Computational Complexity of Perfect-Phylogeny-Related Haplotyping Problems

Michael Elberfeld
Institut für Theoretische Informatik
Universität zu Lübeck
D-23538 Lübeck, Germany
elberfeld@tcs.uni-luebeck.de

Till Tantau
Institut für Theoretische Informatik
Universität zu Lübeck
D-23538 Lübeck, Germany
tantau@tcs.uni-luebeck.de

Technical Report
SIIM-TR-A-08-02
Schriftenreihe der Institute für Informatik/Mathematik der
Universität zu Lübeck

June 3, 2008

Abstract

Haplotyping, also known as haplotype phase prediction, is the problem of predicting likely haplotypes based on genotype data. This problem, which has strong practical applications, can be approached using both statistical as well as combinatorial methods. While the most direct combinatorial approach, maximum parsimony, leads to NP-complete problems, the perfect phylogeny model proposed by Gusfield yields a problem, called PPH, that can be solved in polynomial (even linear) time. Even this may not be fast enough when the whole genome is studied, leading to the question of whether parallel algorithms can be used to solve the PPH problem. In the present paper we answer this question affirmatively, but we also give lower complexity bounds on its complexity. In detail, we show that the problem lies in Mod_2L, a subclass of the circuit complexity class NC^2, and is hard for logarithmic space and thus presumably not in NC^1. We also investigate variants of the PPH problem that have been studied in the literature, like the perfect path phylogeny haplotyping problem and the combined problem where a perfect phylogeny of maximal parsimony is sought, and show that some of these variants are TC^0-complete or lie in AC^0.

1 Introduction

We investigate the computational complexity of haplotype phase prediction problems. Haplotype phase prediction is an important preprocessing step in genomic disease and medical condition association studies. In these studies two groups of people are identified, where one group has a certain disease or medical condition while the other has not, and one tries to find correlations between group membership and the genomic data of the individuals in the groups. The genomic data typically consists of information about which bases are present in an individual's DNA at so-called SNP sites (single nucleotide polymorphism sites). While the DNA sequences of different individuals are mostly identical, at SNP sites there may be variations. Low-priced methods for large-scale inference of genomic data can read out, separately for each SNP site, the bases present, of which there can be two since we inherit one chromosome from our father and one from our mother. However, since the bases at different sites are determined independently, we have no information about which chromosome the base belongs to. For *homozygous sites,* where the same base is present on both chromosomes, this is not a problem, but for *heterozygous sites* this information, called the *phase* of an SNP site, is needed for accurate correlations. The idea behind *haplotype phase prediction* or just *haplotyping* is to computationally predict likely phases based on the laboratory data (which misses this information). For an individual, the genomic input data without phase information is called the *genotype* while the two predicted chromosomes are called *haplotypes*.

There are both statistical [10, 11, 17] as well as combinatorial approaches to haplotyping. The present paper will treat only combinatorial approaches, of which there are two main ones: Given a set of observed genotypes, the maximum parsimony approach [5, 6, 12, 15, 18] tries to minimize the number of different haplotypes needed to explain the genotypes. The rationale behind this approach is that mutations producing new haplotypes are rare and, thus, genotypes can typically be explained by a small set of distinct haplotypes. Unfortunately, the computational problem resulting from the maximum parsimony approach, called MH for *maximum parsimony haplotyping*, is NP-complete [25]. This is remedied by Gusfield's perfect-phylogeny-based approach [16]. Here the rationale is that mutation events producing new haplotypes can typically be arranged in a perfect phylogeny (a sort of "optimal" evolutionary tree). The resulting problem, called PPH for *perfect phylogeny haplotyping*, can be solved in polynomial time as shown in Gusfield's seminal paper [16]. It is also possible to combine these two approaches, that is, to look for a minimal set of haplotypes whose mutation events form a perfect phylogeny, but the resulting problems are – not very surprisingly – NP-complete once more [1, 26].

Due to the great practical importance of solving the PPH problem efficiently, a lot of research has been invested into finding quick algorithms for it and also for different variants. These efforts have culminated in the recent linear-time algorithms [8, 22, 23] for PPH. On the other hand, for a number of variants, in particular when missing data is involved, NP-completeness results can be obtained. (In the present paper we only consider the case where complete data is available.) These results have sparked our interest in, ideally, determining the exact computational complexity of these problems in complexity-theoretic terms. For instance, by Gusfield's result, PPH $\in$ P, but is it also hard for this class? Note that this question is closely linked to the question of whether we can find an efficient parallel algorithm.

