Computational Problems in Perfect Phylogeny Haplotyping: Xor-Genotypes and Tag SNPs

Tamar Barzuza[1], Jacques S. Beckmann[2,3], Ron Shamir[4] and Itsik Pe'er[5]

[1]Dept. of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel.
tamar.barzuza@weizmann.ac.il
[2]Dept. of Molecular Genetics, Weizmann Institute of Science, Rehovot 76100, Israel.
jacqui.beckmann@weizmann.ac.il
[3]Département de Génétique Médicale Ch-1011 Lausanne. Switzerland, jacques.beckmann@hospvd.ch
[4]School of Computer Science, Tel- Aviv University, Tel Aviv 69978, Israel. rshamir@post.tau.ac.il
[5]Medical and Population Genetics Group, Broad Institute, Cambridge MA 02142 US. peer@broad.mit.edu

Abstract

The perfect phylogeny model for haplotype evolution has been successfully applied to haplotype resolution from genotype data. In this study we explore the application of the perfect phylogeny model to other problems in the design and analysis of genetic studies. We consider a novel type of data, xor-genotypes, which distinguish heterozygote from homozygote sites but do not identify the homozygote alleles. We show how to resolve xor-genotypes under perfect phylogeny model, and study the degrees of freedom in such resolutions. Interestingly, given xor-genotypes that produce a single possible resolution, we show that the full genotype of at most three individuals suffice in order to determine all haplotypes across the phylogeny. Our experiments with xor-genotyping data indicate that the approach requires a number of individuals only slightly larger than full genotyping, at a potentially reduced typing cost.

We also consider selection of minimum-cost sets of tag SNPs, i.e., polymorphisms whose alleles suffice to recover the haplotype diversity. We show that this problem lends itself to divide-and-conquer linear-time solution. Finally, we study genotype tags, i.e., genotype calls that suffice to recover the alleles of all other SNPs. Since most genetic studies are genotype-based, such tags are more relevant in such studies than the haplotype tags. We show that under the perfect phylogeny model a SNP subset of haplotype tags, as it is usually defined, tags the haplotypes by genotype calls as well.

Keywords: SNPs, Haplotypes, Perfect Phylogeny, Tag SNPs, Graph Realization

1. Introduction

1.1. Background

Genetic information in nature is usually stored as a linear sequence, written in a molecular DNA alphabet of four letters (nucleotides), A, C, G and T. Higher organisms are *diploid*, i.e., have two near-identical copies of their genetic material arranged in paired molecules called *chromosomes*, one originating from each parent. Such chromosomes are *homologous*, that is, contain essentially the same genes in altered variants. Changes between variants comprise mainly of *Single Nucleotide Polymorphisms* (SNPs), i.e., sequence sites where one of two letters may appear [1]. These SNPs are numerous and it is estimated that any two homologous human chromosomes sampled at random from the population differ on average once in every thousand letters, accounting thus for a few million such differences along the entire genome. The variants of a SNP are called *alleles*. An individual is said to be *homozygous for a SNP* if both homologous chromosomes bear the same allele for this SNP and *heterozygous* otherwise. The sequence of alleles along a chromosome is called a *haplotype*. At first approximation a chromosome can be considered as a patchwork of haplotypes along its length. A *genotype* along homologous chromosomes lists the conflated (unordered pair of) alleles for each SNP (see Fig. 1).

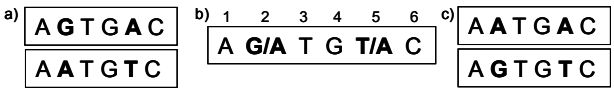


Fig. 1. An example of 6 SNPs along two homologous chromosomes of an individual. (a) This individual's haplotypes. (b) This individual's genotype. Here the Xor-genotype (set of heterozygous SNPs) would be {2,5}. (c) Another potential haplotype pair giving rise to the same genotype.

Both genotype and haplotype data are used in genetic studies. Haplotypes are often more informative [6]. Unfortunately, current experimental methods for haplotype determination are technically complicated and cost prohibitive. In contrast, a variety of current technologies offer practical tools for genotype determination [26]. Given genotype data, the haplotypes must be inferred computationally, in a process called *resolving*, *phasing* or *haplotyping* [7,8,9,10,11,27]. A single genotype may be resolved by different, equally-plausible haplotype pairs (see Fig. 1), but the joint inference of a set of genotypes may favor one haplotype pair over the others for each individual. Such inference is usually based on a model for the data. Informally, most models rely on the observed phenomenon that over relatively short genomic regions, different human genotypes tend to share the same small set of haplotypes [2,3].

1.2. The Perfect Phylogeny Model

During sexual reproduction, only one homologous copy of each chromosome is transmitted to the offspring. Moreover, that copy has alternating segments from the two homologous chromosomes of the parent, due to a segmental exchange process called (*meiotic*) *recombination*. Studies have shown that recombination occurs mainly in narrow regions called *hotspots*. The genomic segments between hotspots are called *blocks*. They show essentially no recombination [4] and their haplotype diversity is limited [2,3]. Within blocks, haplotypes evolve by mutations, i.e., replacement of one nucleotide by another at particular sites (other mutation types are not discussed here). Since mutations are relatively rare [5], it is often assumed, as we do here, that at most one mutation occurs in each site. The *perfect phylogeny model* for haplotype block evolution assumes that all haplotypes in the population have a common ancestor, no recombination and no recurrent mutation.

The *Perfect Phylogeny Haplotyping* problem (PPH) seeks to infer haplotypes that satisfy the perfect phylogeny model (we defer formal definitions to Section 1.5). PPH was first introduced by Gusfield [9], who presented an almost linear solution by reducing PPH to the classical *Graph Realization* problem. Simpler, direct solutions were later given [10,11], which take $O(nm^2)$ for n haplotypes and m SNPs.

1.3. Informative SNPs

Many medical genetics studies first determine the haplotypes for a set of individuals and then use these results to efficiently type a larger population. Having identified the restricted set of possible haplotypes for a region, the identity of a subset of the SNPs in the region may suffice to determine the complete haplotype

of an individual. Such SNPs are called *tag SNPs*, and typing them alone would lose no information on the haplotypes. More generally, we may be interested only in a subset S of all SNPs (e.g., coding and regulatory SNPs only) but can use all available SNPs to tag them. In this setting we call S the set of *interesting SNPs*, and seek a smallest possible *informative SNP set*, i.e., is a subset of all SNPs that captures all the information on S (see Fig. 2). Hence, the identification of few informative SNPs may lead to substantial saving in typing costs. For this reason, the computational problems of finding a minimal tag (or informative) set have been studied [2,13,18].

