Journal of Bioinformatics and Computational Biology

The Incomplete Perfect Phylogeny Haplotype Problem

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

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

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

The problem of resolving genotypes into haplotypes, under the perfect phylogeny model, has been under intensive study recently. All studies so far handled missing data entries in a heuristic manner. We prove that the perfect phylogeny haplotype problem is NP-complete when some of the data entries are missing, even when the phylogeny is rooted. We define a biologically motivated probabilistic model for genotype generation and for the way missing data occur. Under this model, we provide an algorithm, which takes an expected polynomial time. In tests on simulated data, our algorithm quickly resolves the genotypes under high rates of missing entries.

Keywords: haplotype; haplotype block; genotype; SNP; algorithm; complexity; genotype phasing; haplotype resolution; perfect phylogeny.

1. Introduction

A central current challenge in human genome research is to learn about DNA differences among individuals. This knowledge will hopefully lead to finding the genetic causes of complex and multi-factorial diseases. The distinct single-base sites along the DNA sequence, which show variability in their nucleic acids contents across the population, are called *single nucleotide polymorphisms* (SNPs). Millions of SNPs have already been detected[23], and it is estimated that the total number of common SNPs is 10 million[18].

In diploid organisms (e.g. humans) there are two nearly-identical copies of each chromosome. Most techniques for determining SNPs provide a pair of readings, one from each copy, but cannot distinguish from which of the two chromosomes each reading came[16]. The goal of *phasing* (or *resolving*) is to infer that missing information. The original conflated data from both chromosomes are called the *genotype* of the individual, and is represented by a set of two nucleotide readings for each site. The two separated sequences corresponding to the two chromosomes of an

individual are called his/her *haplotypes*. If the two bases in a site are identical (resp. different), the site is called *homozygote* (resp., *heterozygote*). For recent reviews on biological and computational aspects of haplotype analysis see Halldorsson et al.[12] and Hoehe et al.[15].

Resolving the genotypes is a central problem in haplotyping. It has been argued that more accurate association studies can be performed once the genotypes are resolved[16,5]. In the absence of additional information, each genotype can be resolved in 2^{h-1} different ways, where h is the number of heterozygote sites in the genotype. To find the correct way, resolution is done simultaneously on all the available genotypes, and according to a model. A pioneering approach to haplotype resolution was Clark's parsimony-based algorithm[4]. A likelihood-based EM algorithm [7,19] gave better results. Stephens et al.[25] and Niu et al.[20] proposed MCMC-based methods which gave promising results. All of those methods assumed that the genotype data correspond to a single block with no recombination events. Hence, for multi-block data the block structure must be determined separately.

Recently, a new combinatorial formulation of the phasing problem was suggested by Gusfield[11]. According to this model, phasing must be done so that the resulting haplotypes define a *perfect phylogeny tree*. This model assumes that for the studied region along the chromosome, recombination occurred infrequently, and the infinite site model holds[11]. Gusfield showed how to solve the problem efficiently, and simpler algorithms were subsequently developed by Bafna et al.[2] and Eskin et al.[6]. Eskin et al.[6] showed good resolving results with small error rates on real genotypes. They also reported that their algorithm was faster and more accurate in practical settings than Stephens et al.'s method[25].

In real genotype data (e.g., refs[21,8,5]) some of the data entries are often missing, due to technical causes. Current phasing algorithms (which are based on perfect phylogeny) require complete genotypes. This situation raises the following algorithmic problem: Complete the missing entries in the genotypes and then resolve the data, such that the resulting haplotypes define a perfect phylogeny tree. We call this problem *incomplete perfect phylogeny haplotype* (IPPH). It was posed by Halldòrsson et al.[12]. In order to deal with such incomplete data, Eskin et al.[6] used a heuristic to complete the missing entries, and showed very good results. However, having an algorithm for optimally handling missing data entries should allow more accurate resolution. In this paper we address the IPPH problem.

A special case of IPPH was studied in phylogeny by Pe'er et al.[22]. In the *incomplete directed perfect phylogeny* problem, the input is an $n \times m$ species-characters matrix. The characters are binary and directed, i.e., a species can only gain characters, and certain characters are missing in some species. The question is whether one can complete the missing states in a way admitting a perfect phylogeny. Pe'er et al. provided a near optimal $\widetilde{O}(nm)$ time algorithm for the problem[a]. This problem

[a]We use $\widetilde{O}$ notation to suppress polylogarithmic factors in presenting complexity bounds. Formally, $\widetilde{O}(g(n)) := \{f(n) \mid \exists n_0 > 0, \exists c > 0, \exists d > 0, \forall n \geq n_0 : \ 0 \leq f(n) \leq c[\log n]^d g(n)\}$.

is a special case of IPPH in which all the sites in all genotypes are homozygote, and the root is known.

The IPPH problem has two variants: *rooted* (or *directed*) and *unrooted* (or general). In the rooted version, one haplotype is given as part of the input. This haplotype is referred to as the root of the tree, even though it may not be the real evolutionary root of the tree. This holds, since each of the haplotypes can be used as a root in the perfect phylogeny tree[10]. The unrooted version is a more direct formulation of the practice in biology, since in phasing, the root of the haplotypes is not given. However, we argue that the more restricted rooted version is of practical importance: Though theoretically finding a root might take an exponential time, in practice often there is one genotype which is complete and homozygote in all sites, which can be used as a root. As we shall demonstrate in Section 5 on simulated and real biological data, virtually always at least one such genotype exists. If there is no such genotype, one can use a genotype with few undetermined sites and enumerate the values in these sites. In the rare cases that this too is not feasible, one can physically separate the two chromosomes of a single individual and sequence one haplotype, as was done in Patil et al.[21]. This procedure is considerably more expensive than standard genotyping techniques, but it will be performed only for one individual, so the price is small. Thus, both variants of IPPH are biologically important.

In this paper, we show that rooted IPPH is NP-complete. The hardness of unrooted IPPH follows immediately from the hardness of determining the compatibility of unrooted partial binary characters (incomplete haplotype matrix)[24]. This was observed first by R. Sharan (private communication). However, this result does not imply the hardness of rooted version. In fact, our proof for rooted IPPH is quite involved.

To cope with the theoretical hardness of IPPH, we invoke a probabilistic approach. We define a stochastic model for generating the haplotypes and for the way missing entries occur in them. The model assumptions are mild and seem to apply to biological data. In addition, we assume that the number of sites m grows much more slowly than the number of genotypes n. Specifically, we assume that $m = o(n^{.5})$. As m is bounded by the block size which in practice is not more than a modest constant (10-30), this condition also holds in practice. We design an algorithm which always finds the correct solution, and under the assumptions above takes an expected time of $\widetilde{O}(m^2 n)$. A similar probabilistic approach leading to comparable results was developed simultaneously and independently by Halperin and Karp[13].

To test our algorithm, we applied it to simulated data using biologically realistic values of the parameters, and calculated an upper bound Γ on the main factor in the running time. Γm gives a bound on the number of times the polynomial algorithm of Peer et al.[22] would be invoked to complete the calculation. Γ may be exponential, but under the model assumptions it was shown to have an expected polynomial time. On data with 200 genotypes and 30 sites, we show that on average $\Gamma < 4000$

even when only two haplotypes are present and the rate of missing entries is 50%. For a more realistic case of five haplotypes and 20% missing entries, $\mathbb{E}[\Gamma] < 100$. Hence, the algorithm runs in modest time even far beyond the range of its provable performance.

The paper is organized as follows: Section 2 presents definitions and preliminaries. Section 3 shows the hardness result. Section 4 presents the algorithm and the probabilistic analysis. Section 5 summarizes our experimental results.

A preliminary version of this study was published in the Proceeding of the Second RECOMB Satellite Meeting on SNPs and Haplotypes[17].

2. Preliminaries

In this section we provide basic defnitions, lemmas and observations that are needed for our analysis. Figure 1 demonstrates the main definitions.

Given n genotypes, the haplotype inference problem is to find n pairs of haplotypes vectors that could have generated the genotypes vectors. Formally, the input is an $n \times m$ *genotype matrix* M, with $M[i,j] \in \{0,1,2\}$. The i-th row $M[i,*]$ describes the i-th genotype. The j-th column describes the alleles in the j-th location: 0 or 1 for two homozygote alleles, and 2 for a heterozygote site. A $2n \times m$ binary matrix M' is an *expansion* of the genotype matrix M if each row $M[i,*]$ expands to two rows denoted by $M'[i,*]$ and $M'[i',*]$, with $i' = n+i$, satisfying the following: for every i, if $M[i,j] \in \{0,1\}$, then $M[i,j] = M'[i,j] = M'[i',j]$; if $M[i,j] = 2$, then $M'[i,j] \neq M'[i',j]$. M' is also called a *haplotype matrix* corresponding to M.

Definition 1. Perfect Phylogeny Tree for a Matrix
A *perfect phylogeny* for a $k \times m$ haplotype matrix M' is a tree T with a root r, exactly k leaves and integer edge labels, and a binary label vector $(l^v(1) \ldots l^v(m))$ for each node v, that obeys the following properties:
1. Each of the rows in M' is the label of exactly one leaf of T.
2. Each of the columns labels exactly one edge of T.
3. Every edge of T is labelled by one column.
4. For any node v, $l^v(i) \neq l^r(i)$ if and only if i labels an edge on the unique path from the root to v. Hence, given the root label, the root-node paths provide a compact representation of all node labels.