Before we list the results obtained in the present paper, let us first describe the problems that we investigate (detailed definitions are given in the next section). The input is always a *genotype matrix*, whose rows represent individuals and whose columns represent SNP sites. An entry in this matrix encodes the measurement made for the given individual and the given SNP site. The question is always whether there exists a *haplotype matrix* with certain properties that *explains* the genotype matrix. A haplotype matrix explaining a genotype matrix has twice as many rows, namely for each individual two haplotypes, one from the father and one from the mother, and these two haplotypes taken together must explain exactly the observed genotype in the input matrix for this individual.

The *perfect phylogeny model* is an evolutionary model according to which mutations at a specific site can happen only once, in other words, there cannot be any "back-mutations." For haplotype matrices this means that there must exist an (evolutionary) tree whose nodes can be labeled with the haplotypes in the matrix in such a way that all haplotypes sharing a base at a given site form a connected subtree. In the perfect *path* phylogeny model [14] the phylogeny is restricted to be a very special kind of tree, namely a path. In the *directed* version of the perfect phylogeny model the tree is rooted and the root label is part of the input.

Problem	*Question:* Given a genotype matrix, an integer d (where applicable), and a root label (where applicable), does there exist an explaining haplotype matrix ...
MH	...that has at most d different haplotypes?
PPH	...that admits a perfect phylogeny?
DPPH	...that admits a perfect phylogeny with the given root label?
MPPH	...that admits a perfect phylogeny and has at most d different haplotypes?
PPPH	...that admits a perfect path phylogeny?
DPPPH	...that admits a perfect path phylogeny with the given root label?
MPPPH	...that admits a perfect path phylogeny and has at most d different haplotypes?

Our Results. In this paper we show that PPH is hard for logarithmic space under first-order reductions. This is the first lower bound on the complexity of this problem. We also show PPH $\in$ Mod_2L, which is a close (but not matching) upper bound on the complexity of this central problem.

We show that the PPPH problem, where the tree topology of the perfect phylogeny is restricted to a path, is (provably) easier: This problem lies in FO, which is the same as the uniform circuit class AC^0 (constant depth, unbounded fan-in, polynomial size circuits). This implies, in particular, that PPPH cannot be hard for logarithmic space – unlike PPH. To obtain this results we extend the partial order method introduced by Gramm et al. [14] for directed perfect path phylogenies to the undirected case. We also show that MPPPH is complete for $\text{TC}^0 = \text{FO(COUNT)}$.

Restricting the tree topology to a path is one way of simplifying the PPH problem. Another approach that has been studied in the literature is to restrict the number of heterozygous entries in the genotype matrices. We show that PPH is in FO when genotype matrices are restricted to contain at most two heterozygous entries per row or at most one heterozygous entry per column. In contrast, if we allow at most three heterozygous entries per row, PPH is still L-hard.

Related Work. Perfect phylogeny haplotyping was suggested by Gusfield [16]. The computational complexity of the PPH problem is quite intriguing since, at first sight, it is not even clear whether this problem is solvable in polynomial time. Gusfield showed that this is, indeed, the case and different authors soon proposed simplified algorithms with easier implementations [2, 9]. A few years later three groups independently devised linear time algorithms for the problem [8, 22, 23]. All these papers are concerned with the time complexity of PPH and all these algorithms have at least linear space complexity.

Haplotyping with perfect path phylogenies was introduced by Gramm et al. [14] in an attempt to find faster algorithms for restricted versions of PPH. For instance, Gramm et al. present a simple and fast linear-time algorithm for DPPPH and they show that the version with incomplete data is fixed parameter tractable with respect to the number of missing entries per site. These results all suggested (but did not prove) that the PPH problem is somehow "easier" than PPPH. The results of the present paper, namely that PPPH $\in$ FO while PPH is L-hard, settle this point.

Our result that MPPPH is TC^0-complete contrasts sharply with the NP-completeness of MPPH proved in [1, 26]. One might try to explain these contrasting complexities by arguing that "considering perfect *path* phylogenies rather than perfect phylogenies makes the problem trivial anyway, so this is no surprise," but this is not the case: For instance, the incomplete perfect path phylogeny problem, IPPPH, is known to be NP-complete [14] just like IPPH.

Van Iersel et al. [26] have studied the complexity of MPPH for inputs with a bounded number of heterozygous entries and they show that for certain bounds the problem is still NP-complete while for other bounds it lies in P. These results are closely mirrored by the results of the present paper, only the classes we consider are much smaller: The basic PPH problem is L-hard and so are some restricted versions, while other restricted versions lie in FO. It turns out that, despite the different proof techniques, the restrictions that make MPPH lie in P generally also make PPH lie in FO, while restrictions that cause MPPH to be NP-hard also cause PPH to be L-hard.

Organization of This Paper. After the following preliminaries section, Section 3 presents our results on PPH, including the variants of PPH with restricted inputs matrices. In Section 4 we present the results on PPPH and the maximum parsimony variant MPPPH.