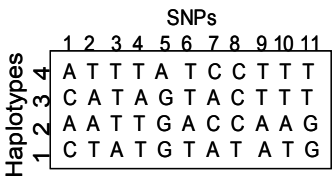


Fig. 2. Tag SNPs and informative SNPs. The set {1,2} is a tag SNP set. If {9,10,11} is the interesting SNP set, then the interesting set distinguishes the haplotypes 1, 2 and {3,4}, but does not distinguish between haplotypes 3 and 4. Therefore {1,2} and {6,8} are both informative SNPs sets but {4,5} and {2,3} are not.
Notice that the same genotype **A/C T/A** is obtained for the tag SNP set {1,2} from the two pairs of haplotypes {1,2} and {3,4}.

Finding the minimum set of tag SNPs within an unconstrained block is NP-hard [12]. When the perfect phylogeny model is assumed, in the special case of a single interesting SNP, a minimal set of informative SNPs was shown to be detectable in $O(nm)$ time, for n haplotypes and m SNPs [13].

1. 4 Contribution of this work

We study here several problems arising under the perfect phylogeny model during genetic analysis of a region, along the process from haplotype determination toward their utilization in a genetic study. Our analysis focuses on a single block.

Some experimental methods such as DHPLC [14] can determine whether an individual is homozygous or heterozygous for each SNP, but cannot distinguish between the two homozygous sites. Typing SNPs in such manner will provide, for each individual, a list of the heterozygous sites, which we refer to as the individual's *xor-genotype*. Xor-genotypes are less informative than regular ("full") genotypes; but their generation may be less costly. Therefore, it is of interest to infer the haplotypes based primarily on xor-genotypes instead of full genotypes. In Section 2 we introduce the *Xor Perfect Phylogeny Haplotyping* problem (XPPH), study the limitations of using only xor-genotypes, and the additional genotype information required. Section 2.2 presents an efficient solution to XPPH based on the graph realization problem [15]. We implemented our solution and evaluated the XPPH strategy in Section 2.3. Our tests show that the method compares favorably with standard genotyping.

Section 3 studies informative SNPs under the perfect phylogeny model. We generalize the minimum informative set (and tag set) problems by introducing a cost function for SNPs, and seek minimum cost sets. The cost is motivated by the facts that typing some SNPs may be technically harder (e.g., those in repetitive or high GC regions), and that some SNPs are more attractive for direct typing (e.g., protein-coding SNPs, due to prior assumptions on their functionality). In section 3.2 we find minimal cost informative SNP sets in $O(m)$ time for any number of interesting SNPs, when the perfect phylogeny tree is given. This generalizes the result of [13]. Section 3.3 discusses a practical variant of the tag SNPs set, i.e., the phasing tag SNPs set: As we usually have only genotypic (conflated) information on the SNPs, a practical goal would be to find a set of SNPs that give the same information as tag SNPs, but instead of knowing their haplotype we only know their genotype. We prove that the set of informative SNPs is guaranteed to have this quality, and that this is guaranteed only under the perfect phylogeny model.

We conclude with a discussion in Section 4. Throughout the manuscript, many proofs are omitted, due to lack of space.

1.5. Preliminaries

We denote the two alleles for each SNP by 0 and 1. A haplotype is represented by a binary vector. A set of haplotypes is represented by a binary matrix H, where each row is a haplotype vector and each column is the vector of SNP alleles. We denote the allele of haplotype i for SNP j by H_{ij} or by h_j for the haplotype $h=H_i$. A *genotype* is the conflation (mixing) of two haplotypes. For example, the pair of haplotypes 00100 and 10001 gives rise to the genotype {0,1}{0,0}{0,1}{0,0}{0,1}.

The perfect phylogeny model is formalized as follows: Let $H_{n\times m}$ be a binary matrix of n distinct haplotypes over m SNPs. A *perfect phylogeny* for H is a pair (T,f) where $T=(V,E)$ is a tree with $\{1,\ldots,n\}\subseteq V$ and $f:\{1,\ldots,m\}\rightarrow E$ is an assignment of SNPs to edges such that (1) every edge of T is labeled at least once and (2) for any two rows k, l, $H_{kj}\neq H_{lj}$ iff the edge $f(j)$ lies on the unique path from node k to node l in T. The analysis of this model is heavily based on a fundamental property (cf. [21]):

Theorem 1: There is a perfect phylogeny for H iff H does not contain a 4×2 submatrix in which all four rows are different. Such a submatrix is called a *four-gamete*.

2. Xor Haplotyping

In this section we formally define the problem of Xor Perfect Phylogeny Haplotyping, provide an algorithm for the problem and discuss how to handle ambiguity in the data. We then show how to obtain the actual haplotypes efficiently using a small amount of additional full genotypes.

2. 1. Problem definition

Definition: A *xor-genotype* of a haplotype pair $\{h,h'\}$ is the set of their heterozygote SNPs, i.e., $\{s|h_s \neq h'_s\}$ (see Fig. 1). A set of haplotypes H *explains* a set of xor-genotypes X if each member of X is a xor-genotype of a pair of haplotypes in H.

Problem 1: Xor Perfect Phylogeny Haplotyping (XPPH)

Input: A set $X=\{X_1,\ldots,X_n\}$ of xor-genotypes over SNPs $S=\{s_1,\ldots,s_m\}$, such that $X_1 \cup \ldots \cup X_n = S$.

Goal: Find a haplotype matrix H with a perfect phylogeny (T,f), such that H explains X, or determine that no solution exists. (See Fig. 3)

Hereafter, we shall omit the term "or determine that no solution exists" from problem statements for brevity. This requirement is part of the goal of all the algorithms in this study.

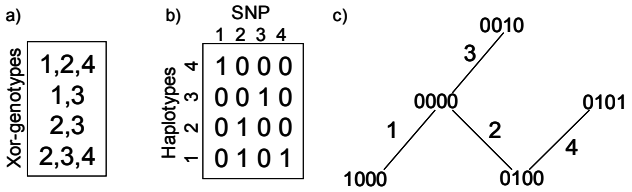


Fig. 3. An example of a solution to XPPH. (a) X -The four xor-genotypes. (b) H - The inferred haplotypes. (c) (T,f) - A perfect phylogeny for H. Notice that H explains X by taking the haplotype pairs {1,4},{3,4},{2,3} and {1,3}. Note that T includes a haplotype (0000) that is not in H.