An equivalent definition appeared in ref.[2]. Note that we disallow edges with multiple labels, and replace them by paths with a single label per edge.

Problem 1. The Perfect Phylogeny Haplotype Problem (PPH) [11]
Given a matrix M, find an expansion M' of M which admits a perfect phylogeny.

We now define a generalization of PPH that allows missing data entries. The input to our problem is an *incomplete genotype matrix*, i.e., a matrix M with $M[i,j] \in \{0,1,2,?\}$, where '?' indicates a missing data entry. The process of replacing each '?' by 0,1 or 2 is called *completing* the matrix M.

Problem 2. Incomplete Perfect Phylogeny Haplotype Problem (IPPH)
Given an incomplete genotype matrix M, can one complete M, so that there exists an expansion M' of M, which admits perfect phylogeny?

The following definitions are implicit in refs.[3,6].

Definition 2. Perfect Phylogeny Forest
Let M be a haplotype matrix, and let $P = (V_P, E_P)$ be a perfect phylogeny tree corresponding to M. The *perfect phylogeny forest* of P is a directed forest $F = (V_F, E_F)$ whose vertices are the edges of P, and for $u, v \in V_F$, u is a parent of v in F if and only if the edge corresponding to u in P is a parent of the edge corresponding to v in P.

Hence, the vertices of perfect phylogeny forest correspond to M''s columns, and reflect the order of mutations in the phylogeny tree. Clearly, each perfect phylogeny tree can be converted into perfect phylogeny forest and vice versa. Thus, M' admits a perfect phylogeny tree iff it admits a perfect phylogeny forest. For a column $j \in \{1, 2, \ldots, m\}$ of M', we denote by u_j its corresponding vertex in the perfect phylogeny forest.

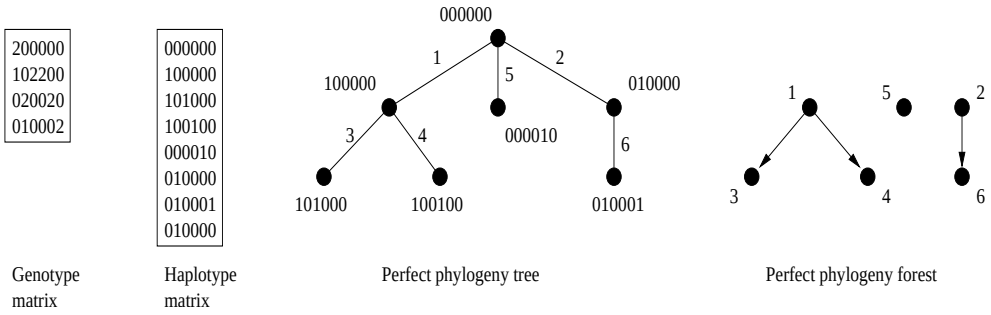

Fig. 1. Genotypes, haplotypes and trees. A genotype matrix M, a haplotype matrix M' that is an expansion of M, the perfect phylogeny tree of M', and the corresponding perfect phylogeny forest.

For a perfect phylogeny forest F, we say that two vertices are in *parenthood relation* if one is an ancestor of the other. Otherwise, we say that they are in *brotherhood relation*. Note that brothers can either be in different connected components, or be in the same component and have the root on the path connecting them.

The following special case of IPPH will be a main subject of our investigation.

Problem 3. Incomplete Perfect Phylogeny Haplotype, rooted version (ROOTED-IPPH)
Given an incomplete genotype matrix M and a haplotype r, can one complete M, such that there exists an expansion M' of M, which admits an perfect phylogeny, with r as a root?

In this problem, we can assume w.l.o.g. that the input root haplotype is $r_0 = (0, \ldots, 0)$ (ref. [10]). The following lemma explains the connection between F and M', and is key for our construction:

Lemma 1. *(Bafna et al.[2], Eskin et al.[6]) Let M' be a haplotype matrix, and let $F = (V_F, E_F)$ be a perfect phylogeny forest, corresponding to perfect phylogeny tree with the root $r_0 = (0, 0, \ldots, 0)$. F is perfect phylogeny forest of M' iff for all $u_a, u_b \in V_F$ and for every haplotype i:*

(1) If u_a is an ancestor of u_b then $M'[i, a] = 1$ or $M'[i, b] = 0$.
(2) If u_a and v_b are in brotherhood relation, then $M'[i, a] = 0$ or $M'[i, b] = 0$.

In the rest of this section, we provide our own definitions, building on those introduced above, and prove several lemmas which will be needed for our analysis.

Definition 3. Constrained Mixed Graph
A *constrained mixed graph* (CMG) is a triplet $G_c = (V, E, X)$, where $G = (V, E)$ is a graph and $X = \{X_1, X_2, ..., X_p\}$, where for each i: $X_i \subseteq V$. The sets X_i are called *XOR relations*. G has four types of edges: undirected, dashed undirected, directed and dashed directed.

Definition 4. Parenthood Connected Components
Two vertices u and v in a constrained mixed graph are in the same *parenthood connected component* if there exists a path between u and v consisting only of undirected or directed edges (a parenthood relation). Note, that edge directions are not important in this definition.

Definition 5. Constrained Mixed Completion Graph
For a constrained mixed graph $G_c = (V, E, X)$, we define its *constrained mixed completion graph* $G' = (V, E')$ to be a complete graph (with a single edge for each pair $u, v \in E$), where E' contains two types of edges: directed and dashed undirected. The edge types induce a labelling $L : E' \to \{0, 1\}$, where a directed edge is labelled with 0, and dashed undirected edge is labelled with 1. G' must maintain all the following properties:

(1) All G' edges maintain the following properties:
   (a) If $e : (u, v) \in E$ is an undirected edge then E' must contain a directed edge from u to v or from v to u.
   (b) Directed edges and dashed undirected edges in G are unchanged in G'.
   (c) If $e : (u, v) \in E$ is a dashed directed edge from u to v then the corresponding $e' : (u, v) \in E'$ must be a dashed undirected edge or a directed edge from u to v.

(2) G' contains a spanning forest $F = (V, E_F \subseteq E')$, consisting of directed edges only, such that:

(a) If node $u \in V$ is an ancestor of $v \in V$ in F, then there is a directed edge from u to v in G'.
(b) For any two nodes in V, if neither node is an ancestor of the other in F, then they are connected by a dashed undirected edge in G'.

(3) For each XOR relation X_i, for every three vertices: $x_{i,a}, x_{i,b}, x_{i,c} \in X_i$, the following holds: $L(x_{i,a}, x_{i,b}) \oplus L(x_{i,b}, x_{i,c}) \oplus L(x_{i,a}, x_{i,c}) = 0$.[b]

Problem 4. Constrained Mixed Graph Completion Problem (CMGC)
Given a constrained mixed graph G, provide a constrained mixed completion graph of G, if such exists.

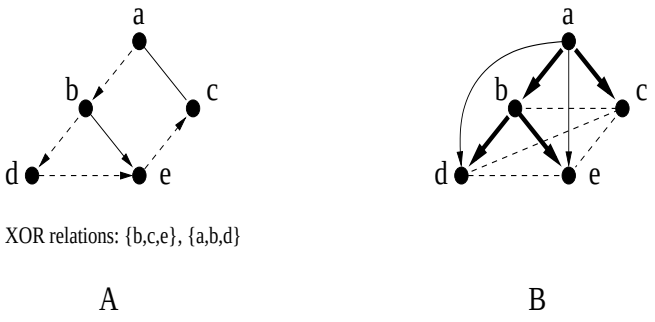

Fig. 2. Example of CMGC problem. A: an instance of a graph for CMGC with XOR relations. B: a possible solution for this instance. The edges of the forest appear in bold.

An example of CMGC problem is presented in Figure 2. The decision version of CMGC problem is to decide whether there exists a constrained mixed completion graph G' for G. An important property of the constrained mixed completion graph, is that it can be viewed as a directed spanning forest F, with additional edges between nodes, according to the relation of those nodes in the forest: a dashed undirected edge for a brotherhood relation, and a directed edge for a parenthood relation.

The following notation is adopted from Eskin et al.[6]: $c(M, x)$ is defined as the set of rows of M containing the value x in column c. Let c, c' be columns and let x, y be elements of $\{0, 1\}$. The pair c, c' *induces* (x, y) in M if $((c(M, x) \cap c'(M, y)) \cup (c(M, x) \cap c'(M, 2)) \cup (c(M, 2) \cap c'(M, y)) \neq \emptyset$. Let $R(M, c, c')$ be the set of pairs (x, y) such that (c, c') induces (x, y) in M. Note, that $R(M, c, c')$ does not contain pairs with '?', but only '0' and '1'. Note also that $R(M, c, c')$ contains (0,0) for every c, c' by our assumption that the root is $(0, \ldots, 0)$.