2 Basic Notations and Definitions

2.1 Haplotypes, Genotypes, Perfect Phylogenies, Induced Sets

Conceptually, a haplotype describes genetic information from a single chromosome. When SNP sites are used for this purpose, a haplotype is a sequence of the bases A, C, G and T. In the human genome, at any given SNP site at most two different bases can be observed in almost all cases (namely an original base and a mutated version). Since these two bases are known and fixed for a given site, we can encode one base with 0 and the other with 1 (for this particular site). For example, the haplotypes AAGC and TATC might be encoded by 0110 and 1100. A genotype combines two haplotypes by joining their entries. For example, the haplotypes AAGC and TATC lead to the genotype $\{A,T\}\{A\}\{G,T\}\{C\}$ and so do the two haplotypes AATC and TAGC. Given a genotype, we call sites with only one observed base *homozygous* and sites with two bases *heterozygous*. It is customary to simplify the representation of genotypes by encoding a homozygous entry by 0 or 1 (depending on the single base present) and heterozygous entries with the number 2. Thus, we encode the above genotype by 2120.

Formally, a *haplotype* is a vector of 0's and 1's. A *genotype* is a vector of 0's, 1's and 2's. Given a vector c, let $c[i]$ denote the ith component. Two haplotypes $h_1, h_2 \in \{0,1\}^n$ *explain* a genotype $g \in \{0,1,2\}^n$ if for each $i \in \{1,\dots,n\}$ we have $g[i] = h_1[i] = h_2[i]$ whenever $g[i] \in \{0,1\}$ and $h_1[i] \neq h_2[i]$ whenever $g[i] = 2$. To examine multiple genotypes or haplotypes, we arrange them in matrices. The rows of a *haplotype matrix* are haplotypes and the rows of a *genotype matrix* are genotypes. We call a column of a matrix *polymorphic* if it contains both 0-entries and 1-entries or a 2-entry. We say that a $2n \times m$ haplotype matrix B *explains* an $n \times m$ genotype matrix A if for each $i \in \{1,\dots,n\}$ the rows $2i-1$ and $2i$ of B explain the row i of A.

Perfect phylogenies for haplotype and genotype matrices are defined as follows:

Definition 2.1. A haplotype matrix B *admits a perfect phylogeny* if there exists a rooted tree T_B, called *a perfect phylogeny for B*, such that:

1. Each row of B labels exactly one node of T_B.
2. Each column of B labels exactly one edge of T_B.
3. Each edge of T_B is labeled by at least one column of B.
4. For every two rows h_1 and h_2 of B and every column i, we have $h_1[i] \neq h_2[i]$ if, and only if, i lies on the path from h_1 to h_2 in T_B.

A genotype matrix A *admits a perfect phylogeny* if there exists a haplotype matrix B that explains A and admits a perfect phylogeny.

We say that T_B is a *perfect path phylogeny* if the topology of T_B is a path, that is, T_B consists of at most two disjoint branches emanating from the root. If the root of T_B is labelled with a haplotype given beforehand, we call T_B *directed.* Since the roles of 0's and 1's can be exchanged individually for each column, we will always require the given haplotype to be the all-0-haplotype. Formally, we say that a haplotype matrix B *admits a directed perfect (path) phylogeny* if there exists a perfect (path) phylogeny as in Definition 2.1 with the root label 0^n.

The *four gamete property* is a well-known alternative characterization of perfect phylogenies: A haplotype matrix admits a perfect phylogeny if, and only if, no column pair contains the submatrix $\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$. By the four gamete property it is important to know for each pair of columns which combinations of 0's and 1's are (or must be) present in the pair. Following Eskin, Halperin, and Karp [9] we call these combinations the *induced set* of the columns. Formally, given a haplotype matrix B and a pair of columns (c,c') the *induced set* $\operatorname{ind}_B(c,c') \subseteq \{00,01,10,11\}$ contains all bitstrings $ab \in \{00,01,10,11\}$ such that there is a row r in B such that the entry in column c is a and the entry in column c' is b. For a genotype matrix A and two columns, the *induced set* $\operatorname{ind}_A(c,c') \subseteq \{00,01,10,11\}$ is the intersection of all $\operatorname{ind}_B(c,c')$ where B is any haplotype matrix that explains A. This means, for instance, that the induce of the two columns of the genotype matrix $\begin{bmatrix} 0 & 1 \\ 2 & 0 \end{bmatrix}$ is $\{01,00,10\}$, the induce of $\begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}$ is $\{00,10,11\}$, and the induce of $\begin{bmatrix} 0 & 0 \\ 2 & 2 \end{bmatrix}$ is $\{00\}$.