2.2. An algorithm for XPPH

2.2.1. Inference up to bit flipping

A first step in understanding XPPH is the observation that the solution is never unique. Rather, flipping the alleles of SNPs in a solution yields yet another solution, as we now explain.

Definition: Two haplotype matrices $H_{n\times m}$ and $H'_{n\times m}$ are *equivalent up to bit flipping* (denoted $H \leftrightarrow H'$) if for any two rows k, l, $H_{kj} \neq H_{lj} \Leftrightarrow H'_{kj} \neq H'_{lj}$. $H \leftrightarrow H'$ iff one can be obtained from the other by exchanging the roles of 1 and 0 for some columns. Notice that $\leftrightarrow$ is a set-theoretic equivalence relation.

Observation 1: If $H \leftrightarrow H'$ then X can be explained by H iff it can be explained by H'.

Observation 1 implies that XPPH can only be solved up to bit flipping based on the data given by X.

In some cases, however, there may be several alternative sets of haplotypes that explain X and are not $\leftrightarrow$-equivalent. In that case, we will not be able to determine which of those sets is the one that really gave rise to X. Our only achievable goal is therefore to identify when the solution obtained is guaranteed to be the correct one. We will next show that this is guaranteed by the uniqueness of the solution. The analysis relies on the following property of perfect phylogenies:

Key property: Let (T,f) be a perfect phylogeny for H. If $H_{ij}=0$ then for all k, $H_{kj}=0$ iff nodes i and k are in the same component of $T \backslash f(j)$.

Definition: (T,f) is a *perfect phylogeny for X* if (T,f) is a perfect phylogeny for some H that explains X.

Proposition 1: When X has a unique perfect phylogeny then the haplotype matrix that explains it is unique up to bit flipping (i.e., up to $\leftrightarrow$-equivalence).

Proof: It suffices to prove that if (T,f) is a perfect phylogeny for H then there is no H' such that (T,f) is a perfect phylogeny for H' and $\neg(H \leftrightarrow H')$. First we observe that there is a unique correspondence between

the nodes of T and the haplotypes in H. This correspondence is obtained as follows. We first identify the haplotype $h \in H$ of an arbitrary leaf v. This is done by observing the SNPs that correspond to the edge incident on v. h is the only haplotype that is distinct from all others in these SNPs. The haplotypes of all other nodes are now readily determined by the key property. This generates the uniuqe correspondence. The actual haplotypes are set by fixing arbitrarily the haplotype vector of one node and setting all others according to the key property. □

Proposition 1 motivates a new formulation of Problem 1':

Problem 1′: XPPH:

Input: A set $X=\{X_1,\ldots,X_n\}$ of xor-genotypes over SNPs $S=\{s_1,\ldots,s_m\}$, such that $X_1 \cup \ldots \cup X_n = S$.

Goal: Find a unique perfect phylogeny (T,f) for X, or determine that there are multiple perfect phylogenies for X.

Proposition 1 implies that a solution to Problem 1′ (if unique) is guaranteed to be a perfect phylogeny for the correct set of haplotypes, i.e., the haplotypes that actually gave rise to X.

Gusfield's work [9] leads to an interesting and useful connection between xor-genotypes and paths along the perfect phylogeny tree, as follows.

Definition: We say that a pair (T,f) *realizes* $X_i \subseteq S$ if X_i is the union of edge labels that constitute a path in T. (T,f) is said to *realize a collection* X of subsets if each $X_i \in X$ is realized by (T,f).

Proposition 2: (T,f) is a perfect phylogeny for X iff X is realized by (T,f).

The following formulation for XPPH is finally obtained:

Problem 1″: XPPH:

Input: A set $X=\{X_1,\ldots,X_n\}$ of xor-genotypes over SNPs $S=\{s_1,\ldots,s_m\}$, such that $X_1 \cup \ldots \cup X_n = S$.

Goal: Find the unique realization (T,f) for X, or determine that there are multiple realizations for X.

Intuitively, the larger the number of typed SNPs, the greater the chances to have a unique realization. Occasionally, however, a dataset X may have multiple realizations even with many SNPS.. This is the case of the data including xor-equivalent SNPs:

Definition: We say that $s,s' \in S$ are *xor-equivalent w.r.t. X* and write $s \approx^X s'$ if for all i: $s \in X_i \Leftrightarrow s' \in X_i$.

Fortunately, xor-equivalent SNPs may be redundant. This is the case of the data including haplotype-equivalent SNPs:

Definition: We say that $s,s' \in S$ are *haplotype-equivalent w.r.t. H* and write $s \approx^H s'$ if for all i,j: $H_{is} \neq H_{js} \Leftrightarrow H_{is'} \neq H_{js'}$. Note that $\approx^H$ and $\approx^X$ are equivalence relations.

Observation 3: Haplotype-equivalence implies xor-equivalence but not vice versa. (See Fig 4).

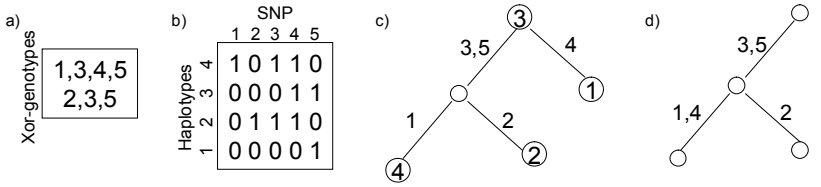


Fig. 4. Xor-equivalence and haplotype-equivalence. (a) X – The xor-genotypes. (b) H – The haplotypes matrix. Haplotypes 1 and 4 form the first xor-genotype, and haplotypes 2 and 3 form the second. The pairs of xor-equivalent SNPs are {1, 4} and {3, 5}, while only 3 and 5 are haplotype-equivalent. (c) (T,f) – A realization for X that is a perfect phylogeny for H. (d) (T',f') – Another realization for X that is not a perfect phylogeny for H.

We next show that haplotype-equivalent SNPs are redundant.

Notation: Denote by $S^H \subseteq S$ the set that is obtained by taking one representative from each haplotype-equivalence class. Denote by H^H the haplotype matrix that is obtained by restricting H to S^H.

Observation 4: (1) To obtain a perfect phylogeny (T,f) for H, one can obtain a perfect phylogeny (T,f') for H^H and then set $f(s)=f'(s^H)$ for every $s \in S$ that is haplotype-equivalent to s^H. (2) (T,f') is a unique perfect phylogeny for H^H since S^H contains no haplotype-equivalent SNPs.

Observation 4 implies that haplotype-equivalent SNPs are redundant, hence may be merged to label a single edge in (T,f) (See Fig 4c); and by doing so, we discard the degrees of freedom that are due to haplotype-equivalent SNPs.