Let c, c' be two columns such that $c(M, 2) \cap c'(M, 2) \neq \emptyset$. Let M' be an expansion of the M, after completing the missing entries, which admits a perfect phylogeny. We

[b]The operator $\oplus$ denotes the boolean xor operator.

say that M' resolves the pair of columns (c, c') *unequally* if $\{(0,0),(0,1),(1,0)\} = R(M', c, c')$ and *equally* if $\{(0,0),(1,1)\} = R(M', c, c')$. According to Lemma 1, M' must resolve the pair (c, c') either equally or unequally, and can not resolve the pair in both ways.

For an incomplete genotype matrix M, we build a constrained mixed graph $G_c(M)$, where each column in M has a corresponding vertex in G_c. The edges represent the possible relations of the columns in the perfect phylogeny forest, and are determined according to lemma 1: For each two vertices u_a, u_b: (1) If $R(M, a, b)\backslash\{(0,0)\} = \{(1,1),(1,0)\}$ then u_a is an ancestor of u_b in F. The edge (u_a, u_b) is set as a directed edge from u_a to u_b. (2) If $R(M, a, b)\backslash\{(0,0)\} = \{(1,1)\}$ then u_a, u_b are in parenthood relation in F, but it is unknown which of the vertices is the ancestor. The edge (u_a, u_b) is set as an undirected edge. (3) If $R(M, a, b)\backslash\{(0,0)\} = \{(1,0),(0,1)\}$ then u_a, u_b are in brotherhood relation in F. The edge (u_a, u_b) is set as a dashed undirected edge. (4) If $R(M, a, b)\backslash\{(0,0)\} = \{(1,0)\}$ then either u_a is an ancestor of u_b in F, or u_a, u_b are in brotherhood relation in F. The edge (u_a, u_b) is set as a dashed directed edge from u_a to u_b. (5) If $R(M, a, b)\backslash\{(0,0)\} = \emptyset$ then the relation of u_a, u_b in F is unknown. In that case: $(u_a, u_b) \notin E$. The labelling of unlabelled edges corresponds to selecting the type of edge in the completion of G_c for solid undirected and for dashed directed edges.

In addition, for each row i, the set of columns $a_1, \ldots, a_t$, such that $M[i, a_1] = , \cdots, = M[i, a_t] = 2$, imply a XOR relation on the corresponding vertices $u_{a_1}, \ldots, u_{a_t}$. Each pair of vertices of G_c is labelled with $L : (u_a, u_b) \rightarrow \{0, 1, ?\}$ as follows: A solid (directed or undirected) edge, i.e., a parenthood relation, is labelled with 0; dashed undirected edge, i.e., a brotherhood relation, is labelled with 1; and all other cases, i.e., an unknown relation, are labelled with '?'. The last set is called *unlabelled pairs*.

Step 1: Primary Label Completion

A *primary label completion of* $G_c(M)$ is an assignment of a label s to unlabelled pairs of vertices, by performing the following step as long as possible: Find three vertices $x_{i,a}, x_{i,b}, x_{i,c} \in X_i$, such that $L(x_{i,a}, x_{i,b})$ and $L(x_{i,b}, x_{i,c})$ are set and $L(x_{i,a}, x_{i,c})$ is not, and assign: $L(x_{i,a}, x_{i,c}) = L(x_{i,a}, x_{i,b}) \oplus L(x_{i,b}, x_{i,c})$.

Define U_{G_c} to be the set of unlabelled pairs after primary label completion was performed. U_{G_c} is independent of the order of choosing the triplets[2].

Step 2: Secondary Label Completion

A *label completion of a constrained mixed graph* $G_c(M)$ is an assignment of a label in {0,1} to pairs $(u_a, u_b) \in U_{G_c}$, such that for each XOR relation X_i, for every three vertices $x_{i,a}, x_{i,b}, x_{i,c} \in X_i$, the condition: $L(x_{i,a}, x_{i,b}) \oplus L(x_{i,b}, x_{i,c}) \oplus L(x_{i,a}, x_{i,c}) = 0$ is satisfied.

After secondary label completion, we can perform label resolution using the incomplete haplotype matrix, which is defined as follows: Given an incomplete genotype matrix M, an *expansion* for M is in an incomplete haplotype matrix M' which satisfies the expansion rules for complete matrices, and also preserves all '?' values.

Formally, each row $M[i,*]$ expands to two rows denoted by $M'[i,*]$ and $M'[i',*]$, such that for every i, if $M[i,j] \in \{0,1,?\}$, then $M[i,j] = M'[i,j] = M'[i',j]$; if $M[i,j] = 2$, then either $M'[i,j] = 0$ and $M'[i',j] = 1$, or $M'[i,j] = 1$ and $M'[i',j] = 0$.

Step 3: Label Resolution

A *label resolution* of a Genotype matrix M is an expansion of M to an incomplete haplotype matrix M', according to the label function L: For every two columns a, b, if there exists i, such that $M[i,a] = M[i,b] = 2$, if $L(u_a, u_b) = 0$ resolve (a,b) equally and if $L(u_a, u_b) = 1$ resolve (a,b) unequally. The output of this process is an incomplete haplotype matrix.

Label resolution of an incomplete genotype can be done by algorithm $\mathcal{E}2M$ proposed by Bafna et al.[2]. Observe, that any submatrix $M[i,(a,b)]$, where $M[i,a]$ and $M[i,b]$ are not both equal 2, has a unique expansion in any incomplete haplotype matrix. Hence, for such submatrix, the resolution is not influenced by the label function.

Primary label completion was suggested by Bafna et al.[2] as part of an algorithm for complete genotype matrix phasing. Interestingly, Bafna et al. proved that once primary label completion is performed, for any possible (legal) secondary label completion of U_{G_c}, label resolution of the genotype matrix results in a haplotype matrix, which admits a perfect phylogeny. This is true for a complete genotype matrix (with no missing entries), but not for the incomplete case. A simple example for that is an incomplete haplotype matrix that does not admit a perfect phylogeny (see e.g., Pe'er et al.[22], Figure 2). Now consider such a matrix to be the input genotype matrix, by duplicating each haplotype to form a fully homozygote genotype. Here, no primary and secondary resolution is needed, since there are no heterozygotes in the matrix. Thus, every secondary resolution results in an incomplete haplotype matrix (namely, the same input matrix), which does not admit a perfect phylogeny. The following lemma describes a weaker connection between secondary label completion and the solution of IPPH.

Lemma 2. *Suppose M is an incomplete genotype matrix that has a completion that admits a perfect phylogeny. Then there exists some secondary label completion of U_{G_c}, such that a label resolution of the incomplete genotype matrix M gives an incomplete haplotype matrix, that can be completed to M'.*

Proof. Suppose M can be completed to a binary matrix M^*, so that there exists an expansion M' of M^*, which admits a perfect phylogeny. Let C be the set columns of M. The perfect phylogeny implies some label function f_L on the pairs of the vertices of $G_c(M)$, i.e., $\forall i,j \in C: \ f_L(u_i,u_j) \in \{0,1\}$. This complete label function can not contradict the XOR relations of $G_c(M)$ (for proof, see ref.[2]). Next, primary label completion of $G_c(M)$, for the known pairs, must match the labels in f_L, as there is only one possible primary label completion. Then, we can chose the following secondary label completion: $(u_i,u_j) \in U_{G_c}: \ L(u_i,u_j) = f_L(u_i,u_j)$, which obviously

gives an equivalent label function to f_L. Thus, using this secondary label completion of M, a label resolution of the incomplete genotype matrix M gives an incomplete haplotype matrix, that can be completed to M'. □

3. The Hardness Result

In this section we show that IPPH-rooted is NP-complete. Clearly, the problem belongs to NP. To prove NP-hardness, we will show the following polynomial reductions: 3-SAT $\propto$ CMGC $\propto$ ROOTED-IPPH.

We note that this also implies an alternative proof of the hardness of unrooted IPPH: to form the reduction ROOTED-IPPH $\propto$ IPPH, given an instance (M, r) of ROOTED-IPPH, we simply add the genotype row r to M. The resulting matrix M^* is the input to IPPH. In a solution to the latter, there will be a leaf labelled with r, and thus it solves the former problem. Conversely, if M has a solution with root r then it is also a solution for M^*. The same idea was used for another purpose by Bafna et al.[2].

We first prove the reduction from CMGC:

Theorem 1. *CMGC* $\propto$ ROOTED-IPPH

Proof. Given a constrained mixed graph $G_c = (V, E, X)$ for the CMGC problem, we build a matrix M, and set $r = (0, 0, \ldots, 0)$. (M, r) will serve as input for ROOTED-IPPH. Let $|X| = p$. M has dimensions $(2|E| + p) \times |V|$. For each $e \in E$ there are two corresponding rows, and their indices are denoted by N_e^0 and N_e^1. For each $X_i \in \{X_i\}_{1 \le i \le p}$ there is one row with index N_{X_i}. The column $i \in \{1, 2, ..., |V|\}$ corresponds to vertex u_i in G_c.

