common haplotype* is a vector, whose

components are the probabilities of having the allele '1' in each site of a haplotype created by it. Hence, a vector of only zeroes and ones corresponds to a standard (consensus) common haplotype, and a vector with fractional values allows for deviations from the consensus with certain (small) probabilities, independently for each site. In this way, a common haplotype may appear in different genotypes in distinct forms. A similar model was used in [12] in the context of block partitioning of haplotype (phased) data.

A precise definition of the stochastic model is as follows. Assume that the genotype matrix M contains only one block. Let k be the number of common haplotypes in that block. Let $\{\boldsymbol{\theta_i}\}_{1\le i\le k}$ be the probability vectors of the common haplotypes, where $\boldsymbol{\theta_i} = (\theta_{i,1}, \ldots, \theta_{i,m})$ and $\theta_{i,j}$ is the probability to observe '1' in the j^{th} site of the i^{th} common haplotype. (Consequently, $1-\theta_{i,j}$ is the probability to observe '0' in that site.) Let $\alpha_i > 0$ be the probability of the i^{th} common haplotype in the population, with $\sum_{i=1}^{k}\alpha_i = 1$. Each row in the matrix M is generated as follows:

- Choose a number i between 1 and k according to the probability distribution $\{\alpha_1, \ldots, \alpha_k\}$. i is the index of the first common haplotype.
- The haplotype $(x_1, \ldots, x_m)$ is generated by setting, for each site j independently, $x_j = 1$ with probability $\theta_{i,j}$.
- Repeat the steps above for the second haplotype and form their confluence. The result is the genotype in that row.

For generating a matrix with several blocks, the process is repeated for each block independently. Our main task will be to show how to infer the parameters and the haplotypes from genotype data of a single block. This inference also gives a likelihood for the block. Given a multi-block matrix, a dynamic programming algorithm is used to find the maximum likelihood block partitioning.

3. INFERRING THE MODEL PARAMETERS

For a single genotype $\boldsymbol{g_j}$, assuming its creators $\boldsymbol{\theta_a}$ and $\boldsymbol{\theta_b}$ are known, the probability of obtaining $\boldsymbol{g_j}$ is:

$$f(\boldsymbol{g_j};\boldsymbol{\theta_a},\boldsymbol{\theta_b}) = \prod_{i=1}^{m}\begin{cases}(1-\theta_{a,i})(1-\theta_{b,i}) & g_{j,i}=0\\ \theta_{a,i}\theta_{b,i} & g_{j,i}=1\\ \theta_{a,i}(1-\theta_{b,i})+\theta_{b,i}(1-\theta_{a,i}) & g_{j,i}=2\end{cases}.$$

We denote by I_i and J_i the index of the first and second creator of genotype g_i, respectively. The complete likelihood of all genotypes is:

$$L(M) = \prod_{i=1}^{n}\alpha_{I_i}\alpha_{J_i}f(\boldsymbol{g_i};\boldsymbol{\theta_{I_i}},\boldsymbol{\theta_{J_i}}).$$

Let the random variable $A_j^{(i)}$ be the number of times the vector $\boldsymbol{\theta_j}$ appears as a creator of genotype $\boldsymbol{g_i}$. Clearly, $A_j^{(i)}$ can be 0, 1, or 2. The log likelihood can be written as:

$$\begin{aligned}l(M) &= \sum_{i=1}^{n}[\log\alpha_{I_i}+\log\alpha_{J_i}+\log f(\boldsymbol{g_i};\boldsymbol{\theta_{I_i}},\boldsymbol{\theta_{J_i}})]\\ &= \sum_{i=1}^{n}[\sum_{a=1}^{k}A_a^{(i)}\log\alpha_a + \sum_{1\le a<b\le k}A_a^{(i)}A_b^{(i)}\log f(\boldsymbol{g_i};\boldsymbol{\theta_a},\boldsymbol{\theta_b})\\ &\quad + \sum_{a:\ A_a^{(i)}=2}\log f(\boldsymbol{g_i};\boldsymbol{\theta_a},\boldsymbol{\theta_a})].\end{aligned}$$