However, identifying haplotype-equivalent SNPs is not trivial when we only have xor-genotype information, which as Observation 3 implies may not suffice. In other words, the closest we can get to merging haplotype-equivalent SNPs is merging the xor-equivalent SNPs, which by Observation 3 may lead to information loss (See Fig 4d).

Definition: Denote by $S^X \subseteq S$ the set that is obtained by taking one representative from each xor-equivalence class. Denote by X^X the xor-genotypes that are obtained by restricting X to S^X. X^X is called the *canonic version* of X.

We show next that when the canonic version of X has a unique realization, then there was no information loss in merging xor-equivalent SNPs, since xor-equivalence implies haplotype-equivalence in this particular case.

Theorem 2: Let (T,f') be a unique realization for X^X. Extent the mapping f' to S by setting $f(s)=f'(s^X)$ for every s that is xor-equivalent to s^X. Then (T,f) is a perfect phylogeny for the correct haplotype matrix that gave rise to X.

Proof: By Proposition 2, (T,f') is a unique perfect phylogeny for X^X, and by Proposition 1 it is a perfect phylogeny for the correct haplotype matrix on S^X. We will next show that in the special case where (T,f') is unique, xor-equivalence implies haplotype-equivalence for the data set X. Then, by Observation 4, (T,f) is a perfect phylogeny for the correct haplotype matrix that gave rise to X. Suppose to the contrary that SNPs $s_1,s_2 \in S$ are xor-equivalent but not haplotype equivalent. Consider the unique perfect phylogeny (T^S,f^S) of H^H. Since s_1 and s_2 are not haplotype-equivalent they label distinct edges, e_1 and e_2 respectively, in T^S. Notice that $f^{-1}(e_1) \cup f^{-1}(e_2)$ are xor-equivalent. Let (T^S_1,f^S_1) be obtained from (T^S,f^S) by contracting e_1 (identifying e_1's nodes), and by taking $f^S_1(s)=e_2$ for $s \in f^{-1}(e_1)$. (T^S_2,f^S_2) is similarly obtained from (T^S,f^S) by contracting e_2. Then both (T^S_1,f^S_1) and (T^S_2,f^S_2) realize X^X, and $(T^S_1,f^S_1) \neq (T^S_2,f^S_2)$; in contradiction to the uniqueness of (T,f'). □

The formulation of Problem 1″ leads to a connection between XPPH and the graph realization problem:

Problem 2: The Graph Realization Problem (GR)

Input: A collection $P=\{P_j\}$ of subsets, $P_1,\ldots,P_n \subseteq S$.

Goal: Find a pair (T,f) that realizes P.

Observation 2: Problem 1″ is now exactly the graph realization problem (when restricting the solution to GR to be unique).

The graph realization problem was first defined in matroid theory by Tutte [16], who proposed an algorithm of $O(mn^2)$ time, where $|P|=m$ and $|S|=n$. Gavril and Tamari [17] subsequently solved it in time $O(m^2n)$. Later, Bixby and Wagner [15] presented an $O(\alpha(m,n)mn)$ time algorithm, ($\alpha(m,n)$ is the inverse Ackermann function, $\alpha(m,n) \leq 4$ for all practical values of m,n). All three algorithms required linear space. These algorithms determine the existence of a graph realization and also the uniqueness of such a solution, hence they can be applied to solve XPPH.

The above discussion implies that the following procedure solves XPPH: Let M be the incidence matrix of X and S, i.e., $M_{ij}=1$ iff $s_j \in X_i$. Find S^X and X^X. (This can be done by a radix-sort of the columns of M in $O(nm)$ bitwise operations.) Then solve the graph realization problem on X^X. If the solution is unique it implies a perfect phylogeny for X.

In case that the xor-genotypes data cannot be augmented and there are several solutions to the GR problem, we may wish to choose one of them as a perfect phylogeny for X. Additional considerations may help in the choice [9]. We have developed a method for efficiently representing all the solutions for the graph realization problem by extending the algorithm in [17]. This representation is intuitive and implementation is straightforward. Details are omitted in this abstract.

2.2.2. Assigning actual haplotypes

In the previous section we concluded that even when XPPH has a single solution, the assignment of haplotypes to the tree nodes can be done only up to bit flipping. In order to obtain a concrete assignment, the input data must be augmented by additional genotyping of a selected set of individuals. We will prove that it suffices to fully genotype at most three individuals, and show how to select them. First, we explain how the additional genotype data are used to resolve the haplotypes. Denote by G_i the genotype of individual i (whose xor-genotype is X_i). Hereafter, we consider only those individuals with $X_i \neq \emptyset$.

Problem 3: Haplotyping on the Tree
Input: (a) A collection of non-empty xor-genotypes X; (b) a perfect phylogeny (T,f) for X, which is unique up to haplotype-equivalent SNPs; and (c) complete genotypes of the individuals $\{i_1,\ldots,i_p\}$.
Goal: Infer the haplotypes of all the individuals.

Haplotyping across the tree is based on the above key property, which determines the alleles of a SNP j for all haplotypes, based on its allele in some particular node. More specifically, all those alleles are determined given a genotype G_i, homozygote for SNP j, whose haplotypes correspond to identifiable nodes in T. Consequently, G_i resolves the bit-flip degree of freedom for each SNP $s \in S \setminus X_i$. Hence:

Proposition 3: The haplotypes can be completely inferred by $G_1,\ldots,G_p$ iff $X_1 \cap \ldots \cap X_p = \emptyset$.

The proposition brings about a criterion by which individuals should be selected for full genotyping. It motivates the following set-selection problem:
Problem 4: Minimum Tree Intersection (MTI)
Input: A collection of sets $X=\{X_1,\ldots,X_n\}$ and a perfect phylogeny (T,f) for X.
Goal: Find a minimum subset of X whose intersection is empty.
Note that the prefect phylogeny condition here is crucial: Without the condition that each X_i is a path in the tree, the problem is equivalent to the NP-hard set-cover problem.