For a genotype with at least two heterozygous entries there exist multiple pairs of explaining haplotypes. If a genotype g contains $[2\ 2]$ in columns c and c', then two explaining haplotypes for g contain either $\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$ or $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ in c and c'. In the first case we say that g *is resolved equally in* (c,c') and in the second case we say that g *is resolved unequally in* (c,c'). By the four gamete property, when a genotype matrix admits a perfect phylogeny, for each column pair all genotypes are resolved either equally or unequally. The resolution of a column pair is often, but not always, determined by the induce: In order to obtain a perfect phylogeny, a column pair that induces 00 and 11 must be resolved equally and a column pair that induces 01 and 10 must be resolved unequally.

2.2 Complexity Classes, Circuit Classes, Descriptive Complexity Theory

The classes L, P, and NP denote logarithmic space, polynomial time, and nondeterministic polynomial time, respectively. The class $\mathrm{Mod}_2\mathrm{L}$ contains all languages L for which there exists a nondeterministic logspace Turing machine such that $x \in L$ if, and only if, the number of accepting computation paths on input x is odd. The circuit classes AC^0, TC^0, NC^1, and NC^2, all of which are assumed to be uniform in the present paper, are defined as follows: AC^0 is the class of problems that can be decided by a logspace-uniform family of constant-depth and polynomial-size circuits over and-, or- and not-gates with an unbounded fan-in. For TC^0 we may additionally use threshold gates. The class NC^1 contains all languages that can be decided by a logspace-uniform family of polynomial-size, $O(\log n)$-depth circuits over and-, or- and not-gates with bounded fan-in. For the class NC^2 the depth is $O(\log^2 n)$.

We use several notions from descriptive complexity theory, which provides equivalent characterizations of the classes AC^0 and TC^0. In descriptive complexity theory inputs are encoded as logical structures. A genotype matrix is described by the logical structure $\mathscr{A} = (I^{\mathscr{A}}, A_0^{\mathscr{A}}, A_1^{\mathscr{A}}, A_2^{\mathscr{A}}, n_r^{\mathscr{A}}, n_c^{\mathscr{A}})$ as follows: $I^{\mathscr{A}}$ is a finite set of indices for rows and columns and $n_r^{\mathscr{A}}$ and $n_c^{\mathscr{A}}$ are elements from $I^{\mathscr{A}}$ that are the maximum row and column indices, respectively. The relation $A_0^{\mathscr{A}} \subseteq I^{\mathscr{A}} \times I^{\mathscr{A}}$ indicates 0-entries, that is, $(r,c) \in A_0^{\mathscr{A}}$ holds exactly if the row r has a 0-entry in column c. The relations $A_1^{\mathscr{A}}$ and $A_2^{\mathscr{A}}$ are defined similarly, only for 1- and 2-entries. We assume that the universe $I^{\mathscr{A}}$ is ordered (we always have free access to a predicate $<$), but will not need the bit-predicate (see [19] for a discussion of its importance).

Given a formula, the set of all finite logical structures satisfying the formula (that are models of the formula) can be regarded as a language. If the formula is a first-order formula, the described language is called *first-order definable.* The class of all such languages is denoted FO and equals AC^0. For example, the formula $(\exists r,c)[r \leq n_r \wedge c \leq n_c \wedge A_2(r,c)]$ is true for genotype matrices that contain a row with a heterozygous entry and, therefore, defines the set of genotype matrices with at least one heterozygous entry. The computational power of first-order logic can be increased by adding an additional number domain and counting quantifiers. This class, which is called FO(COUNT), equals TC^0.

To describe mappings between logical structures, one can use *first-order queries,* which are tuples of defining formulas for the relations of the image structure. Since L is closed under reductions that can be described by first-order queries, we use these *first-order reductions* to prove L-hardness.

The inclusion structure between the described classes is known to be the following:

$$\mathrm{FO} = \mathrm{AC}^0 \subsetneq \mathrm{FO(COUNT)} = \mathrm{TC}^0 \subseteq \mathrm{NC}^1 \subseteq \mathrm{L} \subseteq \mathrm{Mod_2L} \subseteq \mathrm{NC}^2 \subseteq \mathrm{P} \subseteq \mathrm{NP}\,.$$

3 Complexity of Haplotyping via Perfect Phylogenies

In the present section we study the complexity of PPH as well as the variants where the number of heterozygous entries in the input is restricted. In Section 3.1 we show that PPH and its directed variant are L-hard and in $\mathrm{Mod_2L}$. Thus, both problems are in NC^2 by the inclusion $\mathrm{Mod_2L} \subseteq \mathrm{NC}^2$ shown in [3], but not in NC^1, unless $\mathrm{L} = \mathrm{NC}^1$. In Section 3.2 we additionally take restrictions into account and show that the hardness result still holds when we restrict the input to genotype matrices with at most three heterozygous entries per row. In contrast, we show that PPH is first-order definable for genotype matrices with at most two heterozygous entries per row or at most one heterozygous entry per column.