Let $I_{\{A_a^{(i)}=2\}}$ be an indicator random variable for the event $A_a^{(i)} = 2$. Then we can replace the last sum in $l(M)$ by $\sum_{a=1}^{k} I_{\{A_a^{(i)}=2\}}\log f(\boldsymbol{g_i};\boldsymbol{\theta_a},\boldsymbol{\theta_a})$. Since I_i and J_i, for $1\le i\le n$, are unknown, we use the EM approach (see, e.g., [17]). We denote the set of parameters by $\vartheta \equiv \{\alpha_i, \boldsymbol{\theta_i}\colon 1\le i\le k\}$. Given an initial set of parameters ϑ_0, we want to find another set of parameters ϑ of higher likelihood. This can be done by maximizing the conditional expectation:

$$\begin{aligned}Q_{M,\vartheta_0}(\vartheta) = \mathbb{E}_{\vartheta_0}[l|M] &= \sum_{i=1}^{n}[\sum_{a=1}^{k}\mathbb{E}_{\vartheta_0}[A_a^{(i)}|\boldsymbol{g_i}]\log\alpha_a\\ &\quad + \sum_{1\le a<b\le k}\mathbb{E}_{\vartheta_0}[A_a^{(i)}A_b^{(i)}|\boldsymbol{g_i}]\log f(\boldsymbol{g_i};\boldsymbol{\theta_a},\boldsymbol{\theta_b})\\ &\quad + \sum_{a=1}^{k}\mathbb{E}_{\vartheta_0}[I_{\{A_a^{(i)}=2\}}|\boldsymbol{g_i}]\log f(\boldsymbol{g_i};\boldsymbol{\theta_a},\boldsymbol{\theta_a})].\end{aligned}$$

In order to find $\arg\max_\vartheta Q_{M,\vartheta_0}(\vartheta)$, we need that $\forall i,j:\ 1\le i\le k;\ 1\le j\le m$, $\frac{\partial Q}{\partial\alpha_i}=0$ and $\frac{\partial Q}{\partial\theta_{i,j}}=0$.

Expectation:
The first step is to find $\boldsymbol{\alpha}$, such that Q is maximized. The conditional probabilities are:

$$\begin{aligned}P_{\vartheta_0}[A_j^{(i)}=1|\boldsymbol{g_i}] &= \frac{\sum_{1\le x\le k,\ x\ne j}2\alpha_x\alpha_j f(\boldsymbol{g_i};\boldsymbol{\theta_x},\boldsymbol{\theta_j})}{\sum_{x=1}^{k}\sum_{y=1}^{k}\alpha_x\alpha_y f(\boldsymbol{g_i};\boldsymbol{\theta_x},\boldsymbol{\theta_y})},\\ P_{\vartheta_0}[A_j^{(i)}=2|\boldsymbol{g_i}] &= \frac{\alpha_j\alpha_j f(\boldsymbol{g_i};\boldsymbol{\theta_j},\boldsymbol{\theta_j})}{\sum_{x=1}^{k}\sum_{y=1}^{k}\alpha_x\alpha_y f(\boldsymbol{g_i};\boldsymbol{\theta_x},\boldsymbol{\theta_y})}.\end{aligned} \quad (1)$$

We use Equations (1) to calculate the conditional expectation:

$$\mathbb{E}_{\vartheta_0}[A_j^{(i)}|\boldsymbol{g_i}] = P_{\vartheta_0}[A_j^{(i)}=1|\boldsymbol{g_i}] + 2P_{\vartheta_0}[A_j^{(i)}=2|\boldsymbol{g_i}].$$

The requested α_j can then be written as follows:

$$\alpha_j = \frac{1}{2n}\sum_{i=1}^{n}\mathbb{E}_{\vartheta_0}[A_j^{(i)}|\boldsymbol{g_i}].$$

In order to calculate the vectors $\boldsymbol{\theta_i}$ for $1\le i\le k$, we first need to get the conditional expectations:

$$\begin{aligned}\mathbb{E}_{\vartheta_0}[A_a^{(i)}A_b^{(i)}|\boldsymbol{g_i}] &= P_{\vartheta_0}[A_a^{(i)}=1, A_b^{(i)}=1|\boldsymbol{g_i}]\\ &= \frac{2\alpha_a\alpha_b f(\boldsymbol{g_i};\boldsymbol{\theta_a},\boldsymbol{\theta_b})}{\sum_{x=1}^{k}\sum_{y=1}^{k}\alpha_x\alpha_y f(\boldsymbol{g_i};\boldsymbol{\theta_x},\boldsymbol{\theta_y})},\\ \mathbb{E}_{\vartheta_0}[I_{\{A_a^{(i)}=2\}}|\boldsymbol{g_i}] &= P_{\vartheta_0}[A_a^{(i)}=2|\boldsymbol{g_i}].\end{aligned} \quad (2)$$