Theorem 3: If $X_1 \cap \ldots \cap X_n = \emptyset$ then there is a minimum tree intersection set of size at most 3.
Proof: Consider the path X_1, and w.l.o.g. label the SNPs according to their order along that path as $(1,\ldots,k)$. For each i, the set $X_1 \cap X_i$ defines an interval in that order. If $X_1 \cap X_i = \emptyset$ for some i then $\{X_1, X_i\}$ are a solution. Otherwise all intervals overlap X_1. Denote these intervals by $[l_j, r_j]$ for $j=2,\ldots,n$. Take the interval that ends first and the interval that begins last, i.e., $L=\text{argmin}_j(r_j)$ and $R=\text{argmax}_j(l_j)$. Since $X_1 \cap \ldots \cap X_n = \emptyset$ then $[l_2,r_2] \cap \ldots \cap [l_n,r_n] = \emptyset$, hence it follows that $[l_R,r_R] \cap [l_L,r_L] = \emptyset$. We get $(X_1 \cap X_L \cap X_R) = \emptyset$. □

In case no SNP is present in all X_i-s, the above proof provides an algorithm for finding three individuals whose full genotypes solve MTI. A slight modification allows finding two individuals instead of three when possible. The time complexity is $O(nm)$. Let $Y = X_1 \cap \ldots \cap X_n$.

Corollary 1: There are at most three individuals whose genotypes can resolve all the haplotypes on the SNP set $S \setminus Y$, and they can be found in $O(nm)$ time.

In case $Y \neq \emptyset$, the SNPs in Y can be inferred up to bit flipping.

2.3 Experimental Results

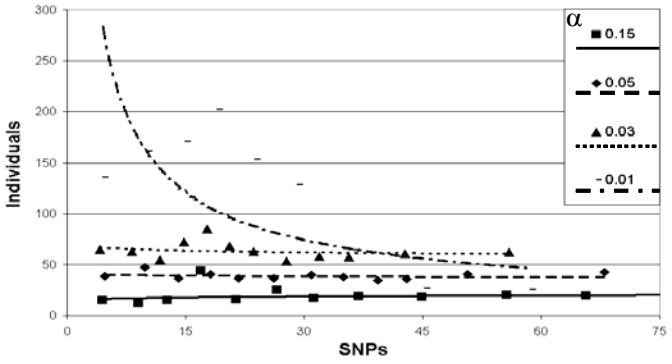
We implemented Gavril and Tamari's algorithm for Graph Realization [17]. Although it is not the asymptotically fastest algorithm available, it is simpler to implement and modify than [15]. Moreover, as the block size is usually bounded in practice by $m<30$, the quadratic dependence of the algorithm on m is not a handicap. Our implementation, GREAL, was written in C++, and is available at http://www.cs.tau.ac.il/~rshamir/greal. Another implementation due to Chung and Gusfield has recently been announced [19].

We used a standard population genetics simulator due to Hudson [22] to generate data samples under the perfect phylogeny model. In each run we generated $c=2400$ chromosomes with a prescribed number of SNPs, preserving the default values for all other simulation parameters. An important parameter in the experiments was the *minor allele frequency cutoff*, denoted by α: For a given value of α, we only used SNPs whose less frequent allele occurred in $\geq \alpha c$ chromosomes. The resulting haplotypes were randomly paired to generate xor-genotypes of individuals.

How many individuals are required to get a single solution?

We evaluated this measure by randomly adding individuals one by one and reapplying GR till the solution is unambiguous. The results (Fig. 5) show that for $\alpha \geq 0.03$, the number of individuals required to obtain a single solution is roughly an α-dependent constant, irrespective of the number of SNPs, and is practically bounded by 70. When rare alleles ($\alpha=0.01$) are present, the behavior is less predictable and the variance is very large. However, comprehensive sampling of the haplotypes is usually not achieved when rare alleles are present; fortunately, performance is satisfactory above the accepted α cutoff of 0.05.

Fig. 5. Conditions for uniqueness of the solution. The plots show the number of xor-genotypes (y-axis) needed for obtaining a single solution for a given number of SNPs (x-axis). Different lines (or least squares curves) correspond to different thresholds on the minor allele frequency cutoff . Note that the interpolated curve for =0.01 is an extremely rough estimate.



XPPH vs. PPH

Since xor-genotypes contain less information, they may have a potential economic advantage over full genotypes. However, the number of individuals required for obtaining the haplotypes is larger. We compared the number of individuals needed by XPPH and by PPH. Chung and Gusfield [23] evaluated experimentally the number of individuals required for obtaining a unique solution to PPH. We computed the same statistic for XPPH (Fig. 6a). For 50 SNPs, 50 xor-genotypes guarantee ~90% chance of uniqueness, and increasing the number of individuals has only a minor effect. Essentially the same results hold for 100 SNPs. In comparison to [23], the chances for a unique XPPH solution with > 50 xor-genotypes is only a few percent lower than for PPH data with the same number of full genotypes.

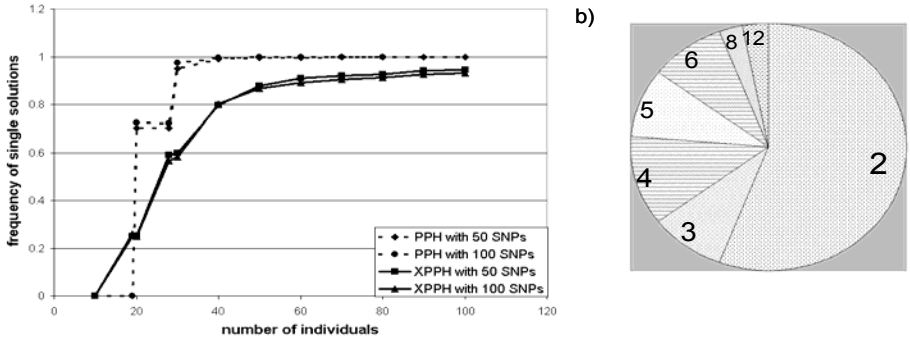


Fig. 6. The chance for a unique solution. (a) The frequency of a unique solution (y-axis) versus the number of individuals tested (x-axis). XPPH statistics are based on 5000 runs for 50 or 100 SNPs after filtering with α=0.05. PPH statistics from [23] are plotted for comparison. (b) The distribution of the number of non-unique solutions in deep coverage studies. Statistics were collected for 35 configurations of the number of SNPs (100-2000) and the number of individuals, which was at least 10 times the number of SNP equivalence classes. (α=0.05).

How high is the multiplicity of non-unique solutions?

We further focused on outlier ambiguous datasets, i.e., those that are ambiguous despite the examination of many individuals. For such datasets, the number of possible solutions is of much practical interest: If this number is limited, each solution may be tested separately. Indeed, the results (Fig. 6b) show that in this situation, when the solution is not unique, there are only a handful of solutions, usually only 2. Note that we assume equivalence of $\approx^{H}$ and $\approx^{X}$ for outlier datasets, which we confirmed for the datasets used here.

