Algorithm for haplotype resolution and block partitioning for partial XOR-genotype data

Nadezhda Sazonova a,*, Edward Sazonov a, E. James Harner b

aDepartment of Electrical and Computer Engineering, Clarkson University, Potsdam, NY 13676, USA
bDepartment of Statistics, West Virginia University, Morgantown, WV 26505, USA

Abstract

Problems of haplotyping and block partitioning have been extensively studied with regard to the regular genotype data, but more cost-efficient data called XOR-genotypes remain under-investigated. Previous studies developed methods for haplotyping of short-sequence partial XOR-genotypes. In this paper we propose a new algorithm that performs haplotyping of long-range partial XOR-genotype data with possibility of missing entries, and in addition simultaneously finds the block structure for the given data. Our method is implemented as a fast and practical algorithm. We also investigate the effect of the percentage of fully genotyped individuals in a sample on the accuracy of results with and without the missing data. The algorithm is validated by testing on the HapMap data. Obtained results show good prediction rates both for samples with and without missing data. The accuracy of prediction of XOR sites is not significantly affected by the presence of 10% or less missing data.

1. Introduction

Single nucleotide polymorphisms (SNPs) are the most common type of mutations expressed by changes in a single position within DNA sequence, which are observed in at least 1% of the population. Most SNPs are bi-allelic meaning that they are defined by only two possible nucleotides (alleles) at their specific positions. Variation in the human genome underlies the differentiating features present in the population. Humans are diploid organisms, i.e., each chromosome is made of two distinct copies which are separately called haplotypes. A pair of haplotypes (inherited from the mother and the father) is a result of genotype decomposition and is considered to be a genotype with a known phase, i.e., genotypes with alleles assigned to one of the two chromosomes.

Knowledge of haplotypes decomposition of any genotype is extremely valuable in the large-scale analysis of complex human diseases, which are affected by combinations of multiple linked mutations and a set of environmental factors. For this reason, haplotype-based analysis has proven to be much more powerful in mapping complex human diseases than single-locus (SNP) based studies. Available molecular technologies do not allow cheap and efficient haplotype sequencing (also called haplotyping or haplotype resolution). Thus, the problem of haplotyping heavily relies on computational methods. Recent studies have demonstrated that the human genome has discrete block structures with limited haplotype diversity within each block. Using haplotypes partitioned into blocks allows yet another significant improvement in the process of mapping complex human diseases since it narrows the study of the genotype (or a pair of haplotypes) to the analysis of a specific region.

There has been significant effort done in developing haplotyping and block partitioning methods [1–15], which resulted in fast and efficient algorithms. The demand for a new type of genotype data characterized by some degree of incompleteness has appeared more recently. This type of data can be produced in an extremely cost-efficient way. In particular, there is a number of less expensive molecular technologies (such as Denaturing High-Performance Liquid Chromatography (DHPLC) [16]) that produce genotypes with partial information with regard to SNPs: they can only distinguish if a particular SNP position is heterozygous or homozygous for a given individual, but provide no information regarding the type of allele (out of the two possible) it carries at each position. This DHPLC method permits significant reduction of the cost of genotyping compared to conventionally used genotyping methods.

Genotypes produced as a result of such partial genotyping technologies are called XOR-genotypes. Efficient methods of haplotyping and block partitioning for XOR-genotypes are still under investigation. Thus, it is essential to develop effective computational methods that can use XOR-genotypes for haplotyping and block partitioning.

Up to date there exists only one strategy [17–19] that was developed to solve the problem of XOR-haplotyping. The core of
this strategy is the construction of the perfect phylogeny tree out of collection of XOR-genotypes (using GREAL – an implementation of the algorithm for Graph Realization). The constructed phylogenetic tree is then used to select few individuals which need to be fully genotyped. The phylogenetic tree and selected fully genotyped individuals are then applied to the rest of the sample to make complete haplotype inference. This algorithm also requires availability of full genotypes for completely homogeneous individuals. Although this strategy requires at most three heterogeneous individuals to be fully genotyped, there is still room for further improvement. First, the Perfect Phylogeny assumption used in GREAL is only valid within short sequences of SNPs or blocks, which usually are limited by 30 SNPs [18] and almost never exceed size of 100 SNPs. Thus, GREAL is not well suited for the long-range genotype data. Second, a special selection of individuals to be fully genotyped is performed as one of the steps. Thus, no arbitrarily mixed collections of XOR and full genotypes are assumed, which may be required to be practical in real world applications. Third, the strategy was implemented as several steps that need to be conducted separately: application of GREAL, introduction of the fully genotyped sequences, resolution of the rest of the sample. A direct XOR-genotypes-to-haplotypes algorithm would be preferred for practical applications. Forth, the utility of the XOR-haplotyping could be further improved by taking into consideration missing data that is often an unavoidable component of the real data samples.