Maximization:
Now $\frac{\partial Q}{\partial\theta_{i,j}}$ can be calculated, using Equations (2):

$$\begin{aligned}&\frac{\partial Q}{\partial\theta_{i,j}} =\\ &\sum_{s=1}^{n}\Bigg[\sum_{1\le a\le k,\ a\ne i}\mathbb{E}_{\vartheta_0}[A_a^{(s)}A_i^{(s)}|\boldsymbol{g_s}]\cdot\begin{cases}\frac{1}{\theta_{i,j}-1} & g_{s,j}=0\\ \frac{1}{\theta_{i,j}} & g_{s,j}=1\\ \frac{1-2\theta_{a,j}}{\theta_{a,j}+\theta_{i,j}-2\theta_{a,j}\theta_{i,j}} & g_{s,j}=2\end{cases} +\\ &+\mathbb{E}_{\vartheta_0}[I_{\{A_a^{(s)}=2\}}|\boldsymbol{g_s}]\cdot\begin{cases}\frac{2}{\theta_{i,j}-1} & g_{s,j}=0\\ \frac{2}{\theta_{i,j}} & g_{s,j}=1\\ \frac{1-2\theta_{i,j}}{\theta_{i,j}-\theta_{i,j}^2} & g_{s,j}=2\end{cases}\Bigg].\end{aligned}$$

An inspection of the system of equations $\frac{\partial Q}{\partial\theta_{i,j}}=0$ for all $\theta_{i,j}$ reveals that for each j, the set of equations for $\{\theta_{i,j}: 1\le$

$i \leq k\}$ can be solved separately. In other words, for each j we have k polynomials with k variables: $\{\theta_{i,j} | 1 \leq i \leq k\}$. These equations can be solved numerically in practice, since k is assumed to be small.

Using this approach, we iteratively recalculate the parameters of the model, until convergence of the likelihood to a local maximum. Once the parameters are found, resolving is performed as follows: for each genotype $\boldsymbol{g_i}$, we find $P_\vartheta[A_a^{(i)} = 1, A_b^{(i)} = 1|\boldsymbol{g_i}]$ and $P_\vartheta[A_a^{(i)} = 2|\boldsymbol{g_i}]$, for each a and b. The indices of the creators of $\boldsymbol{g_i}$ are then determined by $\arg\max\{\max_{a \neq b} P_\vartheta[A_a^{(i)} = 1, A_b^{(i)} = 1|\boldsymbol{g_i}], \max_a P_\vartheta[A_a^{(i)} = 2|\boldsymbol{g_i}]\}$. Once the creators $\boldsymbol{\theta_a}$ and $\boldsymbol{\theta_b}$ of genotype $\boldsymbol{g_i}$ are known, its alleles at each heterozygote read j are $h_{i,j}^a = 1$, $h_{i,j}^b = 0$ if $\theta_{a,j} > \theta_{b,j}$, and $h_{i,j}^a = 0$, $h_{i,j}^b = 1$, otherwise.

Each of the common haplotypes is represented by a vector of probabilities $\boldsymbol{\theta_i}$. The corresponding binary common haplotype vector $\widehat{\boldsymbol{\theta}}_{\boldsymbol{i}}$ is obtained by rounding: $\widehat{\theta}_{i,j} = 0$ if $\theta_{i,j} \leq 0.5$ and $\widehat{\theta}_{i,j} = 1$ otherwise.

3.1 Finding the Number of Common Haplotypes in Each Block

The calculations of the maximum likelihood solution assume that k is known. In real biological data, we know that k is small, but its value is unknown. To overcome this obstacle, we calculate the likelihood $L(M, k)$ of a block with k common haplotypes, for $k = 1, \ldots, u$, where u is a small number (usually 5). It is easy to see that $L(M, i)$ is monotone non-decreasing in i. Let $\Delta(M, k) := L(M, k + 1) - L(M, k)$. In practice, when k exceeds the correct number of common haplotypes, $\Delta(M, k)$ becomes small. Thus, we choose the first k such that $\Delta(M, k) \leq \epsilon$, where ϵ is a parameter of the algorithm.