3. Informative SNPs

3.1. Problem definition

In this section we study informative SNPs under the perfect phylogeny model. We begin by introducing some terminology, concordant with [13].

Definition: Let $H=\{H_1,\ldots,H_n\}$ be a set of haplotypes over a SNP set $S=\{s_1,\ldots,s_m\}$. Let $S''\subseteq S$ be a given subset of interesting SNPs. The set $S'\subseteq S\backslash S''$ is *informative* on H w.r.t. S'' if for each $1\le k,l\le n$, whenever there is a SNP $s''\in S''$ for which $H_{ks''}\neq H_{ls''}$, there is a SNP $s'\in S'$ for which $H_{ks'}\neq H_{ls'}$.

Note that that assumption that the informative and interesting SNPs are disjoint is made without loss of generality, since we can duplicate interesting SNPs as part of the candidates for the informative set. We generalize the Minimum Informative SNPs problem [13] by introducing a cost function, as follows:

Problem 5: Minimum-Cost Informative SNPs (MCIS):

Input: (a) A set of haplotypes $H=\{H_1,\ldots,H_n\}$ over a SNP set $S=\{s_1,\ldots,s_m\}$ along with a perfect phylogeny (T,f) for H.

(b) A set of interesting SNPs $S''\subseteq S$.

(c) A cost function $C:S\rightarrow R^+$.

Goal: Find a set $S'\subseteq S\backslash S''$ of minimum total cost that is informative w.r.t. S''.

(T,f) may already be known if H was found by solving XPPH. Alternatively, it can be computed in $O(mn)$ time from haplotypes [21].

A common task which is related to picking an informative SNP set is to describe all of the haplotype variation in the region [20]. Formally, we seek a *tag set* $S'\subseteq S$ *s.t.* for each $1\le l,k\le n$, there is $t\in S'$ for which $H_{kt}\neq H_{lt}$. In order to find tag SNPs of minimum cost, one could duplicate the SNP set S and define one of the copies as interesting. A solution to MCIS on the duplicated instance is a tag SNP set of minimal cost. Hence we shall focus on the more general MCIS problem.

3.2 An algorithm for MCIS

3.2.1 Problem decomposition

Recall that if $T=(V,E)$ is a perfect phylogeny for $H_{n\times m}$ then $\{1,\ldots,n\}\subseteq V$, i.e., the haplotypes of H label nodes in the perfect phylogeny. If a node of T is labeled by a haplotype from H we say it is *observed*. Otherwise we say it is *ancestral*. Ancestral nodes represent haplotypes that have been intermediate stages in evolution but did not survive to the present, or were not collected in the sample. It is easy to see that the leaves of T are always observed. The observed internal nodes in T can be used for a decomposition of T as follows:

Definition: An *ancestral component* is a subtree of T in which all the internal nodes are ancestral and all the leaves are observed.

Since the leaves of T are observed, T can be represented as a union of edge-disjoint ancestral components, where each union step merges two components by identifying copies of the same observed node. Two different components can share at most one observed node, but do not share ancestral node. Partitioning T into ancestral components is straightforward. We now show that in order to find informative SNPs we can divide the tree into ancestral components and find informative SNPs for each single component separately. The subproblem on a component is defined as follows: Denote an instance of MCIS by the input tuple $I=(H,S,C,T,f,S'')$. Let $T_1,\ldots,T_p$ be T's ancestral components where $T_i=(V_i,E_i)$. Denote by $S_i\subseteq S$ the SNPs that label E_i. The input tuple for T_i is $I_i=(H_i,S_i,C_i,T_i,f_i,S_i'')$ where the sets and functions are the restriction of the original sets and functions to S_i,

Theorem 4: Suppose for every i, $IS(I_i)$ solves I_i. Then $IS(I)=IS(I_1)\cup\ldots\cup IS(I_p)$ solves I.

Proof: We shall show that $IS(I)$ is informative w.r.t. S'' iff $IS(I_i)$ is informative w.r.t. S_i'' for all i; The theorem then will follow by the additivity of the cost function. If haplotypes k,l belong to the same observed component T_i, and there is a SNP s such that $H_{ks}\neq H_{ls}$, then by the key property it must be that $s\in S_i$. Therefore, the informativeness of $IS(I)$ implies the informativeness of $IS(I_i)$ for all i. For the opposite direction, suppose there are $t\in S''$ and $1\le l,k\le n$ such that $H_{kt}\neq H_{lt}$. Let T_i be the subtree which contains the edge with label t (i.e., $t\in S_i$). Then by the key property, there are l',k' in T_i such that $H_{k't}\neq H_{l't}$, where l',k' are the observed nodes of T_i that are on the path from k to l in T. But then there is $s'\in IS(I_i)\subseteq IS(I)$ such that $H_{k's}\neq H_{l's}$. Hence, by the key property, $H_{ks}\neq H_{ls}$.□

3.2.2 Solving MCIS on an ancestral component

In this section we solve MCIS restricted to a single ancestral component. We first reformulate it in terms of the tree edges, and then show how to solve it. We introduce the following notations: Edges labeled by

interesting SNPs are called *target edges*. The set of target edges is $\tau=\{e|f^{-1}(e)\cap S''\neq\emptyset\}$. It specifies the interesting information in terms of tree edges. An edge is *allowed* if it is labeled by some non-interesting SNP. The set of allowed edges is $\alpha=\{e|f^{-1}(e)\cap(S\backslash S'')\neq\emptyset\}$. These are the edge-analogs of potentially informative SNPs. Edges in $\tau\backslash\alpha$ are called *forbidden*. Forbidden edges cannot be used as informative, but edges in $\tau\cap\alpha$ can.

We now expand the definition of the cost function to edges: The *cost of an edge e*, denoted $C(e)$, is the minimum cost of a non-interesting SNP that labels *e*. For $e\in\tau\backslash\alpha$ define $C(e)=\infty$. This allows us to provide an equivalent formulation for MCIS:

Problem 6: Minimum Cost Separating Set (MCSS)
Input: The same input as for MCIS.
Goal: Find $E'\subseteq E$ of minimum cost, such that in $G=(V,E\backslash E')$ there are no two observed nodes that are connected by a path containing a target edge.