The construction of M is as follows:

(1) For each $e = (u_a, u_b) \in E$, we add two rows $M[N_e^0, *]$ and $M[N_e^1, *]$, such that $\forall u_c \in V \backslash \{u_a, u_b\}$, $M[N_e^0, c] = M[N_e^1, c] =?$, and:
   (a) If e is an undirected edge then $M[N_e^0, a] = 0, M[N_e^0, b] = 0, M[N_e^1, a] = 1, M[N_e^1, b] = 1$.
   (b) If e is a dashed undirected edge then $M[N_e^0, a] = 0, M[N_e^0, b] = 1, M[N_e^1, a] = 1, M[N_e^1, b] = 0$.
   (c) If e is a directed edge from u_a to u_b then $M[N_e^0, a] = 1, M[N_e^0, b] = 0, M[N_e^1, a] = 1, M[N_e^1, b] = 1$.
   (d) If e is a dashed directed edge from u_a to u_b then $M[N_e^0, a] = 0, M[N_e^0, b] = 0, M[N_e^1, a] = 1, M[N_e^1, b] = 0$.

(2) For each $\{X_i\}_{1 \le i \le p}$, we add one row $M[N_{X_i}, *]$, such that $\forall u_j \in X_i$: $M[N_{X_i}, j] = 2$ and $\forall u_k \in V \backslash X_i$: $M[N_{X_i}, k] =?$.

This completes the description of the reduction. Clearly the reduction is polynomial.

($\Rightarrow$)

Suppose that ROOTED-IPPH(M,r) = TRUE, i.e., M has an expansion M' that admits a perfect phylogeny tree, with r_0 as a root. Thus, M' has a directed perfect phylogeny forest $F = (V_F, E_F)$. Let $\widehat{F} = (V_F, \widehat{E}_F)$ be a complete graph, where for each $u, v \in V_F$, we add a directed edge from u to v if u is an ancestor of v in F, or a dashed undirected edge if neither node is in ancestor of the other.

We claim that $\widehat{F}$ is a constrained mixed completion graph of G_c. This is proven by checking that all three properties of $\widehat{F}$ as a constrained mixed completion graph of graph G_c hold (compare Definition 5). Property 2 hold since by the construction of $\widehat{F}$, F is a rooted spanning forest of $\widehat{F}$ as required. In order to prove property 3 we use Lemma 2 in Bafna et al.[2]: the structure of the rows $\{M[N_{X_i}, *]\}_{1 \le i \le p}$ forces that for each of the XOR relations, for every three vertices $x_{i,a}, x_{i,b}, x_{i,c} \in (X_i \subseteq V_F)$, the equation $L(x_{i,a}, x_{i,b}) \oplus L(x_{i,b}, x_{i,c}) \oplus L(x_{i,a}, x_{i,c}) = 0$ holds. Finally, property 1 holds, since for an edge $e \in E$ the values in the two corresponding rows $\{M[N_e^j, *]\}_{j \in \{0,1\}}$ are determined in step 1 of the construction of M: The edge (u, v) in graph G_c determines the possible relations of u and v in F. Since, by the assumption, M has an expansion M', that admits a perfect phylogeny forest F, it follows that for each $u, v \in F$, the edge $e' = (u, v) \in E_F$ must be set according to $e = (u, v) \in E$ in G_c: If e is an undirected edge then e' must be a directed edge; if e is a dashed undirected edge then e' must be a dashed undirected edge; if e is a directed edge from u to v then e' must be a a directed edge from u to v; and if e is a dashed directed edge from u to v then e' must be a dashed undirected edge or a directed edge from u to v. This proves property 1. Thus, $\widehat{F}$ is the constrained mixed completion graph of G_c, and CMGC(G_c) = TRUE.

($\Leftarrow$)

Suppose that CMGC (G_c) = TRUE, i.e., there exists a constrained mixed completion graph G' for G_c. According to the second property of G', there exists a directed forest $F = (E_F, V)$, which spans V. Due to the third property, the completion of edges in G_c, does not violate the XOR relations. We create an expansion M' of M as follows: resolve the '2' of the genotypes in those rows, according to G': for two vertices $\{u_a, u_b \in X_i\}_{1 \le i \le p}$, in case $M[N_{X_i}, a] = M[N_{X_i}, b] = 2$, if there is an undirected dashed edge between $u_a, u_b \in V$, then resolve the pair of columns (a, b) unequally, and if there is an directed edge between $u_a, u_b \in V$, then resolve the submatrix equally. Since those edges are completed in G' according to XOR relations (see Definition 5, property 3), each of the '2's in these rows can be resolved accordingly.

We denote the remaining $2|E| \times |V|$ matrix by M^*. Note that $M^*[i, j] \in \{0, 1, ?\}$. We call the $\{0, 1\}$ entries *"constants"*, and the '?' entries *"variables"*. We denote the set of column indices of constants in row i by C_i, and the set of column indices of variables in this row by V_i. Complete the variables entries in the matrix M^* to

create a matrix M^{**} as follows:

$$M^{**}[i,j]_{j\in V_i} = \begin{cases} 1 & \text{if } \exists c \in C_i \text{ s.t.: } M^*[i,c] = 1 \wedge u_j \text{ is an ancestor of } u_c \\ 0 & \text{otherwise} \end{cases}$$

M^{**} is a binary matrix and an expansion of M. We claim that M^{**} admits a perfect phylogeny forest. Moreover, this forest is F. This will be proven by showing that each two columns in M^{**} do not contradict F, and thus, according to Lemma 1, F is a perfect phylogeny forest of M^{**}.

Consider two vertices $u_a, u_b \in V$ and their corresponding columns a, b in M^{**}. For each row i, we examine the three possible cases for M^{**}:

(1) $u_a, u_b \in C_i$

$M(i,a)$ and $M(i,b)$ were set according to the edge $(u_a, u_b) \in E$, which by definition of G', does not contradict F.

(2) $u_a \in C_i, u_b \in V_i$

First, suppose $M^{**}[i,a] = 0$: If $M^{**}[i,b]$ is set to 0, then there is no contradiction for any relations of u_a and u_b in F. Otherwise, if $M^{**}[i,b]$ is set to 1, then there exists $c \in C_i$ such that $M'[i,c] = 1$ and u_b is an ancestor of u_c. Suppose, on the contrary, that u_a, u_b contradict F in row i. This means, that there are two rows j, k such that $M^*[j,a] = 1$,$M^*[j,b] = 0$,$M^*[k,a] = 1$,$M^*[k,b] = 1$, i.e., according to these rows, u_a is an ancestor of u_b in F. Since u_b is an ancestor of u_c, then u_a must be an ancestor of u_c. However, according to the construction of M, u_a cannot be an ancestor of u_c, since $M^{**}[i,a] = 0$ and $M^{**}[i,c] = 1$ and $a, c \in C_i$.

Second, suppose $M^{**}[i,a] = 1$: If $M^{**}[i,b]$ is set to 0, clearly u_b is not an ancestor of u_a, so M^{**} does not contradict F. Otherwise, if $M^{**}[i,b]$ is set to 1, then there exists $c \in C_i$, such that $M^{**}[i,c] = 1$ and u_b is an ancestor of u_c. In case $c = a$, u_a and u_b can not be in a brotherhood relation. In case $c \neq a$, u_a and u_c are in parenthood relation, and since u_b is an ancestor of u_c, it follows that u_a and u_b can not be in a brotherhood relation.

It follows that, in this case, M^{**} does not contradict F.

(3) $u_a, u_b \in V_i$

First, suppose that $M'[i,a]$ and $M'[i,b]$ are both set to 0. Obviously, the submatrix does not contradict F.

Second, suppose w.l.o.g. that $M'[i,a]$ is set to 0 and $M'[i,b]$ is set to 1. There exists $c \in C_i, c \neq a$ such that $M'[i,c] = 1$ and u_b is an ancestor of u_c. Suppose, on the contrary, that u_a, u_b contradict F in row i. This means, that there are two rows j, k such that $M^*[j,a] = 1$,$M^*[j,b] = 0$,$M^*[k,a] = 1$,$M^*[k,b] = 1$, i.e., according to these rows, u_a is an ancestor of u_b in F. Since u_b is an ancestor of u_c, then u_a must be an ancestor of u_c. However, in that case, $M'[i,a]$ should have been set to 1.

Third, suppose that $M'[i,a]$ and $M'[i,b]$ are both set to 1. There exist $c_a, c_b \in C_i$ such that $M'[i,c_a] = 1$, $M'[i,c_b] = 1$ and u_a is an ancestor of u_{c_a} and u_b is an ancestor of u_{c_b}. Clearly, u_{c_a} and u_{c_b} are in parenthood relation,

so w.l.o.g. suppose that u_{c_a} is an ancestor of u_{c_b}. Thus, both u_a and u_b are ancestors of u_{c_b}, and it follows that u_a and u_b can not be in brotherhood relation.

It follows that, in this case, M^{**} does not contradict F. □

Theorem 2. 3-SAT $\propto$ CMGC

Proof. For a 3-SAT instance Φ we build a CMGC graph G_c. Denote the variables of Φ by $\{Y_i\}_{1\leq i\leq t}$ and the clauses by $\{C_j\}_{1\leq j\leq s}$. Our construction will be formed from four types of CMG sub instances. First we define these four graph structures :

variable base graph contains two vertices denoted by x_0^i and x_1^i, with no edge between them. This graph is denoted by Var_i.

clause base graph (see Figure 3) contains 6 vertices denoted by $\{c_t^j\}_{0\leq t\leq 5}$. The edges are indicated in Figure 3. This graph is denoted by Cl_j.

positive variable connector (see Figure 3) contains 12 vertices denoted by $\{a_t^{i,j}\}_{0\leq t\leq 5}$, and $\{b_t^{i,j}\}_{0\leq t\leq 5}$. The edges are indicated in Figure 3. This graph is denoted by Pos.

negative variable connector (see Figure 3) contains 8 vertices denoted by $\{d_t^{i,j}\}_{0\leq t\leq 3}$ and $\{e_t^{i,j}\}_{0\leq t\leq 3}$. The edges are indicated in Figure 3. This graph is denoted by Neg.