This paper describes a new method for performing XOR-haplotyping. Our algorithm XOR-HAPLOGEN is designed to introduce a number of substantial improvements over existing methods: the proposed algorithm is able to process very long sequences of SNPs (the size of the input genotype matrix is currently limited by approximately 350,000 entries); the algorithm can process arbitrarily mixed XOR-genotype data (part of the data can be represented by XOR and part by the full genotype sequences); XOR-HAPLOGEN produces the haplotype matrix for the entire set of input individuals together with a block structure and its profile (lists of haplotype patterns, their frequencies for each block and block boundary scores); finally, XOR-HAPLOGEN also incorporates missing data into the model and, thus, is able to infer missing data entries. XOR-HAPLOGEN is an improved version of our earlier developed algorithm HAPLOGEN [20] for haplotyping and block partitioning that was originally designed for use on a fully genotyped set of individuals and showed accuracy and speed comparable to that of existing methods for haplotype resolution [11,13–15]. Procedurally XOR-HAPLOGEN is an extended version of HAPLOGEN to enable it to process XOR-genotypes in addition to regular genotypes. XOR-HAPLOGEN is implemented as an R-package and has polynomial time complexity $O(n^2m^3)$, where $n$ is the number of individuals and $m$ is the length of genotypes in a sample. The proposed algorithm is designed to process any number of XOR-genotypes with the only requirement that there would be a non-empty set of fully genotyped individuals in the sample. XOR-HAPLOGEN achieves good accuracy in haplotype decomposition with and without presence of missing data and is not significantly affected by the percentage of fully genotyped individuals in a sample.

Operation of XOR-HAPLOGEN is loosely based on the principles of genetic algorithms: the population of potential solutions is encoded as binary strings; evolution of the population is performed by application of genetic operations such as mutation and selection based on the fitness value of each individual.

The rest of the paper is organized as follows: Section 2 describes the main assumptions and concepts used in the proposed algorithm. Section 3 explains the proposed algorithm in details and Section 4 provides results from validating the algorithm on publicly available data sets. Conclusions and directions for future work are given in Section 5.

2. Assumptions and underlying concepts

The genotype data used for this study are encoded by a genotype matrix with entries from the set $\{0, 1, 2, 3, 9\}$, where $0$ or $1$ encode homogeneous sites with known alleles, $2$ encodes heterogeneous sites, $3$ encodes homogeneous sites with unknown alleles and $9$ encodes sites with missing data. The goal of XOR-HAPLOGEN is to produce a $(0, 1)$ haplotype matrix where every two rows correspond to a particular genotype. Additionally the algorithm computes the block structure with the complete profiles (list of distinct patterns and their frequencies) for each block. The assumptions (parsimony principle) and the basic approach (block identification criteria and block-extension algorithm) described below are inherited from the previous version of the algorithm, i.e., HAPLOGEN.

2.1. Parsimony principle

The first basic assumption that defines the framework of the proposed approach is the principle of pure parsimony which is a widely used, empirically supported idea [1]. The principle of pure parsimony aims at achieving the minimum number of distinct haplotypes to resolve the input genotypes. In XOR-HAPLOGEN this principle is reflected in the fitness function evaluating the quality of a potential solution and in the selection of the best solution from the population of potential solutions.

2.2. Block identification criteria

The second assumption is represented by the operational description of a block, which is the linkage disequilibrium (LD)-based block identification criterion. Since blocks exhibit low haplotype diversity they also tend to have high degree of LD. On the other hand, regions with high LD tend to have limited haplotype diversity [2]. Haplotype diversity, in turn, is represented by a number of common (covering more than 80% of the data) haplotype patterns in a block.