We will focus on the directed version DPPH rather then PPH since Eskin, Halperin and Karp [9] have shown that PPH reduces to DPPH via an easy construction: In each column search downward for the first homozygous entry and if it equals 1, exchange the roles of 0 and 1 in this column. Indeed, this construction is so simple that it can be implemented using a first-order query: for each homozygous entry we have to decide whether it is inverted and this depends on the value of the first homozygous entry in the same column, which in turn is easy to determine for an ordered universe (recall that we always have access to an ordering of the universe). Note that DPPH trivially reduces to PPH by adding a row with only 0-entries to the matrix. Interestingly, the path variants PPPH and DPPPH are also equivalent via first-order reductions, which is shown in Section 4, but this is a bit harder to prove.

3.1 Complexity of the PPH Problem

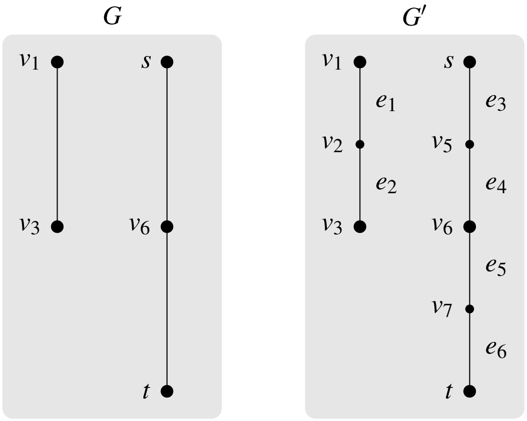
In the present section we prove two theorems on the complexity of PPH. The first gives a lower bound, namely that PPH is hard for logarithmic space under first-order reductions. The second gives an upper bound, namely PPH $\in \mathrm{Mod_2L}$. Since DPPH is first-order equivalent to PPH, the same results hold for the directed version, also. The bounds show that both problems are in NC^2, but not in NC^1 under common assumptions from complexity theory. We thank Arfst Nickelsen for hinting at the basic idea of the proof of Theorem 3.1 in a personal communication.

Theorem 3.1. PPH *is hard for* L *under first-order reduction.*

Proof. We present a first-order reduction from UGAP (the reachability problem for undirected graphs, also known as UREACH or U-s-t-CON) to the complement of PPH. This implies the claim since UGAP is L-hard via first-order reductions [19] and L is closed under complement.

The construction of a genotype matrix A from an undirected graph G with two distinct nodes s and t consists of two steps. In the first step we replace each edge in G by a path of length 2 and obtain a new graph G'. Let us call the nodes in G' that were already present in G the *old nodes* and let us call the added nodes on the midpoints of the length 2 paths the *new nodes*. The length of a path in G' between two old nodes is always even.

For the second step let n be the number of nodes and m be the number of edges in G'. We construct an $(m+4) \times (n+1)$ genotype matrix A as follows: Each edge of G' corresponds to exactly one row of A and the remaining rows are called r_1, r_2, r_3 and r_4. Each node of G' corresponds to exactly one column of A and the remaining column is called d. From now on we denote both an edge e and its corresponding row by e and similarly for a node v and the corresponding column v. The entries of A are as follows: For each edge $e = \{v,w\}$ from G' there are 2-entries in e's row in columns v, w, and d. The row r_1 is set to 1 in columns s and d, the row r_3 is set to 1 in column t and the row r_4 is set to 1 in column d. The remaining entries of A are set to 0. Note that the rows r_1 and r_2 force an equal resolution of the column pair (s,d) and that rows r_3 and r_4 force an unequal resolution of the column pair (t,d). Figure 1 shows an example of this construction. Clearly, the construction can be computed by a first-order query.



$$A$$

$$\begin{array}{c|ccccccccc}
 & v_1 & v_2 & v_3 & s & v_5 & v_6 & v_7 & t & d \\
e_1 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
e_2 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 2 \\
e_3 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 2 \\
e_4 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 0 & 2 \\
e_5 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 0 & 2 \\
e_6 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \\
r_1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
r_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
r_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
r_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}$$

Figure 1: This figure shows an example of the reduction from UGAP to PPH. First, the graph G' is constructed based on the graph G by adding new nodes (depicted a bit smaller). Then the genotype matrix A is build from G'. The first eight columns of A correspond to the nodes in G' and the first six rows of A correspond to the edges in G'.

We now prove the following claim: *There exists a path between s and t in G if, and only if, A does not admit a perfect phylogeny.* Proving this claim proves the theorem.