3.2 Finding the Blocks

To find the optimal block partition, we seek one that maximizes the overall likelihood of the data. The procedure is straightforward dynamic programming as in [27]. We first calculate for each j and for each $i > j$ the value l_{ji}, the log likelihood of the best solution forming a single block spanning columns i through j, as described above. Let T_i be the maximum log likelihood of a multi-block solution on the submatrix of M induced on the columns $1, \ldots, i$, where $T_0 = 0$. Then the following recursive formula is used to compute T_i:

$$T_i = \max_{0 \leq j \leq i-1} \{T_j + l_{ji}\} .$$

T_m gives the total log likelihood of the complete multi-block solution.

3.3 Matching Pairs of Blocks

So far, we have shown how to find the haplotypes of each individual within each block. This determines which alleles within the block appear together on the same chromosome. Our next challenge is to perform a similar task on the inter-block level, i.e., to determine for each individual which of the two haplotypes in each block occur on the same chromosome, and in this way to determine its complete chromosome pair. We call this problem *matching pairs of blocks*. If there are b blocks and the two haplotypes within each of them are distinct, then there are 2^{b-1} possible matchings. We seek a simultaneous solution for all individuals which will be "best" in a precise sense. This problem was presented in [5], where a combinatorial algorithm was proposed for solving it.

Our solution for the problem will be based on the observation that common haplotypes tend to pair unevenly across block boundaries [4]. Specifically, a common haplotype in one block may tend to appear on the same chromosome with another common haplotype in the next one, forming stretches that join together common haplotypes in several blocks.

The problem is solved in the following way: Let t and $t + 1$ be the indices of two consecutive blocks. Let $\{a_i^t, b_i^t\}$ and $\{a_i^{t+1}, b_i^{t+1}\}$ be the common haplotypes in blocks t and $t + 1$ respectively, for genotype $\boldsymbol{g_i}$. These can be matched as $\{(a_i^t, a_i^{t+1}), (b_i^t, b_i^{t+1})\}$ or $\{(a_i^t, b_i^{t+1}), (b_i^t, a_i^{t+1})\}$. In block t there are up to k different common haplotypes, denoted by $\{a_s^t\}_{1 \leq s \leq k}$. Recall, that the probability that the common haplotype a appears in a genotype is α_a. Let $P_{trans}(a, b)$ be the transition probability from common haplotype a in block t to common haplotype b in block $t+1$, i.e., the probability that if common haplotype a appears in block t, then common haplotype b appears in block $t + 1$ on the same chromosome. Denote by A_i^t an indicator random variable that has value 1 iff the matching for the i^{th} genotype is $\{(a_i^t, a_i^{t+1}), (b_i^t, b_i^{t+1})\}$, and let $\bar{A}_i^t = 1 - A_i^t$. Over all, with respect to blocks t and $t + 1$, the log likelihood function is:

$$l = \sum_{i=1}^{n} [\ln \alpha_{a_i^t} + \ln \alpha_{b_i^t} + A_i^t \ln(P_{trans}(a_i^t, a_i^{t+1}) P_{trans}(b_i^t, b_i^{t+1})) + \bar{A}_i^t \ln(P_{trans}(a_i^t, b_i^{t+1}) P_{trans}(b_i^t, a_i^{t+1}))].$$

Here too we find the parameters $\{P_{trans}(a_i, b_j)\}$ using maximum likelihood estimation by an EM approach: The transition probabilities can be obtained from $\{\mathbb{E}[A_i^t]\}_{1 \leq i \leq n}$ and vise versa by closed formulas, so calculating $\{\mathbb{E}[A_i^t]\}_{1 \leq i \leq n}$ and the transition probabilities can be performed iteratively, until convergence of the likelihood. Once the transition probabilities are known, the decision which of the two possible pairs matching to choose can be done for each $1 \leq i \leq n$, according to:

$$\arg\max\{P_{trans}(a_i^t, a_i^{t+1}) P_{trans}(b_i^t, b_i^{t+1}), P_{trans}(a_i^t, b_i^{t+1}) P_{trans}(b_i^t, a_i^{t+1})\}.$$

If in some block the two common haplotypes of a certain genotype originate from the same common haplotype, then the two possible matchings are identical. In that case, we preform the procedure on the haplotypes in the closest flanking blocks that have distinct common haplotypes (using all the haplotypes in theses blocks). This heuristic procedure aims to reveal longer range dependency between blocks.

4. RESULTS

Our algorithm was implemented in a C++ program called GERBIL (GEnotype Resolution and Block Identification using Likelihood). In the implementation, all initial parameters are chosen at random, the complete procedure is repeated 100 times and the maximal likelihood solution is selected. Running times on 2 GHz Pentium PC were less than 1 minute for resolving one block of 20 SNPs with 150 genotypes. Partitioning into blocks and phasing for a few hundred SNPs took several hours. GERBIL will be available in the future at www.cs.tau.ac.il\~rshamir.