Most popular measures commonly used to evaluate the degree of linkage disequilibrium are $\chi^2$ and $D^2$ [21] which are pair-wise LD measures. The obvious deficiency of these measures is that they are limited to two loci. Several approaches have been developed to amend this deficiency, but most of them either do not describe multilocus LD directly or are computationally inefficient. The multilocus LD measure called the Normalized Entropy Difference $\varepsilon$ (NED) [21] overcomes these limitations. A sequence of $m$ bi-allelic loci can be seen as a system with the possible haplotypes as its states. This sequence can assume $2^m$ states (haplotypes) of which only few are present. The NED ($\varepsilon$) measures the degree of deviation of the entropy of the loci sequence from its equilibrium state, is defined as [21]:

$$\varepsilon = \frac{S_e - S_0}{S_e} = \frac{\Delta S}{S_e} = 1 - \frac{S_e}{S_0} \quad (1)$$

where $S_e$ is the entropy in the equilibrium case and $S_0$ is the actual entropy of the loci sequence, computed as suggested in [21].

XOR-HAPLOGEN uses $\varepsilon$ throughout the process of block partitioning in order to detect valid blocks. It is also possible to assess the significance of this LD coefficient. It has been shown [21] that the statistic $2n\Delta S$ approximately follows $\chi^2$ distribution with $2^n - (m - 1)$ degrees of freedom and can be used to test the significance of the deviation of the system from its linkage equilibrium within any particular block.
2.3. Block-extension algorithm

In XOR-HAPLOGEN the block structure of a haplotype decomposition is determined by the block-extension algorithm, which is based on the fact that blocks are usually quite long [2,5] and is implemented by sequentially combining smaller-size blocks into the bigger ones. The block-extension algorithm is applied to the currently available haplotype decomposition.

The block-extension algorithm is a greedy algorithm which incorporates the idea similar to one in the hierarchical clustering: initially every 1 position (SNP) is considered to be a block, then at each iteration pairs of blocks merge into longer blocks if a specific criterion is met. Overlapping blocks are also considered and their strength is compared in order to form the strongest block out of several consecutive overlapping blocks. The procedure is iteratively repeated until merging is no longer efficient, i.e., the formation of longer blocks does not satisfy criterion’s threshold in any of the proposed merges. The block-extension algorithm can be used with any block identification criterion that allows equivalent comparison for the blocks of different lengths. NED is chosen as one of such possible criteria. However, it should be pointed out that the NED criterion used for block identification has certain limitations. Namely, the asymptotic distribution of its value works best for the blocks usually not exceeding length of 8–10 SNPs, but for longer sequences the NED is always insignificant. To amend this limitation the block-extension algorithm can be applied using an additional block identification criterion called coverage of the common patterns: 4 or 5 patterns with the highest frequencies within a block are assessed on their coverage of the data. The admissible threshold is set to 80% and the overlapping blocks are compared in their coverage to create the one with the highest coverage. The number of the most frequent patterns (4 or 5) is fixed in advance, however, a greater number of common patterns is not practical [5,13].

Due to these considerations, the block-extension algorithm is divided into two parts: first, the NED-based and then the coverage-based block-extension algorithms are applied. NED-based algorithm produces blocks that in general do not exceed length of 10 SNPs, and the coverage-based algorithm then uses these smaller-size blocks to create possibly longer blocks each providing at least 80% coverage.

3. XOR-HAPLOGEN

The outline of XOR-HAPLOGEN algorithm is illustrated in Fig. 1 and described below in details. Procedurally, the algorithm is a combination of two levels of optimization search represented by the inner and outer loops with different optimization criteria for each level. The goal of each iteration of the outer loop is to produce an updated block structure based on the current haplotype decomposition, and the goal of each iteration of the inner loop is to compute the updated haplotype decomposition and evaluate how well it satisfies the optimization criterion given the current block structure. Thus, the best haplotype decomposition for a given block structure is found by the series of inner (lower-level) iterations within every outer iteration, and the best haplotype decomposition together with the block structure are found by the series of the outer (upper-level) iterations.

Genetic search is performed at the inner loop level and can be described as follows. The individuals are (0, 1)-strings representing distinct haplotype patterns within each block. At each iteration of the inner loop there are as many populations as there are blocks. Thus, inner loop provides multiple independent genetic searches (one for each block) simultaneously. The size of a population within any particular block is not fixed in advance and changes from iteration to iteration as will be discussed further in Sections 3.1 through 3.4. The selection and reconstruction process used in the inner loop are performed in such a way so that the haplotype population is possibly reduced from one generation to another (to guarantee the parsimonious principle). Every iteration of the inner loop consists of the following steps: mutation, fitness evaluation of the haplotype patterns, selection of patterns, reconstruction of haplotype pairs and, finally, selection of the best solutions (i.e., haplotype resolution within each block). The number of inner loop iterations is proportional to the number of genotypes in the data.