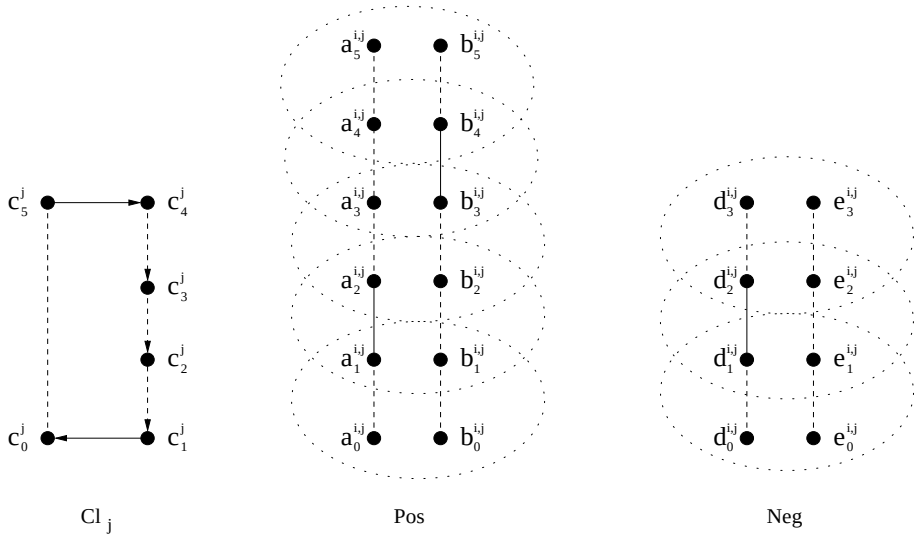

Fig. 3. The building blocks of the reduction. Clause base graph (left), positive variable connector (middle), and negative variable connector (right). In each case, the circled vertex sets represent XOR relations. Edge types (directed, undirected, solid, dashed) are as shown in the graphs.

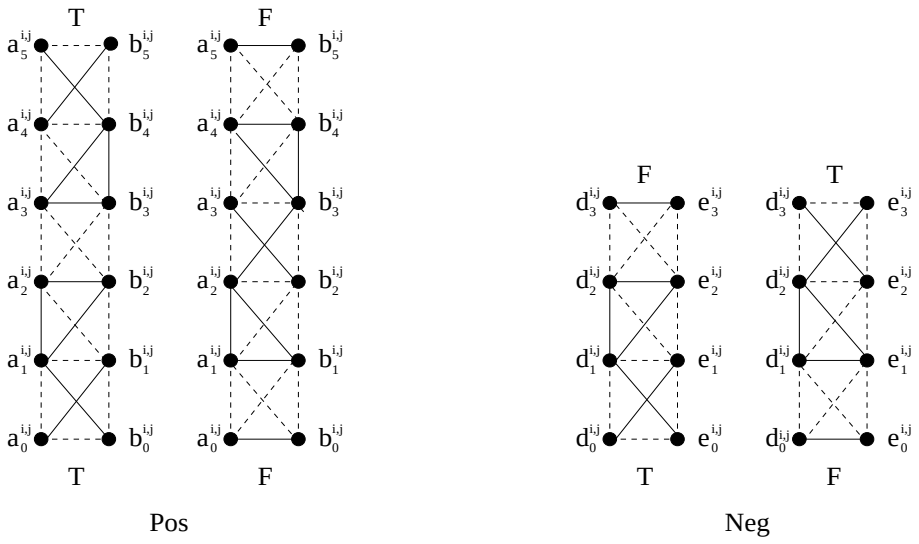

Fig. 4. Completion of variable positive and negative connectors. Note that in *Pos* (left) the completion propagates the type of the edge from the bottom to the top. In *Neg* (right) the types at the top and the bottom are reversed.

The XOR relations constrain the ways to complete the variable connectors. In fact, that there are two possible ways to complete the positive variable connector and the negative variable connector with undirected edges. Both of the ways for both types of connectors are presented in Figure 4. An important key to understanding the reduction, is that in the positive connector, the type (dashed or non-dashed) of edge $(a_0^{i,j}, b_0^{i,j})$ is the same as the type of the edge $(a_5^{i,j}, b_5^{i,j})$. In the negative connector, the type of edge $(d_0^{i,j}, e_0^{i,j})$ is the opposite from the edge $(d_3^{i,j}, e_3^{i,j})$. These two types will play the role of True and False in the reduction.

The construction of G_c is done as follows:

1. For each variable $\{Y_i\}_{1 \le i \le t}$ create a copy of variable base graph Var_i.
2. For each clause $\{C_j\}_{1 \le j \le s}$ create a copy of clause base graph Cl_j.
3. For all $1 \le j \le s$, for all $1 \le k \le 3$ do:
4. if Y_i is the k-th literal in clause C_j then do:
   create a copy of positive variable connector with superscripts i, j.
   identify $a_0^{i,j}$ with x_0^i and $b_0^{i,j}$ with x_1^i.
   identify $a_5^{i,j}$ with c_k^i and $b_5^{i,j}$ with c_{k+1}^i.
5. if $\neg Y_i$ is the k-th literal in clause C_j then do:
   create a copy of negative variable connector with superscripts i, j.
   identify $d_0^{i,j}$ with x_0^i and $e_0^{i,j}$ with x_1^i.
   identify $d_3^{i,j}$ with c_k^i and $e_3^{i,j}$ with c_{k+1}^i.

This concludes the reduction which is clearly polynomial. For convenience, we also

call an undirected dashed edge a *positive edge*, and a directed or undirected (solid) edge a *negative edge*.
($\Rightarrow$)

Suppose that 3-SAT(Φ) = TRUE. There exists a satisfying truth assignment $\tau : (Y_i) \rightarrow \{T, F\}$ for Φ. For each variable graph $\{Var_i\}_{1 \leq i \leq t}$ complete the edge according to the assignment: $\forall 1 \leq i \leq t : (x_0^i, x_1^i)$ is determined to be a positive edge if $\tau(Y_i) = T$, or a negative edge, otherwise. Now, resolve the XOR relations in all the variable connectors. In each the clause base graphs Cl_j, at least one of the three edges (c_1^j, c_2^j), (c_2^j, c_3^j), and (c_3^j, c_4^j), is a positive edge. It follows, that in each clause base graph there is more than one parenthood connectivity component. Each such component has only solid edges between members, and there is a directed edge between vertices c_a^j and c_b^j, only if $a = b + 1$. It follows, that a directed tree can be built in each of the components of a clause base graph, under the constrains of G_c.

In addition, any of the two possible completions of each of the variable connectors, for any assignment, provides parenthood connectivity components in the variable connectors as follows: each component is a connected component in the subgraph of the solid edges only. These components can be directed in a transitive fashion (see Figure 4). Thus, in each variable connector, one can form a directed sub-tree in each component, according to G_c constrains. Note, that subgraphs of two different variable connectors Con_1 and Con_2 may be in the same parenthood connectivity component. This may happen only when two variable connectors are connected to the same clause base graph, to edges (c_1^j, c_2^j) and (c_3^j, c_4^j) respectively, and when (c_2^j, c_3^j) is a directed solid edge and (c_1^j, c_2^j) and (c_3^j, c_4^j) are undirected dashed edges. In this case, there is only one directed edge which connects Con_1 and Con_2, so trees T_1 and T_2 can be built on Con_1 and Con_2 separately and directed using the common edge, and then T_1 and T_2 can be united to a spanning directed tree on $Con_1 \cup Con_2$.

It follows that the graph can be divided into h parenthood connectivity components $\{R_i\}_{1 \leq i \leq h}$, where a directed spanning tree T_i can be built in each of this components, under the constrains of G_c. Since each of the trees is in different parenthood connectivity component, then $\bigcup_{i=1}^{h} T_i$ is a directed forest spanning on G_c vertices. The constrained mixed completion graph can now be accomplished simply by completing the rest of the missing edges, in each parenthood connectivity component according to its spanning tree, and between the components, by undirected dashed edges. It follows that CMGC (G_c)=TRUE.
($\Leftarrow$)

Suppose that 3-SAT(Φ) = FALSE. Then for each truth assignment to the variables at least one of the clauses has *all* literals assigned to be FALSE. This implies that in any completion of G_c, there will be always one clause base graph Cl_j, such that all the three edges: (c_1^j, c_2^j), (c_2^j, c_3^j) and (c_3^j, c_4^j), are negative, i.e., solid directed edges. Thus c_5^j must be an ancestor of c_0^j in the forest. But this contradicts the undirected dashed edge between c_0^j and c_5^j, so a spanning forest which satisfied

G_c constraints does not exist. Thus, CMGC (G_c)=FALSE. □

4. An Algorithmic Solution for IPPH

In spite the negative results of Section 3, we provide an efficient algorithmic approach to IPPH. We propose a probabilistic model for data generation and argue that the model holds for biological data. Under this model, we provide an algorithm that takes an expected polynomial time for both the rooted and the unrooted versions of IPPH. A similar probabilistic approach leading to comparable results was developed simultaneously and independently by Halperin and Karp[13].