We applied GERBIL to two published data sets, and compared the results to prior analysis of the same data. We describe how we dealt with missing data entries and outline

the methods that we used to evaluate the results, and then present the results on each data set.

Missing entries in the genotype matrix were completed in the original data, before the algorithm is performed, by the following heuristic. For each missing entry, we look at the window that spans 15 sites before and 15 sites after this site, and seek the closest other genotype within this window, where closeness is measured by the number of matching entries. The missing entry is then completed as the site value in that closest genotype. For an alternative approach to complete missing entries see [5].

4.1 Measures for Comparing Solutions

The data set of Daly et al. [4], on which we tested GERBIL, could be resolved to a large extent using pedigrees. The pedigree-based solution was assumed to be correct, and we used three methods for comparing different phasing solutions to it:

1. **Block Error Rate** - This test measures the error rate in a solution of a specific single block, w.r.t. the true solution. Let the two true haplotypes for genotype $\boldsymbol{g_i}$ be $\boldsymbol{t_1^i}, \boldsymbol{t_2^i}$ and let the two inferred haplotypes be $\boldsymbol{h_1^i}, \boldsymbol{h_2^i}$. Define the number of errors in genotype g_i as $e_i = \frac{1}{2}\min\{[d(\boldsymbol{t_1^i}, \boldsymbol{h_1^i}) + d(\boldsymbol{t_2^i}, \boldsymbol{h_2^i})], [d(\boldsymbol{t_1^i}, \boldsymbol{h_2^i}) + d(\boldsymbol{t_2^i}, \boldsymbol{h_1^i})]$, where d is the Hamming distance. If the number of heterozygote sites in genotype $\boldsymbol{g_i}$ is r_i, then the error rate is $\frac{\sum_{i=1}^{n} e_i}{\sum_{i=1}^{n} r_i}$.

2. **Average Block Error Rate** - This test measures the error rate in a multi-block solution with respect to the true solution. Let e^j be the total number of errors in block j (the numerator in the expression above), and let r^j be the total number of heterozygote sites in the the genotypes in block j (the denominator in the expression above), and let B be the number of blocks in the matrix. The measure is $\frac{\sum_{j=1}^{B} e^j}{\sum_{j=1}^{B} r^j}$.

3. **Switch Test** [14] - This test assumes matching of block pairs has been performed. It compares two solutions, $h = (h_1, h_2), t = (t_1, t_2)$ each of which is a pair of complete haplotype rows of sister chromosomes. Define the number of switches between h and t as the minimum number of times one has to 'jump' from one haplotype in h to the other in order to obtain t, when scanning the haplotypes from end to end. An example of switch test is shown in Table 1. This test is arguably more adequate than just counting the number of errors as above, since a whole group of errors can be corrected by changing the single decision to switch the group with that on the other haplotype. The total number of switches divided by the total number of heterozygote sites is called the *switch rate*.

4.2 Chromosome 5p31 Genotypes

The data set of Daly et al. [4] contains 129 pedigrees of father, mother and child, each genotyped at 103 SNP sites in chromosome 5p31. The original children data contain 13287 typed sites, of which 3873 (29%) are heterozygote alleles and 1334 (10%) are missing. After pedigree resolving, only 4315 (16%) of the 26574 single SNPs remained unknown (unresolved or missing data). Following [5], we used only the genotypes of the children and compared our solution to the pedigree-based solution from [4].

t	h
1: 11110001111	1: 00000000000
2: 00001110000	2: 11111111111

Table 1: Example of switch test. The number of switches that has to be done on h in order to obtain t is 2. Viewed as a single block, the minimum number of errors between the solutions is 3.

As a first step we applied GERBIL separately on each of the original blocks reported by Daly et al. The differences between the common haplotypes calculated by us and the true ones (which were constructed using the pedigrees) are minor: only in 4 common haplotypes there is a difference. In total, 10 bases out of 344 (2.9%) differed.

The results of GERBIL for resolving and block partitioning are presented in Table 2. We identified 8 blocks with total log likelihood of -4112.45, compared with log likelihood of -4647.36 of the solution of Daly et al., using the optimal model parameters for that solution. In each block 4-5 common haplotypes were found. The total number of switches in the haplotype matrix was 115 (3%). The average block error rate was 0.7%.