The purpose of the outer loop is to deal with the whole-length genotypes and haplotypes. An iteration of the outer loop performs matching of the haplotype patterns in the adjacent blocks to obtain whole-length haplotype decompositions. The outer loop also reviews and adjusts block structure, evaluates the whole-length solution and selects the best global solution by comparing the current global solution (given by haplotypes decomposition and block partitioning of the entire genotype sample) to the best previous solution. The number of blocks, as well as their boundaries, varies from one outer loop iteration to another. The number of outer loop iterations is proportional to the length of the SNP sequence of the data.

3.1. Initialization

Each haplotype is represented by a binary string of length $m$. Initialization is done by randomly obtaining a feasible decomposition, i.e., for each ambiguous (heterogeneous) position of each genotype permutations $(0, 1)$ and $(1, 0)$ are randomly assigned with equal probabilities to the pair of haplotypes. For XOR-genotypes their homogeneous positions are assigned equal values (0 or 1) to both alleles, this is done randomly with equal probabilities with regard to 0 and 1. Special care is taken at this stage in the sites
with missing data. Namely, the initialization at sites with missing data is performed randomly so as to assign with equal probabilities one of the four possible values (0, 0), (1, 1), (1, 0) or (0, 1) to the pair of haplotypes. Although at the initialization step allele frequencies at XOR, heterogeneous and missing data sites are close to 0.5, they are optimized during genetic search as a result of haplotypes pattern selection/haplotypes reconstruction and mutation. Current block structure is found by applying the block-extension algorithm to the trivial block structure where each SNP position is considered to be a block.

3.2. Fitness evaluation

The fitness function \( f(h) \) of a haplotype pattern \( h \) is the probability of occurrence of this haplotype given the genotype data and the current block partition. Random mating is assumed. Within each block, the probability of a haplotype pattern \( h_{ib} \) given the genotype data \( G \) can be described as:

\[
P(h_{ib}|G) = \sum_{g \in G} \prod_{j \in |G|} P_j^{1_{g_j=1}}(1 - P_j)^{1_{g_j=0}},
\]

where \( G \) is the collection of genotypes \( g \) that are compatible within the current block (i.e., could be used for haplotype reconstruction into haplotypes) with pattern \( h_{ib} \). For every such genotype \( g \) the probability of the pattern \( h_{ib} \) (where \( I_g \) is the collection of indices of ambiguous, missing and homogeneous XOR sites in genotype \( g \) within block \( b \)) is

\[
\prod_{j \in |G|} P_j^{1_{g_j=1}}(1 - P_j)^{1_{g_j=0}}.
\]

Here \( P_j \) is the probability of the allele 1 in the \( j \)th position of haplotype \( h_i \) (respectively, \( 1 - P_j \) is the probability of allele 0). It is clear that the greater the number of genotypes compatible with \( h_{ib} \), the greater is the probability of this pattern given the data. Contribution of the entirely homogeneous fully genotyped individuals to the fitness of the corresponding haplotype pattern is doubled. Selection of the patterns with high fitness \( f(h_{ib}) = P(h_{ib}|G) \) will guarantee the parsimonious principle where the frequencies of haplotypes should be maximized in order to provide a minimum set of distinct haplotypes. Estimation of the relative frequency \( p_j \) of the allele 1 at each site is performed using a Bayesian approach by using the Beta distribution as a prior distribution.

3.3. Selection of the set of fittest patterns within each block

Selection of the fittest subpopulation within each block is performed based on the fact that every genotype can be potentially resolved (“covered”) by at least one pattern out of the selected ones. A subset of individuals (haplotype patterns) in the current population is selected randomly without replacement according to their fitness values. The exact size of this subset is not known in advance, but is determined in the selection process itself as described below.

Selection of patterns within blocks is proportional to the fitness of a pattern within a block given by

\[
P(h_{ib}|G) = \sum_{g \in G} \prod_{j \in |G|} P_j^{1_{g_j=1}}(1 - P_j)^{1_{g_j=0}}.
\]

Fig. 2 provides a simple example to explain the goal of the selection step. Relationship between genotypes within any particular block and haplotype patterns currently resolving these genotypes can be represented as a bipartite graph \((X, Y)\), where \(X\) is the set of genotypes and \(Y\) is the set of currently available patterns. Every edge \( xy \) represents a possibility of resolving genotype \( x \) with pattern \( y \). There are at least 2 edges since every genotype is currently resolved by two patterns and can be potentially resolved by some other patterns. A subset of \( Y \) represented by black filled vertices is an example of a minimal set covering all \( X \) vertices. The goal is to select such a subset of haplotypes patterns.