First, we assume that there is a path between s and t in G and, hence, there is a path between s and t in G'. Let $v_1, \ldots, v_l$ be the nodes and $e_1, \ldots, e_{l-1}$ be the edges on such a path in G'. Then $v_1 = s$, $v_l = t$, and $e_i = \{v_i, v_{i+1}\}$ for each $i \in \{1, \ldots, l-1\}$. The construction of the matrix A enforces certain resolutions for some column pairs in A. For each column pair (v_i, v_{i+1}) we have $\{01, 10\} \subseteq \operatorname{ind}_A(v_i, v_{i+1})$ and therefore it must be resolved unequally. As already stated, (v_1, d) must be resolved equally and (v_l, d) must be resolved unequally. We assume, for the sake of contradiction, that there exists a haplotype matrix B that explains A and admits a perfect phylogeny. Consider the genotype of edge e_1, which is heterozygous at positions v_1, v_2 and d. The enforced resolutions cause the explaining haplotypes for e_1 at positions v_1, v_2, and d to be 010 and 101. Thus, the column pair (v_2, d) is resolved unequally. Next, consider the genotype e_2, which is heterozygous at positions v_2, v_3 and d. Similar to the previous case we see that the explaining haplotypes for e_2 at positions v_2, v_3, and d are 011 and 100. Thus, (v_3, d) is resolved equally. By induction we obtain that (v_i, d) must be resolved equally if i is odd and unequally, otherwise. Since the length of each path from v_1 to v_l in G' is even, l is odd and, therefore, (v_l, d) must be resolved equally, a contradiction.

Conversely, assume that there is no path between s and t in G and, therefore, no path between s and t in G'. Then G' consists of at least two components, one containing s and the other containing t. We construct a haplotype matrix B that explains A by considering each component of G' separately and constructing explaining haplotypes for the genotypes of the edges in the component. Let C_s be the component of G' that contains s and let e be an arbitrary edge from C_s. One node of e, call it v, is an old node with an even distance from s and the other node, call it w, is a new node with an odd distance from s. We construct two haplotypes for e as follows: First note that the entries of the haplotypes are predetermined at the homozygous positions of e. At the heterozygous positions v, w and d we set one haplotype to 010 and the other to 101. This procedure enforces that for each node v in C_s the column pair (v, d) is resolved equally if the distance from s to v is even and unequally otherwise. For the component C_t that contains t we construct the haplotypes slightly differently. Let $e = \{v, w\}$ be an edge from C_t

and v the node with even distance to t. At the heterozygous positions v, w and d we set one haplotype to 100 and the other to 011. Then for each node v in C_t the column pair (v,d) is resolved unequally if the distance from t to v is even and equally otherwise. For components that contains neither s nor t, we construct the haplotypes similarly but with the role of s played by any old node in the component. Finally, we insert the homozygous genotypes r_1, r_2, r_3, and r_4 as haplotypes into B.

The haplotype matrix B admits a perfect phylogeny for the following reason: Let v be a node from C_s. If the distance from v to s is even, then (v,d) is resolved equally and B does not contain the submatrix $[1\ 0]$ in v and d. If the distance is odd, then, due to the unequal resolution of (v,d), B does not contain the submatrix $[1\ 1]$ in v and d. For each node from C_t this property holds for similar reasons with the roles of odd and even distances exchanged. For each node v from a component C that contains neither s nor t, either $[1\ 0]$ or $[1\ 1]$ is also not contained in v and d. Finally, observe that for two nodes v and w the column pair (v,w) is resolved unequally and does not contain the submatrix $[1\ 1]$ in B. Thus B admits a perfect phylogeny since the induced set of each column pair contains at most three elements. □

Theorem 3.2. PPH *is in* Mod_2L.

Proof. PPH can be logspace-many-one reduced to solving systems of linear equations over $\mathbb{Z}/2\mathbb{Z}$. This reduction is implicit in the construction of Theorem 1 of the paper by Eskin, Halperin, and Karp [9]. Solving systems of linear equations over $\mathbb{Z}/2\mathbb{Z}$ is in Mod_2L as shown in [3] and since Mod_2L is closed under logspace-many-one reductions, we get the claim. □

3.2 Complexity of PPH for Restricted Instances

How do restrictions on the number of heterozygous sites influence the complexity of PPH? This question will be addressed in the present section. Following Sharan, Halldórsson, and Istrail [25] we say that a genotype matrix is (k,l)*-bounded* if each row contains at most k and each column at most l heterozygous entries. We use a star to indicate that a parameter is not bounded. We parametrize problems in the same way, so $\text{PPH}(3,*)$ denotes the set of all genotype matrices with at most three heterozygous entries per row that admit a perfect phylogeny.

In the literature (k,l)-bounded variants were first studied for the NP-complete problem MH. The hope was to find restrictions that holds in practice and that make the problem tractable. In different papers parameters k and l have been determined such that $\text{MH}(k,l)$ is either efficiently solvable or NP-complete [4, 20, 21, 25, 26]. Bounded variants of MPPH have also been studied and the main results are the same as for the corresponding variants of MH.