We compared the performance of our algorithm to two previously published phasing algorithms: the algorithm of Eskin et al. [5], which uses the perfect phylogeny criterion (see also [1]), and HaploBlock of Greenspan and Geiger [8], which resolves the genotypes by constructing a Bayesian network. The solution of [5] was taken from [25], and the solution of [8] was obtained by running HaploBlock [26] on the raw data. Table 3 compares the results of the three algorithms using the different error measures. In the switch test criterion, GERBIL made 29% less errors than Eskin et al., and 8% fewer errors than HaploBlock. With respect to average block error, GERBIL made 43% less errors than Eskin et al., and 62% less errors than HaploBlock.

Since HaploBlock partitioned the data into four blocks only, one could argue that the better results that we obtained were due to the increased number of blocks. To test it, we ran GERBIL on the blocks of HaploBlock, applying only the resolving procedure on the given partition, and the number of common haplotypes was assigned to be $k = 5$. The results are presented in Table 4. Our average block error rate was 30% less than HaploBlock.

4.3 OPRM1 Genotypes and Phenotypes

The data set of Hoehe et al. [10] consists of 172 genotypes and 25 SNPs. The SNPs are from the human μ opioid receptor gene (OPRM1) on chromosome 6, which is known to be related to morphine tolerance and dependence [16]. For each individual, its disease phenotype to substance (heroine and cocaine) dependence is available (case / control). No pedigree information is available and thus the true haplotypes are not known. Instead, we ran GERBIL on the data, with and without block partitioning, and tried to find association with the disease phenotype using the resolved haplotypes.

We first used GERBIL to resolve the data as a single block (Table 5). In all GERBIL runs we allowed four common haplotypes. In order to check for association between the resolved haplotypes and disease phenotype, we calculated

SNPs	Common Haplotypes of GERBIL	α_i	Number of Errors	Number of Heterozygotes	Error Rate
1 - 14	GGACAACCGTTACG	0.83	0	450	0
	AATTCGTGGCCCAA	0.13			
	AATTCGTGGTTACG	0.02			
	GGACAACCGCCCAA	0.01			
15 - 28	CCGGAGACGACGCG	0.55	6	583	0.0103
	TGACTGGTCGCTGC	0.24			
	CCGCAGACGACTGC	0.19			
	TGGCAGGTCGCTGC	0.02			
29 - 38	CCCGGATCCA	0.72	1	393	0.0025
	TATAACCGCG	0.17			
	CCCAACCCCA	0.06			
	CCCAACCCAA	0.05			
39 - 44	GCCCGA	0.54	4	210	0.0190
	CCCTGA	0.19			
	CTCTGA	0.14			
	CCATAC	0.13			
45 - 72	TCCCTGCTTACGGTGCAGTGGCACGTAT	0.7	2	942	0.0021
	CTCCCATCCATCATGGTCGAATGCGTAC	0.24			
	CCATCACTCCCCAGACTGTGATGTTAGT	0.05			
73 - 91	TGCACCGTTTAGCACAACA	0.59	9	711	0.0127
	ATTAGTGTTTGACGCGGTG	0.16			
	ATCAGTGATTAGCACGGTG	0.13			
	ATCAGTGATTAGCACGGTG	0.07			
	ATCTCTAATTGGCGTGACG	0.05			
92 - 98	GTTCTGA	0.57	4	294	0.0136
	TGTGTAA	0.28			
	TGTGCGG	0.15			
99 - 103	CGGCG	0.45	0	290	0
	TATAG	0.42			
	TATCA	0.14			
total			26	3873	0.0067

Table 2: Results of GERBIL in phasing and block partitioning on the data of Daly et al.

χ^2 scores, for each common haplotypes vs. the rest, and also for all the haplotypes together. The results of the association tests are summarized in Table 7. For all the common haplotypes together, the p-value was 0.02378; for the third common haplotype, the p-value was 0.0234.

Next, we ran GERBIL with blocks partitioning. We discovered two blocks (Table 6). We checked disease association in the same fashion. The first block was clearly associated to the disease with p-value of 0.0031. In the second block, only the second haplotype was associated with p-value of 0.0385. It is quite clear that association is much more prominent in the first block.

Hoehe et al. resolved the genotypes using the MULTIHAP [24] software, which is based on [6]. Then, the haplotypes were hierarchically clustered into a tree using an agglomerative nearest neighbor approach. The p-values of comparisons of haplotype frequencies and of cases and controls were calculated between the clusters calculated at each level of the hierarchical clustering. The lowest p-value which was achieved was 0.017.