The sought subset of \( Y \) is found in the following manner: all genotypes (vertices of \( X \)) are marked as “covered” (i.e., potentially covered) by the first randomly selected haplotype pattern; if every next selected pattern does not provide any additional coverage, it is removed from further consideration (and is not selected); otherwise, new vertices are also marked as “covered” and the pattern is considered to be selected; this process continues until all genotypes are covered by potentially resolving patterns.

3.4. Reconstruction of haplotype patterns within each block

A subset of the fittest haplotype patterns in the current population is used to construct the next generation of haplotypes for each block. The new generation is built by one-by-one consideration of the genotypes. In this process the patterns selected at previous step are used to resolve as many as possible genotypes in the sample. In case of multiple applicable patterns random selection of one of them proportional to the fitness value is performed.

Reconstruction is the last step in each inner loop iteration. After that the selection of the best solutions (one for each block) is performed by computing the number of common haplotypes, ncomp, in each block of the current haplotypes decomposition and comparing it to its previous best value. Solution corresponding to the minimum of the two values is selected as the best for each block.

3.5. Matching of the pairs of haplotypes in consecutive blocks

Inter-block transitions (connections between adjacent blocks) present the choice of the best pairing (tiling) of the four haplotype patterns at the block boundary for every genotype. Matching of the pairs of block is based on the observed fact that haplotype blocks exhibit long-range dependency [5,13]. If a certain genotype decomposes into the haplotype patterns \( h_{1a} \) and \( h_{1b} \) in the first block \( a \) (in an adjacent pair of blocks) and patterns \( h_{2a} \) and \( h_{2b} \) in the second block \( b \), then there are only 2 possible options for the pairing: \( (h_{1a}, h_{2a}) \), \( (h_{1a}, h_{2b}) \), \( (h_{1b}, h_{2a}) \), \( (h_{1b}, h_{2b}) \). The choice is based
on selecting the greater of the two probabilities \( P(h_{1a}, h_{2a}) = P(h_{1a}, h_{2b}) \cdot P(h_{1b}, h_{2a}) \) and \( P(h_{1b}, h_{2a}) = P(h_{1a}, h_{2b}) \cdot P(h_{1b}, h_{2a}) \). Each of the probabilities \( P(h_{1a}, h_{2a}) = P(h_{1b}, h_{2a}) \) is estimated by computing the fraction of genotypes which these two consecutive patterns can potentially resolve at the same time.

### 3.6. Adjustment of the block structure

After the selection step, reconstruction and inter-block transitions are performed to update the block structure. The current block structure could be completely reset (reinitialized) with probability 0.5 (probability of reinitialization). If the block structure is reinitialized then the block-extension algorithm is applied to the single SNPs as the next step. Otherwise (if not reinitialized), the block structure is updated in such a way as to guarantee that the former blocks no longer satisfying the threshold for the block identification are destroyed and new blocks are created by using the block-extension algorithm. The threshold for block destruction is set to be slightly higher (90% coverage) than the one for block creation (80%) since this allows for the search of more efficient blocks, i.e., those providing higher average coverage.

### 3.7. Evaluation of the current solution

The new solution in the form of the block structure (defined by the boundaries and the patterns within each block) and the whole-length haplotype decomposition for the entire genotype sample is evaluated according to the parsimonious principle. In each iteration, the current solution is compared to the current best solution using two criteria: the total number of common patterns across all blocks, \( ds \), and the number of distinct whole-length haplotypes (common and rare), \( tocompat \). Selection of the best solution is defined by the minimum of both of these criteria. The \( tocompat \) criterion is used to find the best block structure described by the long blocks with high coverage. The \( ds \) criterion is needed to predict the best set of the whole-length haplotypes and was used by other parsimony-based haplotype decomposition studies [5,22]. At this step it is also decided whether to continue search or to terminate the algorithm.

### 3.8. Mutation

This step is used to contribute to the variability of the population(s). Mutation is applied to the pairs of individuals (haplotypes) corresponding to the same genotype and performed separately for the heterogeneous and for homogeneous XOR sites. Mutation for heterogeneous sites is accomplished by switching 0 and 1 between chromosomes with some probability. Mutation for the XOR sites is a change of the allele (0 to 1, or 1 to 0) on both chromosomes simultaneously. The probability of a mutation is determined by the second order Markov chain transition probabilities estimated from the current generation of haplotypes. To ensure that good patterns found in previous iteration are not lost during mutation, it is only applied to the haplotype pairs with low coverage.