We study the complexity of (k,l)-bounded variants of PPH. Our results, summarized in Theorem 3.3, show strong similarities to the complexity of bounded variants of MH and MPPH, but one complexity level further down. We show that $\text{PPH}(3,*)$ is L-hard; and it is known [26] that $\text{MH}(3,*)$ and $\text{MPPH}(3,*)$ are NP-complete. We show $\text{PPH}(2,*) \in \text{FO}$ and $\text{PPH}(*,1) \in \text{FO}$; and it is known [4, 21, 26] that $\text{MH}(2,*), \text{MPPH}(2,*) \in \text{P}$ and $\text{MH}(*,1), \text{MPPH}(*,1) \in \text{P}$. We do not know the complexity of $\text{PPH}(*,2)$; and the complexity $\text{MH}(*,2)$ and $\text{MPPH}(*,2)$ are also open.

Theorem 3.3.

1. $\text{PPH}(3,*)$ *is* L*-hard.*
2. $\text{PPH}(2,*)$ *is first-order definable.*
3. $\text{PPH}(*,1)$ *is first-order definable.*

Proof. We first show that $\text{PPH}(3,*)$ is L-hard. For this, just note that in our proof of Theorem 3.1 we reduce to $(3,*)$-bounded genotype matrices. Thus $\text{PPH}(3,*)$ is L-hard.

For the second statement, $\text{PPH}(2,*) \in \text{FO}$, we first characterize $(2,*)$-bounded genotype matrices that admit a perfect phylogenies. Haplotype matrices that admit a perfect phylogeny can easily be characterized by the property that all induced sets contain at most three elements. The following claim states that this also holds for $(2,*)$-bounded genotype matrices.

Claim. *Let A be a $(2,*)$-bounded genotype matrix. Then A admits a perfect phylogeny if, and only if, $|\mathrm{ind}_A(c,c')| \leq 3$ holds for every column pair (c,c').*

Proof. The only-if-part is trivial. For the if-part let A be a $(2,*)$-bounded genotype matrix and let $|\mathrm{ind}_A(c,c')| \leq 3$ for every column pair (c,c'). We construct a haplotype matrix B for A as follows: For genotypes with at most one heterozygous entry, the explaining haplotypes are completely determined. For each genotype g with two heterozygous positions c and c', we resolve g in (c,c') equally if $\{00,11\} \subseteq \mathrm{ind}_A(c,c')$ and unequally otherwise. The matrix B admits a perfect phylogeny since for each column pair either the pair does not contain the submatrix $[2\ 2]$ in A or all genotypes are resolved consistently with the corresponding induced set of A. □

By the above characterization, a first-order formula for $\text{PPH}(2,*)$ can be constructed as follows: For each column pair it tests whether the induced set does not contain four distinct elements. To decide whether 00, 01, 10 or 11 is contained in an induced set, the formula tests whether there is a genotype with explaining haplotypes that necessarily induce this pair.

For the last statement, $\text{PPH}(*,1) \in \text{FO}$, we also first establish a characterization. An interesting aspect of this characterization is that it involves perfect *path* phylogenies, even though we deal only with perfect phylogenies in the present section.

Claim. *Let A be a $(*,1)$-bounded genotype matrix. For a row r let A_r denote the submatrix of A that consists of the columns where r is heterozygous. Then A admits a perfect phylogeny if, and only if, (a) for every column pair (c,c') we have $|\mathrm{ind}_A(c,c')| \leq 3$ and (b) for every row r the matrix A_r admits a perfect path phylogeny.*

Proof. For the only-if-part let A be a genotype matrix and B a haplotype matrix that explains A and admits a perfect phylogeny. Trivially, each column pair in A induces at most three elements. Let T_B be a perfect phylogeny for B and r be a row of A. Due to item 4 of Definition 2.1, the path in T_B between the two haplotypes that explain r is labeled exactly by the heterozygous columns of r. In order to construct a haplotype matrix B_r for A_r, we delete the columns not that do not lie on this path both from A and B, we delete the corresponding edge labels, and we contract unlabeled edges. In this way, we can construct a haplotype matrix and a perfect path phylogeny for A_r.