Pe'er et al.[22] suggested an algorithm that requires $\widetilde{O}(mn)$ time for solving the rooted version of perfect phylogeny with missing data on an $n \times m$ haplotype matrix. Let the input incomplete haplotype matrix be $\widetilde{M}$, with $\widetilde{M}[i,j] \in \{0,1,?\}$, and let the root be r. We denote by $\mathsf{IDP}(\widetilde{M},r)$ the completed matrix obtained by performing this algorithm on $\widetilde{M}$. We also use $\mathsf{IDP}(\widetilde{M})$ to denote IDP $(\widetilde{M},r_0)$. We use $h(\cdot,\cdot)$ to denote the Hamming distance between two binary vectors. We use $\sigma_0(j)$ and $\sigma_1(j)$ for the numbers of 0s and 1s in the j-th column of M, respectively.

Suppose the root r_0 is known. Given an incomplete matrix M, we build a constrained mixed graph, as described in Section 2. We then perform primary label completion. According to Lemma 2, if M can be completed to M^* so that there exists an expansion M' of M^* that admits a perfect phylogeny, then there exists some secondary label completion of U_{G_c}, that can form the basis to completion of M^*. Thus, the computational challenge is to find such secondary label completion. Suppose we were able to guess the correct secondary label completion. In that case, let $\widetilde{M}$ be the resulting *incomplete* haplotype matrix, generated by performing label resolution accordingly. A completion of $\widetilde{M}$ can be done in polynomial time by computing $\mathsf{IDP}(\widetilde{M})$. Hence, the bottleneck step is finding a secondary label completion.

Due to the hardness result in Section 3, a polynomial time algorithm for finding the correct secondary label completion does not exist, unless P=NP. However, by making several assumptions on the properties of the genotype data, this can be performed by a polynomial expected time algorithm. We now describe these assumptions, and for each one, we provide its biological motivation:

(1) Each entry value in the original genotype matrix is replaced by '?' with probability $\widetilde{p}$, independently of the other values. This assumption makes sense as missing data entries are caused by technical problems in the biological experiment, that tend to generate independent "misses" ('?'s).

    The same value $\widetilde{p}$ may be used for all entries. One may claim, that occasionally different SNPs may have different probability for a missing entry, due to distinct difficulties in sequencing different regions in the human genome. In that case, we denote by $\widetilde{p_i}$ the probability for a missing entry in the i-th SNP and set $\widetilde{p}$ to be: $\widetilde{p} \equiv \max_i\{\widetilde{p_i}\}$.

(2) Each haplotype h_i, which is a node in a perfect phylogeny tree, is chosen to be in a genotype with probability of α_i, independently. This assumption is also

made in the Hardy-Weinberg equilibrium model[14]. Moreover, we assume that these probabilities do not depend on n or m.

(3) The number of columns m grows much more slowly than the number of rows n. Specifically, we use $m = o(n^{.5})$. This assumption applies in all biological scenarios: In future experiments, the number of genotypes is expected to be even larger than today, while m is not expected to grow substantially, since m is the size of a "block", i.e., a region in the chromosome where the number of recombination events in the sampled population is small. A constant bound on m is thus plausible, but for our analysis, a much weaker assumption than that is required.

Prob-IPPH(M):

1. Let $G_c(M) = (V, E, X)$ be the constrained mixed graph of M.
2. Let $\overline{r}$ be a vector such that $\overline{r}_j = 0$ if $\sigma_0(j) > \sigma_1(j)$ and 1 otherwise.
3. Perform primary label completion of G_c(M).
4. For $i = 0 \rightarrow \binom{m}{2}$
   For each possible root $r \in \{0,1\}^m$, such that $h(r, \overline{r}) = i$ do
   Relabel the matrix entries according to r, so that r_0 is the new root.
   For each possible secondary label completion of U_{G_c},
   such that $|\{(u_a, u_b) : (u_a, u_b) \in U_{G_c} \wedge\ L((u_a, u_b)) = 0\}| = i$ do
   Perform label resolution of M to $\widetilde{M}$.
   If IDP($\widetilde{M}$) is compatible then output IDP($\widetilde{M}$) and halt.
5. Output: "no solution".

Fig. 5. An algorithm for IPPH.

Our algorithm was designed to solve IPPH under the assumptions above. Informally, algorithm Prob-IPPH(M) ignores the missing data entries in order to decide the relation between each two columns in the matrix. As we shall prove, if it is impossible to conclude the relation deterministically from the matrix, with high probability, a correct relation is obtained just by guessing.

Theorem 3. *Under the assumptions of the model, algorithm Prob-IPPH(M) solves IPPH correctly within expected time of $\widetilde{O}(m^2 n)$.*

Proof. Correctness: Algorithm Prob-IPPH(M) enumerates all possible roots and all possible relations between every pair of columns (parenthood or brotherhood). Thus, correctness trivially follows.

Complexity: Steps 1-3 can all be done in $O(m^2 n)$ time. The main time consuming step of the algorithm is step 4. The algorithm can stop for any $0 \leq i \leq m^2$. We denote by S_0, an upper bound to the running time of the algorithm, when it stops for $i = 0$, and by E_{S_0} the event that the algorithm stops for $i = 0$. Similarly, we denote by $\overline{S}_0$, an upper bound to the running time of the algorithm, when it does not stop for $i = 0$, and by $E_{\overline{S}_0}$ the event that the algorithm does not stop for $i = 0$.

Trivial upper bounds for S_0 and $\overline{S}_0$ are:

$$\begin{aligned} S_0 &= \widetilde{O}(m^2 n), \\ \overline{S}_0 &= \widetilde{O}(m^2 n 2^{m^2}). \end{aligned} \tag{1}$$

Let $F_{i,j}$ be the set of rows a such that $M[a,i] = M[a,j] = 1$, or $M[a,i] = 1$ and $M[a,j] = 2$, or $M[a,i] = 2$ and $M[a,j] = 1$. Clearly, if $F_{i,j} \neq \emptyset$ then the columns i, j are in parenthood relation.

Definition 6. Informative and Enigmatic Pairs of Columns
A pair of columns i, j in an incomplete genotype matrix is called an *informative pair* if there is at least one row a, such that $a \in F_{i,j}$ in the original *complete* genotype matrix, i.e., the two corresponding vertices of the columns in the perfect phylogeny forest are in parenthood relation. The row a is called an *informative row w.r.t. columns* i, j.
A pair of columns i, j in an incomplete genotype matrix is called an *enigmatic pair* if the relation between i, j can not be concluded directly from these columns, and there exists at least one row a, such that $M[a,i] =?$ or $M[a,j] =?$. Such row a is called an *enigmatic row w.r.t. columns* i, j.

Let $I_{i,j}$ be the event that that the pair of columns i, j is an informative pair, and let $E_{i,j}$ be the event that the pair of columns i, j is an enigmatic pair. Let $I_{i,j}^{(a)}$, denote the event that row $a \in F_{i,j}$. We denote the set of all *pairs* of haplotypes, which create a genotype which belongs to $F_{i,j}$ by $\mathcal{H}_{F_{i,j}}$. Now, according to assumption (2), the probability that row a belongs to $F_{i,j}$ is $\Pr[I_{i,j}^{(a)}] = \sum_{h_a, h_b \in \mathcal{H}_{F_{i,j}}} \alpha_a \alpha_b$. We denote $\Pr[I_{i,j}^{(a)}]$ by $q_{i,j}$.

Let $E_{i,j}^{(a)}$ denote the event that the row a is enigmatic w.r.t. the columns i, j. The probability of $E_{i,j}^{(a)}$, for all a, i, j is $p = 2\widetilde{p}(1-\widetilde{p}) + \widetilde{p}^2$. We now calculate the joint probability $\Pr[I_{i,j}, E_{i,j}]$:

$$\begin{aligned} \Pr[I_{i,j}, E_{i,j}] &= \Pr[\forall a : \{I_{i,j}^{(a)} \rightarrow E_{i,j}^{(a)}\}] - \Pr[\forall a : \{\neg I_{i,j}^{(a)}\}] \\ &= \Pr[\forall a : \{\neg I_{i,j}^{(a)} \vee E_{i,j}^{(a)}\}] - \Pr[\forall a : \{\neg I_{i,j}^{(a)}\}] \\ &= \Pr[\forall a : \{\neg (I_{i,j}^{(a)} \wedge \neg E_{i,j}^{(a)})\}] - \Pr[\forall a : \{\neg I_{i,j}^{(a)}\}] \\ &= [1 - q_{i,j}(1-p)]^n - (1-q_{i,j})^n. \end{aligned}$$

Next, we calculate the conditional probability of a pair of columns to be an informative pair, when the pair is known to be enigmatic:

$$\begin{aligned} \Pr[I_{i,j} | E_{i,j}] &= \frac{\Pr[I_{i,j}, E_{i,j}]}{\Pr[E_{i,j}]} \\ &= \frac{[1-q_{i,j}(1-p)]^n - (1-q_{i,j})^n}{1-(1-p)^n} \\ &\leq \frac{[1-q_{i,j}(1-p)]^n}{1-(1-p)^n}. \end{aligned}$$

The probability for a pair to be not informative, when it is known to be enigmatic is:

$$\begin{aligned}\Pr[\neg I_{i,j}|E_{i,j}] \geq & \quad 1-\frac{[1-q_{i,j}(1-p)]^n}{1-(1-p)^n} \\ = & \quad \frac{1-(1-p)^n-[1-q_{i,j}(1-p)]^n}{1-(1-p)^n} \\ \geq & \; 1-(1-p)^n-[1-q_{i,j}(1-p)]^n.\end{aligned} \tag{2}$$

Note that $\Pr[\neg I_{i,j}|E_{i,j}]$ is the probability for a "success" with respect to columns i,j: Given that the pair i,j is enigmatic (i.e., we can not conclude its relation), the pair is not informative, which means that we can be sure that the columns relation is brotherhood.