### 3.9. Calculation of scores for the block boundaries

The proposed algorithm also calculates estimates of probability of a border between every two adjacent SNPs to be a block boundary. This facilitates strength assessment of any particular block boundary. Since the algorithm goes through a series of iterations, each score is calculated as a proportion of the time the algorithm selects this position as a block boundary in its current best solution:

\[
\text{Scores} = \sum_{\text{iter}} b_{\text{best}},
\]

where \( b_{\text{best}} \) is a (0, 1)-vector representing current outer loop iteration’s best block boundaries solution: 1 at position \( j \) indicates that there is a block boundary to the left of that position, otherwise it is 0.

### 4. Results

We used the HapMap data described in [23,24] and publicly available at http://www.hapmap.org to test XOR-HAPLOGEN. In particular XOR-HAPLOGEN was applied to the CEU data which were collected in 1980 from US residents with northern and western European ancestry by the Centre d’Etude du Polymorphisme Humain (CEPH) and consist of the genotypes of 30 family trios. The HapMap web site provides the phased data which were obtained using the PHASE software. The program PHASE implements methods for estimating haplotypes from population genotype data [8]. The phasing process used the trio information where available so that the published haplotypes are of a very high accuracy [24]. The HapMap Consortium reported extremely low estimated switch rate (error occur every 8 Mb in CEU as given in [24]) which for all practical purposes can be regarded as negligible. We randomly selected a chromosome (chromosome 2) and took first 500 positions having the same location (first 500 positions) along the selected chromosome. These data contained 2.2% of loci that were homogeneous for all 30 genotypes. None of the loci was heterogeneous for all genotypes in the sample. On average, any given genotype had 21% of heterogeneous loci.

#### 4.1. Testing samples without missing data

The CEU data for 30 individuals were then used to test the XOR-HAPLOGEN algorithm in a scenario without missing data. A set of genotypes was selected randomly to be represented by XOR encoding such that the true homogeneous alleles were hidden. Different percentages of fully genotyped sequences were tested that gave rise to 10 data sets containing from 10% to 100% of fully genotyped individuals. As an example, the 10% (3 genotypes) of fully genotyped sample corresponds to 90% (27 genotypes) of the individuals encoded as XOR-genotypes. To eliminate genotype-specific effects on the result of prediction, for each percentage of included full genotypes we generated 10 samples with random selection of individuals to be XOR-genotyped. Several prediction errors were evaluated for each data set. The prediction error for the XOR positions (\( err_{xor} \)) was computed as the fraction of incorrectly predicted XOR sites out of the number of all XOR sites. To assess the amount of the incorrectly inferred homogeneous sites within the entire data (as

### Table 1

Performance of XOR-HAPLOGEN for different levels of full genotyping in the CEU sample.

<table>
<thead>
<tr>
<th>Percentage fully genotyped (%)</th>
<th>Average ( err_{xor} )</th>
<th>Average ( err_{all sites} )</th>
<th>Average ( swr )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.1119</td>
<td>0.1005</td>
<td>0.1696</td>
</tr>
<tr>
<td>20</td>
<td>0.0655</td>
<td>0.0522</td>
<td>0.1078</td>
</tr>
<tr>
<td>30</td>
<td>0.0675</td>
<td>0.0474</td>
<td>0.0836</td>
</tr>
<tr>
<td>40</td>
<td>0.0591</td>
<td>0.0357</td>
<td>0.0719</td>
</tr>
<tr>
<td>50</td>
<td>0.0509</td>
<td>0.0255</td>
<td>0.0588</td>
</tr>
<tr>
<td>60</td>
<td>0.0515</td>
<td>0.0207</td>
<td>0.0407</td>
</tr>
<tr>
<td>70</td>
<td>0.0550</td>
<td>0.0163</td>
<td>0.0381</td>
</tr>
<tr>
<td>80</td>
<td>0.0336</td>
<td>0.0056</td>
<td>0.0256</td>
</tr>
<tr>
<td>90</td>
<td>0.0325</td>
<td>0.0021</td>
<td>0.0210</td>
</tr>
<tr>
<td>100</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0180</td>
</tr>
</tbody>
</table>

* This error is a fraction of incorrectly inferred homogeneous sites out of all homogeneous sites (either in the XOR or fully genotyped form).
Table 2
Performance of XOR-HAPLOGEN for samples with different percentages of full genotypes in the presence of 10% missing data in the CEU data.