In order to compare the significane of the two solutions, one has to correct for multiple testing. Since we performed eight different tests for two blocks (four tests in each block), after Bonferroni correction our p-value, for common haplotype number 4 in block number 1 was 0.0360. Hoehe et al. performed n different tests, where n is the number of haplotypes. To correct Hoehe et al. score for multiple testing, we multiplied their score by the number of distinct *groups* of haplotypes in their dendrogram, which was 5. Notably, this correction is much less strict than ours. Thus, after multiple testing correction, Hoehe et al. p-value was 0.0850, which is 2.3-times larger than ours. Hence, our solution achieves a much better statistical significance.

5. CONCLUDING REMARKS

We have introduced a new stochastic model for genotype generation, based on the biological finding that genotypes can be partitioned into blocks, and in each block, a small number of common haplotypes is found. Our model defined the notion of a probabilistic common haplotype, which might have different forms in different genotypes, thereby accommodating errors and rare mutations. We were able to define a likelihood function for this model. Finding the optimal parameters of the model was achieved using an EM algorithm, according to the maximum likelihood approach.

In tests on real data, our algorithm gave more accurate results than two recently published phasing algorithms [5, 8]. The haplotypes and blocks identified by the algorithm on case/control genotype data of the OPRM1 gene [10] led to finding more significant association with substance abuse phenotype.

Although our model finds a block partitioning that maximizes the overall likelihood, it performs resolving and block partitioning first, and then matches pairs of blocks along the chromosome as a postprocessing step. We plan to unite those two steps into a complete, single model. An additional open problem is to treat the missing data as a part of the model. We believe that solving these problems will lead to additional improvement in performance. Finally, the block patterns are sometimes unclear, and it has been argued that less restrictive models of haplotypes generation are needed (e.g., [23, 2]). We intend to generalize our approach in this spirit.

Acknowledgments

Ron Shamir was supported by a grant from the Israel Science Foundation (grant 309/02). We wish to thank Margret

Hoehe for providing us with the OPRM1 data and for useful comments. We thank Gideon Greenspan, Dan Geiger, Yoav Benjamini, Elazar Eskin and Koby Lindzen for helpful discussions and comments.

6. REFERENCES

[1] V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogeny: A direct approach. Technical Report UCDavis CSE-2002-21, 2002.

[2] V. Bafna, B. V. Halldorsson, R. Schwartz, A. Clark, and S. Istrail. Haplotyles and informative SNP selection algorithms: Don't block out information. In *Proceedings of The Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 19–27, 2003.

[3] A. Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7(2):111–22, 1990.

[4] M. J. Daly et al. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29(2):229–232, 2001.

[5] E. Eskin, E. Halperin, and R. M. Karp. Large scale reconstruction of haplotypes from genotype data. In *Proceedings of The Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 104–113, 2003.

[6] L. Excoffier and M. Slatkin. Maximum-likelihood estimation of mollecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):912–7, 1995.

[7] S. B. Gabriel et al. The structure of haplotype blocks in the human genome. *Science*, 296:2225–2229, 2002.

[8] G. Greenspan and D. Geiger. Model-based inference of haplotype block variation. In *Proceedings of The Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 131–137, 2003.

[9] D. Gusfield. Haplotyping as perfect phylogeny: conceptual framework and efficient solutions. In *Proceedings of The Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 166–175, 2002.

[10] M. R. Hoehe et al. Sequence variability and candidate gene analysis in complex disease: association of μ opioid receptor gene variation with substance dependence. *Human Molecular Genetics*, 9:2895–2908, 2000.

[11] G. Kimmel, R. Sharan, and R. Shamir. Identifying blocks and sub-populations in noisy SNP data. In *proceedings of the Third Workshop on Algorithms in Bioinformatics (WABI)*, pages 303–319, 2003.

[12] M. Koivisto et al. An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries. In *Proceedings of the Pacific Symposium on Biocomputing (PSB)*, volume 8, pages 502–513, 2003.

[13] L. Kruglyak and D. A. Nickerson. Variation is the spice of life. *Nature Genetics*, 27:234–236, 2001.

[14] S. Lin, D. Cutler, M. Zwick, and A. Chakravarti. Haplotype inference in random population samples. *American Journal of Human Genetics*, 71:1129–1137, 2002.

[15] J. Long et al. An EM algorithm and testing strategy for multiple-locus haplotypes. *American Journal of Human Genetics*, 56(3):799–810, 1995.