We use the following definitions:

$$u = \max_{i,j}\{1-p, 1-q_{i,j}(1-p)\}.$$

Due to assumption (2), $0 < u < 1$, and does not depend on n or m. When substituting (3) into inequality (2), we get:

$$\Pr[I_{i,j}|E_{i,j}] \leq 2u^n.$$

Since there are $\binom{m}{2}$ pairs of columns, the probability that at least one of the enigmatic pairs is an informative pair, can be bounded using a union bound:

$$\begin{aligned}\Pr[\exists i,j: \; I_{i,j}|E_{i,j}] &= \Pr[\textstyle\bigcup_{i<j}(I_{i,j}|E_{i,j})] \\ &\leq \textstyle\sum_{i<j}\Pr[I_{i,j}|E_{i,j}] \\ &\leq \quad \binom{m}{2}2u^n.\end{aligned}$$

Thus, the complementary event, which represents "success", can be bounded by:

$$\Pr[\forall i,j: \; \neg I_{i,j}|E_{i,j}] \geq 1-\binom{m}{2}2u^n.$$

If the relation between two columns can not be concluded, then the algorithm starts with a guess of a brotherhood relation. Thus, an error might occur when $i=0$, only if a pair is an informative pair. Since $m = o(n^{.5})$, we replace n with c_1m^2, where c_1 is a constant. There exists m_0, such that $\forall m \geq m_0$ the probability that the algorithm finds the correct solution when i=0, and when the root is known to be $\widetilde{r}$, is:

$$\Pr[E_{S_0}|\text{ root is } \widetilde{r}] \geq 1-m^2u^{c_1m^2}. \tag{3}$$

We now calculate the probability for an error in deciding the root, when $i=0$. Denote by r the root calculated by the algorithm when $i=0$. Let P_i^0, P_i^1 be the probabilities for 0 and 1 in the i-th row, respectively. Hence $P_i^0 + P_i^1 = 1-\overline{p}$, where $\overline{p}$ is the probability for '?' in a haplotype. A specific site in the genotype is missing if at least one out of the two corresponding sites in the haplotypes is missing. Thus, $\widetilde{p} = 1-(1-\overline{p})^2$ or, equivalently, $\overline{p} = 1-\sqrt{1-\widetilde{p}}$. Let n_i^0, n_i^1 be the number of 0 and 1 in the i-th row, respectively. Without loss of generality, suppose

that $P_i^0 < P_i^1$, then the root can be determined to be '1' in the i-th component, according to the majority rule in determination of the root in perfect phylogeny (see [10]). The probability for an error in the i-th component can be bounded using Chernoff bound[1]:

$$\begin{aligned}
\Pr[r_i \neq \widetilde{r}_i] &= \Pr[n_i^0 > n_i^1 \mid P_i^0 < P_i^1] \\
&= \Pr[n_i^0 > \tfrac{1-\overline{p}}{2} \mid P_i^0 < P_i^1] \\
&= \Pr[n_i^0 > nP_i^0 + nP_i^0(\tfrac{1-\overline{p}}{2P_i^0} - 1) \mid P_i^0 < P_i^1] \\
&\leq e^{-\frac{nP_i^0(\frac{1-\overline{p}}{2P_i^0}-1)^2}{4}} \\
&= e^{-bn},
\end{aligned}$$

where $b = \frac{P_i^0}{4}(\frac{1-\overline{p}}{2P_i^0} - 1)^2$ is a constant. Using the union bound again, there exists m_0, such that $\forall m \geq m_0$ the probability for the root to be correct, when $i = 0$

$$\Pr[r = \widetilde{r}] \geq 1 - me^{-bc_2m^2},$$

where c_2 is a constant. Now, we can bound the probability that the algorithm stops when $i = 0$:

$$\begin{aligned}
\Pr[E_{S_0}] &\geq \Pr[E_{S_0}, r = \widetilde{r}] \\
&= \Pr[E_{S_0} | r = \widetilde{r}] \Pr[r = \widetilde{r}] \\
&\geq (1 - m^2u^{c_1m^2})(1 - me^{-bc_2m^2}) \\
&\geq 1 - e^{-c_3m^2},
\end{aligned} \tag{4}$$

where c_3 is a constant. Using inequality (1), we are now able to bound the expected running time of the algorithm:

$$\begin{aligned}
\mathbb{E}[\text{running time}] &\leq \Pr[E_{S_0}]S_0 + (1 - \Pr[E_{S_0}])\overline{S_0} \\
&\leq \widetilde{O}(m^2n) + e^{-c_3m^2}\widetilde{O}(m^2n2^{m^2}),
\end{aligned} \tag{5}$$

Since $m = o(n^{.5})$ we can choose c_1 large enough (c_3 is larger when c_1 increases), such that the second summand vanishes for $n \to \infty$, and thus:

$$\mathbb{E}[\text{running time}] = \widetilde{O}(m^2n).$$

Observe, that in addition to proving that the expected running time is polynomial, we also showed that the running time is polynomial with high probability. □

Note that the above analysis applies also when the root is known. In that case, obviously, we need not enumerate all possible roots, so the worst case running time can only improve. Asymptotically, the expected running time is the same.

5. Experimental Results

In order to assess our algorithm, we applied it on simulated data. The simulations used parameters which were adopted from several large scale biological studies[5,21,8]. By Theorem 3 the algorithm always outputs a correct solution. Although we proved that under our model assumptions the expected running time is $\widetilde{O}(m^2n)$, we wanted to estimate the actual running time, under realistic biological parameters and beyond the range of the model assumptions. Specifically, we wanted to calculate the expected number of different phylogenic tree solutions for a given data set. The proof of Theorem 3 implies that $\Gamma = 2^{|U_{G_c}|}$ is an upper bound on the number of different phylogeny solutions, and the dominant factor in the complexity of the algorithm, in the rooted version of the problem.

In each different experiment, we randomly generated $N = 10^5$ perfect phylogeny trees. We used the following procedure to generate a perfect phylogeny tree of haplotypes: We start with a binary root vector with $m = 30$ sites. Initially, no site is marked. In each step, we randomly pick a node from the current tree and an unmarked site, add a new child haplotype to that node in which only the state of that site is changed, and mark the site. For each tree, we randomly chose k haplotypes for reconstructing the genotypes, where $k = 2, 3, \ldots, 9$. We assigned frequencies, denoted by $\alpha_1, \alpha_2, \ldots, \alpha_k$, to the k chosen haplotypes, such that $\sum_{i=1}^{k} \alpha_i = 1$ and $\forall i: \ \alpha_i \geq 0.05$. For each tree, different frequencies were assigned. Next, we generated 200 genotypes according to the chosen haplotypes and their assigned frequencies. Introducing missing data entries to the genotypes was performed as follows: Each site in the genotypes data was flipped into a missing entry independently with probability p. Since we observed in real data $p \approx 0.1$ (ref.[5,8]), we checked a wider range: $p = 0, 0.05, \ldots, 0.5$. Thus, for each sampled tree $T_j : j = 1, 2, \ldots, N$, we sampled one incomplete genotype matrix M_j of size 200×30. We applied our algorithm on each M_j. We denote $U_{G_c(M_j)}$ by U_j. After performing steps 1-3 of the algorithm, we stopped at $i = 0$ and calculated $2^{|U_j|}$. As was shown in Section 4, if the secondary label completion is known, it is possible in $\widetilde{O}(m^2n)$ time to output the solution to IPPH. Hence, completion of the algorithm, for each M_j, should take less than $2^{|U_j|}\widetilde{O}(m^2n)$ time. The dominating factor in the running time is the random variable $2^{|U_j|}$, whose expectation is approximated by: $\mathbb{E}[\Gamma] = \mathbb{E}[2^{|U_j|}] \approx \frac{1}{N}\sum_{j=1}^{N} 2^{|U_j|}$.

The results are presented in Figure 6. In all experiments, $\mathbb{E}[\Gamma]$ was below 3500 (compared to a theoretical upper bound of $2^{\binom{m}{2}} = 2^{435}$) . When the missing data rate is below 20%, $\mathbb{E}[\Gamma]$ was smaller than 100. Another observation, is that the larger the number of chosen haplotypes, the smaller the value of $\mathbb{E}[\Gamma]$. Notably, in all cases we found a correct root: either by finding at least one haplotype, which is homozygote with no missing entries in all sites, or by using the majority rule described in the algorithm.