<table>
<thead>
<tr>
<th>Percentage fully genotyped (%)</th>
<th>Average err_{XOR}</th>
<th>Average err_{all}</th>
<th>Average swr</th>
<th>Average err_{miss}</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.1147</td>
<td>0.1032</td>
<td>0.1909</td>
<td>0.1796</td>
</tr>
<tr>
<td>20</td>
<td>0.0700</td>
<td>0.0558</td>
<td>0.1259</td>
<td>0.1318</td>
</tr>
<tr>
<td>30</td>
<td>0.0648</td>
<td>0.0454</td>
<td>0.1033</td>
<td>0.1214</td>
</tr>
<tr>
<td>40</td>
<td>0.0581</td>
<td>0.0351</td>
<td>0.0956</td>
<td>0.1154</td>
</tr>
<tr>
<td>50</td>
<td>0.0550</td>
<td>0.0275</td>
<td>0.0793</td>
<td>0.1053</td>
</tr>
<tr>
<td>60</td>
<td>0.0544</td>
<td>0.0220</td>
<td>0.0686</td>
<td>0.0962</td>
</tr>
<tr>
<td>70</td>
<td>0.0595</td>
<td>0.0177</td>
<td>0.0619</td>
<td>0.0919</td>
</tr>
<tr>
<td>80</td>
<td>0.0438</td>
<td>0.0072</td>
<td>0.0504</td>
<td>0.0813</td>
</tr>
<tr>
<td>90</td>
<td>0.0547</td>
<td>0.0038</td>
<td>0.0399</td>
<td>0.0772</td>
</tr>
<tr>
<td>100</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0327</td>
<td>0.0726</td>
</tr>
</tbody>
</table>

4.2. Testing samples with missing data

In the next step XOR-HAPLOGEN was tested on the same data sets with some of the data being masked as “missing”. The level of the missing data was set at 10%. The sites to be masked as missing data were randomly selected for each individual. The same 10 samples with varying percentages of full genotypes were used as for the case without missing data. To evaluate the solution the same three error rates were computed for each sample: err_{XOR}, err_{all}, swr. In addition, the prediction rate for missing sites err_{miss} was computed as a fraction of incorrectly inferred alleles at the missing data sites between the switches (used in computing the switch rate) out of the total number of missing data sites. The average values of each prediction error after running XOR-HAPLOGEN algorithm on all the samples are given in Table 2.

4.3. Discussion

Algorithm’s results consist of predicted XOR and heterogeneous sites and predicted block structure. Although attractiveness of the results depends on one’s threshold for acceptable amount of error and the availability of fully genotyped data, the overall error levels are encouraging. For example, the prediction error for XOR sites in samples without missing data (column 2 of Table 1) dropped significantly from 11.19% (at 10% full genotyping) to less than 6.55% for the samples with only 20% or more of full genotypes. As can be expected, in the samples with missing data the overall error rates are higher due to the loss of information (Table 1 and 2). Fig. 3 illustrates the average prediction errors for XOR sites (inset (a)) and the average switch rates for samples with and without missing data (inset (b)) in a comparative fashion. These graphs reveal one important detail: prediction errors for XOR sites are practically not affected by the presence of missing data for samples with 70% or less of full genotypes unlike the switch rates, which exhibit consistently lower rates for the samples without missing data. The overall difference between the switch rates across all percentages of full genotyping for samples with and without missing data is within 1.5–3% range.

Next, the stability of the block structure for samples with different percentages of included full genotypes was evaluated. Figs. 4 and 5 show the average block boundary scores calculated from 10 generated samples for each percentage of included full genotypes for groups with no missing and missing data, respectively. These figures show a significant amount of noise from the loss of information in homogeneous sites for the data sets with very low percentages of fully genotyped individuals. The noise gradually diminishes to reveal a very distinct block structure as the number of fully genotyped individuals in sample grows. This fact can be demonstrated by plotting a fraction of loci with low scores (for example, with scores less than 0.1 as shown in Fig. 6(a)) and the median scores (Fig. 6(b)) against the percentage of full genotyping. Thus, noise reduction is shown by increasing fraction of loci with low scores and decreasing score median. As can be expected, the fraction of loci with low scores is higher and the score median is lower for samples with missing data due to the additional noise created by the lost alleles.