Proposition 4: MCIS and MCSS are equivalent.
Proof: It suffices to show that an informative set for H w.r.t. S'' separates those observed nodes that are connected by a path containing edges from τ, and vice versa. Observed nodes of T, v_1 and v_2, have corresponding haplotypes of H, H_k and H_l, and vice versa. But then by the key property $H_{ks}\neq H_{ls}$ iff s labels an edge on the path from v_1 to v_2. □

We are now ready to outline a dynamic programming algorithm for MCSS. W.l.o.g. assume $|V|>2$. Take some internal node $r\in V$ and root T at r. For $v\in V$ denote by $T_v=(V_v,E_v)$ the subtree of T that is rooted at v. For a solution $S_v\subseteq E_v$ of the induced sub instance $I(T_v)$, denote by R_v the connected component which contains v in $G_v=(V_v,E_v\backslash S_v)$. The algorithm will scan T from the leaves up and at each node v form an optimal solution for the subtree T_v based on the optimal solutions for the subtrees of its children. When combining such children solutions, we have to take into consideration the possibility that the combination will generate new paths between observed haplotypes, with or without target edges on them. To do this, we distinguish three types of solutions: S_v is called *empty* if there are no observed haplotypes in R_v. It is called *connected* if some observed haplotypes in R_v are connected to v via target edges. S_v is called *disconnected* otherwise, i.e., if there are observed haplotypes in R_v but there is no path connecting an observed haplotype to v via target edges. Let N_v, P_v and A_v denote the respective best empty, connected, or disconnected solutions. We define recursive formulae for their costs as follows:

- For a leaf node $v\in V$ we initialize: $C(N_v)=\infty$, $C(P_v)=\infty$, $C(A_v)=0$.
- For an internal node $v\in V$ with children $\{u_1,\ldots,u_{k(v)}\}$ we write:

$$Tear(i)=\min\left\{C\left(N_{u_i}\right),C\left(P_{u_i}\right)+C\left(v,u_i\right),C\left(A_{u_i}\right)+C\left(v,u_i\right)\right\} \quad (1)$$

$$C\left(N_v\right)=\sum_{i=1}^{k(v)}Tear\left(i\right) \quad (2)$$

$$C\left(P_v\right)=\min\left\{\min_j\left\{C\left(P_{u_j}\right)+C\left(N_v\right)-Tear\left(j\right)\right\},\min_{j|(v,u_j)\in\tau}\left\{C\left(A_{u_j}\right)+C\left(N_v\right)-Tear\left(j\right)\right\}\right\} \quad (3)$$

If $\left\{i|\left(v,u_i\right)\notin\tau\right\}=\phi$ then $C\left(A_v\right)=\infty$

$$\text{Otherwise } C\left(A_v\right)=C\left(A_{u_j}\right)+\sum_{(v,u_i)\notin\tau,i\neq j}\min\left\{C\left(A_{u_i}\right),Tear\left(i\right)\right\}+\sum_{(v,u_i)\in\tau}Tear\left(i\right) \quad (4)$$

$$\text{where } j=\arg\min_{i|(v,u_i)\notin\tau}\left(C\left(A_{u_i}\right)-Tear\left(i\right)\right)$$

The auxiliary value $Tear(i)$ measures the cost of an empty solution for the subtree including the edge (v,u_i) and the subtree of u_i. In computing $C(P_v)$ we have to either pick the cheapest of two alternatives: (a) all the subtrees are empty except one which is connected (first term in (3)), (b) all the subtrees are empty except one that is disconnected but incident on v via a target edge (second term). In computing $C(A_v)$ we

find the best disconnected subtree, and allow the remaining subtrees to be either disconnected or empty. These formulae are implemented in a dynamic program as follows: (1) Visit V in postorder, computing $C(N_v)$, $C(P_v)$ and $C(A_v)$ for each $v \in V$. Obtain the minimal cost by $\min\{C(N_r),C(P_r),C(A_r)\}$. (2) Compute N_v, P_v and A_v by following the traceback pointers to get all those (v,u_j) edges that were chosen by the minimal cost while taking $C(P_{u_i})+C(v,u_i)$ or $C(A_{u_i})+C(v,u_i)$. The time complexity of this algorithm is $O(|S|)$.

3.3 Tag SNPs from genotypes

Up until now we have followed the standard assumption in the computational literature [13,24,25] that tag SNPs need to reconstruct the full binary haplotypes from binary haplotypes of the tag set. As experiments that provide haplotypes are expensive, most studies seek to obtain experimentally only genotypes. For such data, the problem of finding tag SNPs should be reformulated to reflect the fact that the input is *genotypes*, rather than haplotypes: Recall that standard genotyping has three possible calls per site: {0,0}, {1,1} and {0,1}, where the first two are homozygous and the latter is heterozygote. (The calls are often abbreviated to 0,1, and 2 respectively, and the genotype is represented as a vector over {0,1,2}.) The following question arises: Find a subset of SNPs given whose genotype calls one can completely identify the pair of haplotypes of an individual. We call such subset *phasing tag SNPs*.

Formally, let H be a set of haplotypes over a set S of SNPs, and consider genotypes formed from haplotype pairs in H. Denote by $g(k,l)_s$ the genotype formed from H_k and H_l on the SNP set S. We say that $\{i_1,i_2\}$ and $\{j_1,j_2\}$ are *distinct with respect to S* if there is $s \in S$ such that $g(i_1,i_2)_s \neq g(j_1,j_2)_s$.

Definition: $S' \subseteq S$ is a set of *phasing tag SNPs* if every two haplotype pairs from H are distinct with respect to S'. Hence, from the genotype calls of an individual for the set S', one can uniquely determine the exact sequence of the complete set S for each of its two haplotypes.

In general, the definitions of phasing tag SNPs and tag SNPs differ (see Fig. 2). The former is stronger:

Observation 5: If $S' \subseteq S$ are phasing tag SNPs then they are also tag SNPs.

Proof: All homozygous genotype-call vectors are distinct w.r.t. S': for all $i \neq j$, $g(i,i)_{S'} \neq g(j,j)_{S'}$.□

We now show that, surprisingly, under the perfect phylogeny model, tag SNPs and phasing tag SNPs are equivalent. This identifies the commonly used definition with the more theoretically sound one, and therefore justifies the application of the current body of theory on tag SNPs to genotype data.

Theorem 5: Suppose that the haplotypes in H satisfy the perfect phylogeny model on S. A set $S' \subseteq S$ is a tag SNPs set if and only if S' is a phasing tag SNPs set.