To demonstrate that in real biological data, a root is readily available, we chose the genotype data of Daly et al.[5]. This data set consists of 103 SNPs and 129

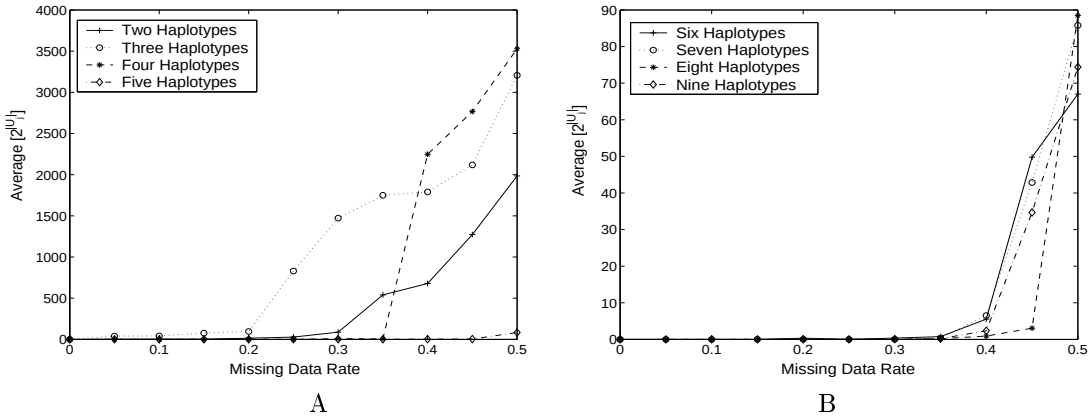

Fig. 6. Simulation results: both figures show the average of $2^{|U_j|}$ (y-axis), which represents the dominating factor in the running time of the algorithm for different missing data rates (x-axis). Each different line in the figures corresponds to a different number of haplotypes chosen from the tree (see legend).

genotypes. We checked all possible $\binom{103}{2} = 5253$ blocks. In *all* the blocks of size 65 or smaller, there was always at least one genotype that was homozygote in all alleles and without any missing entries. This genotype is actually a haplotype, since it can be resolved in only one possible way, and hence, it can be used as a root. Since the size of a block is almost always smaller than 30, this naive simple method can be used for finding a root in biological data.

6. Concluding Remarks

We investigated the incomplete perfect phylogeny haplotype problem. The goal is phasing of genotypes into haplotypes, under the perfect phylogeny model, where some of the data are missing. We proved that the problem in its rooted version is NP-complete. We also provided a practical expected polynomial-time algorithm, under a biologically motivated probabilistic model of the problem. We applied our algorithm on simulated data, and concluded that the running time and the number of distinct candidate phylogeny solutions are relatively small, under a broad range of biological conditions and parameters, even when the missing data rate is 50%. An accurate treatment for phasing of genotypes with missing entries can therefore be obtained in practice. In addition, due to the small number of phylogenetic solutions observed in simulations, incorporation of additional statistical and combinatorial criteria with our algorithm is feasible.

After the completion of this study, Gramm et al.[9] reported on another investigation of the ROOTED-IPPH problem. They proved that this problem is NP-complete even when the phylogeny is a path and only one allele of every polymorphic site is present in the population in its homozygous state. This provides an alternative

proof that the ROOTED-IPPH problem is NP-complete. They also give a linear-time algorithm for the problem for the special case, in which the phylogeny is a path.

Acknowledgments

This research was supported by the Israel Science Foundation (grant 309/02). We thank Roded Sharan for fruitful discussions.

References

1. N. Alon and J. H. Spencer. *The Probabilistic Method*. John Wiley and Sons, Inc., 2000.
2. V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogenty: A direct approach. *Journal of Computational Biology*, 10(3-4):323–340, 2003.
3. V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogenty: A direct approach. *Journal of Computational Biology*, 10(3-4):323–340, 2003.
4. A. Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. *Molecular Biology and Evolution*, 7(2):111–22, 1990.
5. M. J. Daly, J. D. Rioux, S. F. Schaffner, T. J. Hudson, and E. S. Lander. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29(2):229–232, 2001.
6. E. Eskin, E. Halperin, and R. M. Karp. Large scale reconstruction of haplotypes from genotype data. In *Proceedings of the Seventh Annual International Conference on Research in Computational Molecular Biology (RECOMB 03)*, pages 104–113. The Association for Computing Machinery, 2003.
7. L. Excoffier and M. Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Molecular Biology and Evolution*, 12(5):912–7, 1995.
8. S. B. Gabriel, S.F. Schaffner, H. Nguyen, J. M. Moore, J. Roy, B. Blumenstiel, J. Higgins, M. DeFelice, A. Lochner, M. Faggart, S. N. Liu-Cordero, C. Rotimi, A. Adeyemo, R. Cooper, R. Ward, E. S. Lander, M. J. Daly, and D. Altshuler. The structure of haplotype blocks in the human genome. *Science*, 296:2225–2229, 2002.
9. J. Gramm, T. Nierhoff, R. Sharan, and T. Tantau. On the complexity of haplotyping via perfect phylogeny. In *Proceedings of the Second RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotypes*, pages 35–46, 2004.
10. D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, 21:19–28, 1991.
11. D. Gusfield. Haplotyping as perfect phylogeny: conceptual framework and efficient solutions. In *Proceedings of the Sixth Annual International Conference on Research in Computational Molecular Biology (RECOMB 02)*, pages 166–175. The Association for Computing Machinery, 2002.
12. B. V. Halldorsson, V. Bafna, N. Edwards, R. Lippert, S. Yooseph, and S. Istrail. Combinatorial problems arising in SNP. In *Proceedings of the Fourth International Conference on Discrete Mathematics and Theoretical Computer Science (DMTCS)*, volume 2731 of *Lecture Notes in Computer Science*, pages 26–47. Springer, 2003.
13. E. Halperin and R. M. Karp. Perfect phylogeny and haplotype assignment. In *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB 04)*, pages 10–19. The Association for Computing Machinery, 2004.
14. G. H. Hardy. Mendelian proportions in a mixed population. *Science*, 18:49–50, 1908.

15. M. R. Hoehe. Haplotypes and the systematic analysis of genetic variation in genes and genomes. *Pharmacogenomics*, 4(5):547–570, 2003.
16. M. R. Hoehe, K. Kopke, B. Wendel, K. Rohde, C Flachmeier, K. K. Kidd, W. H. Berrettini, and G. M. Church. Sequence variability and candidate gene analysis in complex disease: association of μ opioid receptor gene variation with substance dependence. *Human Molecular Genetics*, 9:2895–2908, 2000.
17. G. Kimmel and R. Shamir. The incomplete perfect phylogeny haplotype problem. In *Proceedings of the Second RECOMB Satellite Workshop on Computational Methods for SNPs and Haplotypes*, pages 59–70, 2004.
18. L. Kruglyak and D. A. Nickerson. Variation is the spice of life. *Nature Genetics*, 27:234–236, 2001.
19. J. Long, R. C. Williams, and M. Urbanek. An EM algorithm and testing strategy for multiple-locus haplotypes. *American Journal of Human Genetics*, 56(3):799–810, 1995.
20. T. Niu, Z. S. Qin, X. Xu, and J. S. Liu. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. *American Journal of Human Genetics*, 70(1):157–69, 2002.
21. N. Patil, A. J. Berno, D. A. Hinds, W. A. Barrett, J. M. Doshi, C. R. Hacker, C. R. Kautzer, D. H. Lee, C. Marjoribanks, D. P. McDonough, B. T. Nguyen, M. C. Norris, J. B. Sheehan, N. Shen, D. Stern, R. P. Stokowski, D. J. Thomas, M. O. Trulson, K. R. Vyas, K. A. Frazer, S. P. Fodor, and D. R. Cox. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. *Science*, 294:1719–1723, 2001.
22. I. Pe'er, R. Shamir, and R. Sharan. Incomplete directed perfect phylogeny. In *Proc. 11th Annual Symposium on Combinaotrial Pattern Matching (CPM '00)*, pages 143–153. Springer, 2000. To appear in *SIAM Journal on Computing*.
23. R. Sachidanandam, D. Weissman, S. C. Schmidt, J. M. Kakol, L. D. Stein, G. Marth, S. Sherry, J. C. Mullikin, B. J. Mortimore, D. L. Willey, S. E. Hunt, C. G. Cole, P. C. Coggill, C. M. Rice, Z. Ning, J. Rogers, D. R. Bentley, P. Y. Kwok, E. R. Mardis, R. T. Yeh, B. Schultz, L. Cook, R. Davenport, M. Dante, L. Fulton, L. Hillier, R. H. Waterston, J. D. McPherson, B. Gilman, S. Schaffner, W. J. Van Etten, D. Reich, J. Higgins, M. J. Daly, B. Blumenstiel, J. Baldwin, N. Stange-Thomann, M. C. Zody, L. Linton, E. S. Lander, and D. Altshuler. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature*, 291:1298–2302, 2001.
24. M. A. Steel. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, 9:91–116, 1992.
25. M. Stephens, N. J. Smith, and P. Donnelly. A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics*, 68(4):978–89, 2001.