[16] H. W. Matthes et al. Loss of morphine-induced analgesia, reward effect and withdrawal symptoms in mice lacking the mu-opioid-receptor gene. *Nature*, 383:819–823, 1996.

[17] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley and Sons, inc., 1997.

[18] T. Niu et al. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *American Journal of Human Genetics*, 70(1):157–69, 2002.

[19] N. Patil et al. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294:1719–1723, 2001.

[20] R. Sachidanandam et al. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 291:1298–2302, 2001.

[21] M. Stephens et al. A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics*, 68(4):978–89, 2001.

[22] C. Venter et al. The sequence of the human genome. *Science*, 291:1304–51, 2001.

[23] J. Wall et al. Assessing the performance of the haplotype block model of linkage disequilibrium. *American Journal of Human Genetics*, 73(3):502–515, 2003.

[24] `http://mahe.bioinf.mdc-berlin.de`.

[25] `http://www1.cs.columbia.edu/compbio/hap/`.

[26] `http://www.cs.technion.ac.il/Labs/cbl`.

[27] K. Zhang, M. Deng, T. Chen, M. Waterman, and F. Sun. A dynamic programming algorithm for haplotype block partitioning. *Proc. Natl. Acad. Sci. USA*, 99(11):7335–9, 2002.

[28] K. Zhang et al. Dynamic programming algorithms for haplotype block partitioning: applications to human chromosome 21 haplotype data. In *Proceedings of The Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 332–340, 2003.

Algorithm	GERBIL	Eskin et al.	HaploBlock
Total Number of Errors	26	46	69
Average Block Error Rate	0.0067	0.0119	0.0178
Switch Test	115	163	125
Switch Rate	0.0297	0.0421	0.0323

Table 3: Performance of three algorithms for phasing and block partitioning. The error measures evaluate the solutions by GERBIL, by Eskin et al. and by Greenspan and Geiger, on the data of Daly et al., compared to the true solution.

Algorithm	Block:	1-24	25-36	37-91	92-103	total
	Number of Heterozygotes	846	537	1906	584	3873
GERBIL	Number of Errors	13	0	29	6	48
	Block Error Rate	0.0154	0	0.0152	0.0103	0.0124
HaploBlock	Number of Errors	28	3	31	7	69
	Block Error Rate	0.0331	0.0056	0.0163	0.0120	0.0178

Table 4: Resolving performance: Comparison of GERBIL and HaploBlock on the data of Daly et al. Here GERBIL used the blocks produced by HaploBlock and performed genotypes resolving only.

Common Haplotype Number	Common Haplotype	α_i
1	000000000000000000000100	0.817
2	101010000000101000000100	0.077
3	101011000000101000001100	0.054
4	000000000000010000010101	0.053

A

Haplotype Number	1	2	3	4	Total
Cases	218	24	19	13	274
Controls	62	2	0	6	70
Total	280	26	19	19	344

B

Table 5: GERBIL results on the OPRM1 data, without block partitioning. A: the common haplotypes identified. B: frequencies of cases and controls for the resolved common haplotypes.

	Block 1: SNPs 1-17		Block 2: SNPs 18-25	
Common Haplotype Number	Common Haplotype	α_i	Common Haplotype	α_i
1	00000000000000000	0.50	00000100	0.85
2	00000000000000100	0.21	10000000	0.097
3	00000000001000000	0.15	00010101	0.035
4	10101000000010100	0.13	00010001	0.017

A

Haplotype Number	1	2	3	4	Total
Cases	137	54	40	43	274
Controls	30	24	14	2	70
Total	167	78	54	45	344

B

Haplotype Number	1	2	3	4	Total
cases	239	21	10	4	274
controls	55	11	2	2	70
total	294	32	12	6	344

C

Table 6: Results of GERBIL on the OPRM1 data, with block partitioning. A: common haplotypes identified by GERBIL. B, C: frequencies of common haplotypes in the resolved data in cases and controls. B: first block; C: second block.

Haplotype Checked	One Block	Two Blocks: First Block	Two Blocks: Second Block
1 vs. rest	0.0839	0.2859	0.0667
2 vs. rest	0.0955	0.0093	0.0385
3 vs. rest	0.0234	0.2676	0.747
4 vs. rest	0.211	0.0045	0.4255
all vs. all	0.02378	0.0031	0.1648

Table 7: Association test results on the OPRM1 data: P-value χ^2 test results on the haplotypes resolved by GERBIL, with and without block partitioning.