Proof: It suffices to prove the "only if" direction. Suppose to the contrary that S' are tag SNPs but not phasing tag SNPs. Let $G_i=\{H_1,H_2\}$ and $G_j=\{H_3,H_4\}$ be distinct haplotype pairs with the same genotype call vector for S', i.e., $g(1,2)_{S'}=g(3,4)_{S'}$. Since S' is a tag SNP set, it distinguishes H_1 and H_3, so there must be $s_1 \in S'$ such that G_i and G_j are heterozygous to s_1, and H_1 and H_3 have different alleles for s_1. Similarly there must be $s_2 \in S'$ such that G_i and G_j are heterozygous to s_2, and H_1 and H_4 have different alleles for s_2. Therefore G_i and G_j are oppositely phased on s_1 and s_2. Since H_1, H_2, H_3, and H_4 are distinct, they violate the 4 gamete rule on s_1,s_2, in contradiction to Theorem 1.□

4. Discussion

We studied here several questions arising in haplotype inference under the perfect phylogeny model. We introduced the model of xor-genotypes, and showed results that lay the computational foundation for the use of such data: (i) Inference of the sample haplotypes (up to negation) by adapting graph realization algorithms. (ii) Only two or three additional full genotypes are needed to completely resolve the haplotypes.

Simulations with genetic data show that xor genotypes are nearly as informative as full genotypes. Hence, genotyping methods that distinguish only between heterozygotes and homozygotes could potentially be applied to large scale genetic studies. Xor-genotypes may have economical advantage over the complete genotypes common today, since the information in a xor-genotype is only a fraction of the information given by a complete genotype. The feasibility and economic benefit of xor-genotype data cannot be appreciated by currently available technologies, but this work lays the foundation for evaluating the cost-effectiveness of technologies for obtaining such data.

The second part of the manuscript studied choosing a subset of the SNPs that fully describes the sample haplotypes. We provided efficient solutions to several optimization problems arising in this topic: We generalized previous results by finding optimal informative SNP set for any interesting set, and more generally, showed how to handle differential costs of SNPs. Finally, we have shown how to find tag SNPs for genotype data, which generalize the definition of tag SNPs to a more practical aspect.

Acknowledgements

We thank Orna Man for helpful ideas and practical knowledge. We thank Gadi Kimmel for productive discussions. We thank the reviewers for their helpful comments. I. P. was supported by an Eshkol postdoctoral fellowship from the Ministry of Science, Israel. R. S. was supported in part by the Israel Science Foundation (grant 309/02).

References

[1] Sachidanandam R, et al. (International SNP Map Working Group). A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 2001; 409(6822): 928-33.

[2] Patil N, et al. Blocks of Limited Haplotype Diversity Revealed by High Resolution Scanning of Human Chromosome 21. *Science*, 2001; 294(5547): 1719-23

[3] Daly MJ, Rioux JD, Schaffner SF, Hudson TJ and Lander ES. High resolution haplotype structure in the human genome. *Nature Genetics*, 2001; 29(2): 229-32.

[4] Jeffreys AJ, Kauppi L and Neumann R. Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex. *Nature Genetics*, 2001; 29(2):109–11.

[5] Nachman MW and Crowell SL. Estimate of the mutation rate per nucleotide in humans. *Genetics*, 2000; 156(1)**:** 297-304.

[6] Gabriel SB, et al. The structure of haplotype blocks in human genome. *Science*, 2002; 296(5576): 2225-9.

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

[8] Excoffier L and Slatkin M. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 1995; 12(5): 921-7.

[9] Gusfield D. Haplotyping as Perfect Phylogeny: Conceptual Framework and Efficient Solutions. *Proceedings of the Sixth Annual International Conference on Computational Biology* 2002 (*RECOMB* '02): 166-75.

[10] Bafna V, Gusfield D, Lancia G, and Yooseph S. Haplotyping as Perfect Phylogenty: A direct approach. Technical Report U.C. Davis CSE-2002-21, 2002.

[11] Eskin E, Halperin E and Karp RM. Efficient reconstruction of haplotype structure via perfect phylogeny. To appear in the *Journal of Bioinformatics and Computational Biology* (*JBCB*), 2003.

[12] Garey MR and Johnson DS Computers and Intractability, p. 222 Freeman, New York, 1979.

[13] Bafna V, Halldórsson BV, Schwartz R, Clark AG and Istrail S. Haplotypes and informative SNP selection algorithms: don't block out information. *Proceedings of the Seventh Annual International Conference on Computational Biology* 2003 (*RECOMB* '03): 19-27.

[14] Xiao W and Oefner PJ, Denaturing high-performance liquid chromatography: A review. *Human Mutation*, 2001; 17(6): 439-74.

[15] Bixby RE and Wagner DK. An almost linear-time algorithm for graph realization, *Mathematics of Operations Research*, 1988; 13(1): 99-123.

[16] Tutte WT. An Algorithm for determining whether a given binary matroid is graphic. *Proceedings of American Mathematical Society*, 1960; 11: 905-917.

[17] Gavril F and Tamari R. An algorithm for constructing edge-trees from hypergraphs, *Networks* 1983; 13:377-388.

[18] Zhang K, Deng M, Chen T, Waterman MS and Sun F. (2002) A dynamic programming algorithm for haplotype block partitioning. *Proceedings of the National Academy of Sciences*, 99(11): 7335-9.

[19] Chung RH and Gusfield D. Perfect Phylogeny Haplotyper: Haplotype Inferral Using a Tree Model. *Bioinformatics*, 2002; 19(6): 780-781.

[20] Johnson GC, et al. Haplotype tagging for the identification of common disease genes. *Nature Genetics*. 2001; 29(2): 233-7.

[21] Gusfield D. Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology. Cambridge University Press 1997.

[22] Hudson RR. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, 2002; 18(2): 337-38.

[23] Chung RH and Gusfield D. Empirical Exploration of Perfect Phylogeny Haplotyping and Haplotypers. *Proceedings of the nineth International Computing and Combinatorics Conference* 2003 (*COCOON* '03): 5-19.

[24] Sebastiani P, Lazarus R, Weiss ST, Kunkel LM, Kohane IS and Ramoni MF. Minimal haplotype tagging. *Proceedings of the National Academy of Sciences of the USA*, 2003; 100(17): 9900-5.

[25] Chapman JM, Cooper JD, Todd JA and Clayton DG. Detecting disease associations due to linkage disequilibrium using haplotype tags: a class of tests and the determinants of statistical power. *Human Heredity*, 2003; 56(1-3): 18-31.

[26] Kwok PY. Genetic association by whole-genome analysis. *Science*, 2001; 294(5547): 1669-70.

[27] Pe'er I and Beckmann JS. Resolution of haplotypes and haplotype frequencies from SNP genotypes of pooled samples. *Proceedings of the Seventh Annual International Conference on Computational Biology* 2003 (*RECOMB* '03): 237-246.