Despite the noise the highest block boundary scores seem to be in a very good consistency even for the lowest percentages of fully genotyped individuals in a sample (when compared to the 100% baseline). For example, from Figs. 4 and 5 it is clear that, among others, the peaks of block boundary scores occurring around 30th, 47–50th, 120th, 150th and 170th positions stand out for most percentages above 0.25 threshold of full genotyping. Thus, noise reduction is shown by increasing fraction of loci with low scores and decreasing score median. As can be expected, the fraction of loci with low scores is higher and the score median is lower for samples with missing data due to the additional noise created by the lost alleles.

Fig. 3. Prediction errors for XOR sites for samples with and without the presence of 10% missing data (a), switch rates for samples with and without the presence of 10% missing data (b).
Fig. 4. Block boundaries scores for samples with different percentages of full genotypes for the CEU XOR data.

Fig. 5. Block boundaries scores for samples with different percentages of full genotypes for CEU XOR data with 10% missing sites.
between two signals. Fig. 7 shows plots of cross-correlation functions for comparing samples of length 200 with 10% and 100% of full genotyping with (inset (b)) and without (inset (a)) missing data. The cross-correlation is highest when the two signals are not shifted with respect to each other (lag 200, equal to signals' length). Similar plotting of every pair of signals within each group (missing and no missing data) exhibits exactly the same behavior of the cross-correlation function.

Therefore, it can be concluded that the block structure produced by the algorithm shows satisfactory pattern consistency even if percentage of fully genotyped individual included in a sample is low. Overall block boundary scores tend to clear up from most of the noise when the percentage of full genotypes is greater or equal than 50%.

XOR-HAPLOGEN took about 3.5 minutes to process each data set on the 2.3 GHz processor (either with or without the missing data) which was within the range of the running time of the best algorithms for regular (full-genotype) haplotyping and block partitioning [11,13,15]. The acceptable level of the prediction accuracy (which itself depends on the percentage of the fully genotyped individuals in the sample) in general should depend on the allowed cost-error trade-off. For this particular set of samples the prediction errors for XOR sites $err_{XOR}$'s decrease significantly for sample with greater than 10% of full genotypes (with or without missing data). Thus it is preferable to use more than 10% of the fully genotyped individuals in a sample. Error rates $err_{all}$ and $swr$ drop to low values and the block structure becomes stable for samples with 50% or more of full genotypes are the preferred choice for practical applications.

5. Conclusions

The proposed algorithm for XOR-haplotyping possesses several new valuable features that increase its practicality. XOR-HAPLOGEN can process large data sets (up to 350,000 entries of the genotype matrix) and, in particular, is suitable for processing long SNP sequences taking into consideration haplotype block structure. XOR-HAPLOGEN can also be applied to the data with missing values. The algorithm is designed to infer haplotype resolution and construct block structure for XOR-genotype data samples with some non-empty subset of individuals represented by full genotypes. In testing the algorithm on the CEU data, with samples including different percentages of full genotypes and also with and without presence of the 10% missing data, the algorithm was able to produce haplotype resolution with low prediction errors for XOR sites (3.25–11.19% for samples without missing data, 5.47–11.47% for samples with missing data), heterogeneous (2.1–16.96% and 3.99–19.09%, respectively) and missing data sites (7.72–17.96% for 10% missing data). The accuracy of prediction was proportional to the percentage of fully genotyped individuals in a sample. While missing data contribute in some degree to the prediction error, the prediction errors for XOR sites are not affected. The prediction of XOR sites is accurate for samples with more than 10% of fully genotyped individuals. XOR-HAPLOGEN may be the first algorithm that allows processing of long-range SNP sequences with flexible amount of XOR-genotypes and possibility of missing data. Due to this fact the comparison of the results to other algorithms was not possible.

The proposed algorithm XOR-HAPLOGEN also produces the block structure which shows consistency for different percentages
of included full genotypes. The block structure is stable for samples with 50% and higher percentages of full genotypes (with and without missing data).

As a final note, XOR-HAPLOGEN showed very competitive computational speed: it was as fast as the best algorithms for haplotyping and block partitioning [11,13–15] designed for full genotypes when tested on the same data.

The fact that some individuals may contribute more information than the others and thus facilitate a significant decrease of the error rates leads the prospects for future research. The future work should include investigation of the ways to perform controlled selection of the fully genotyped individuals in order to increase the overall effectiveness of the proposed method. Although it was shown that for 10% of the missing data the prediction error for XOR sites is not significantly affected, the extent of the effect of the missing data on the accuracy of prediction of XOR sites may also be need to be studied.

Web resources


References