For the if-part let A be a $(*,1)$-bounded genotype matrix such that $|\mathrm{ind}_A(c,c')| \leq 3$ holds for every column pair (c,c') and there exists a haplotype matrix B_r that explains A_r and admits a perfect path phylogeny for every row r. We assemble a haplotype matrix B for A from the matrices B_r and the matrix A in a columnwise fashion. Let c be an arbitrary column. If A has a heterozygous entry in c, we take c from a matrix B_r. In this case, there is exactly one matrix B_r that contains c. If A does not have a heterozygous entry in c, we take c from the matrix A and double each entry since B has twice the number of rows of A. To prove that B admits a perfect phylogeny, let c and c' be two columns. If c and c' are contained in the same matrix B_r, we have $|\mathrm{ind}_{B_r}(c,c')| \leq 3$ and therefore $|\mathrm{ind}_B(c,c')| \leq 3$. Otherwise, c and c' do not contain the submatrix $[2\ 2]$ in A and therefore we have $\mathrm{ind}_A(c,c') = \mathrm{ind}_B(c,c')$. □

By the above claim, a formula for $\text{PPH}(*,1)$ can first test whether the size of the induced set of each column pair is at most three. Then the formula tests whether for each row r the matrix A_r admits a perfect path phylogeny. As we will see in the following section, this property can be tested using a first-order formula, see Theorem 4.4, although the columns over which we quantify have to be restricted to those with a heterozygous entry in row r. □

4 Complexity of Haplotyping via Perfect Path Phylogenies

In the present section we consider perfect *path* phylogenies, a problem variant first advocated in Gramm et al. [14] and later studied in [13]. In the first of these papers it is shown that PPPH is solvable in linear time. We show that it is first-order definable and, therefore, in AC^0. To obtain this result, we first prove DPPPH $\in$ FO and then reduce PPPH to DPPPH by a first-order reduction. While this reduction is similar to the reduction from PPH to DPPH in Section 3, the correctness proof for the path variant differs. Bafna et al. [1] have shown the NP-completeness of MPPH. For the path variant MPPPH we show that it can be described by a first-order formula with counting quantifiers and is, therefore, in TC^0. In addition, we prove that MPPPH is hard for TC^0.

4.1 Complexity of the Basic Decision Problem

In the present section we show that the set of all genotype matrices admitting a perfect path phylogeny can be described using a first-order formula; in other words, we show PPPH $\in$ FO, see Theorem 4.4. In order to prove this, we first show that the simpler problem DPPPH lies in FO and then show that PPPH can be first-order reduced to its directed version.

We start with some notations and a characterization for DPPPH from the literature. Then we present an alternative characterization that can be formalized with first-order logic and a first-order reduction from PPPH to DPPPH. While the construction of this reduction is easy, its correctness proof is more elaborated. Finally, we conclude that PPPH is first-order definable since the class of first-order definable problems is closed under first-order reduction.

We define a partial order on the columns of a genotype matrix as follows: Let $1 \succ 2 \succ 0$. For two columns c and c' we have $c \succeq c'$ if $c[i] \succeq c'[i]$ for each row i. We say that two columns c and c' are *comparable* if $c \succeq c'$ or $c' \succeq c$. If $c \succeq c'$ and $c \neq c'$, then we say that c *dominates* c'. In the following let C be a set of columns. A subset of C is called a *(anti)chain* if its elements are pairwise (in)comparable. An antichain $C' \subseteq C$ is *maximal* if it is not properly contained in any other antichain. A maximal antichain C' of size i is the *highest maximal antichain of size i* if there is no other antichain of size (exactly) i with an element that dominates an element from C'. For a set C of columns there exists at most one highest maximal antichain of size i, which we denote by $\mathrm{hma}_i(C)$. Let $\mathrm{hma}_i(C) = \emptyset$ if there is no such set.

We call two columns *separable* if each column has a 0-entry in the rows where the other has a 1-entry. Following [14] we say that a column set C *has the ppp-property* if there exist two (possibly empty) chains C_1 and C_2 that cover C, so that their maximal elements (if they exist) are separable. We call (C_1, C_2) a *ppp-cover* of C. The following fact characterizes DPPPH.

Fact 4.1 (Gramm et al. [14])**.** *A genotype matrix A admits a directed perfect path phylogeny if, and only if, the set of columns of A has the ppp-property.*

The above characterization does not readily yield a first-order description of DPPPH since we cannot quantify over chains (a second-order quantifier would be needed, lifting the complexity up to the polynomial hierarchy). What we need is a more "element-oriented" characterization such as the one given by the following lemma.

Lemma 4.2. *A column set C has the ppp-property if, and only if, the width of C is at most* 2 *and one of the following statements is true:*

1. $\mathrm{hma}_1(C) = \{c^*\}$ *and* $\mathrm{hma}_2(C) = \emptyset$.
2. $\mathrm{hma}_1(C) = \emptyset$, $\mathrm{hma}_2(C) = \{c_1, c_2\}$, *and c_1 and c_2 are separable.*
3. $\mathrm{hma}_1(C) = \{c^*\}$, $\mathrm{hma}_2(C) = \{c_1, c_2\}$, *and c^* and c_1 are separable or c^* and c_2 are separable.*

Proof. For the only-if-part, let C be a column set that has the ppp-property. Therefore, C can be covered with at most two chains. By Dilworth's theorem [7], which states that the width of a partial order equals the minimum number of chains to cover the order, the width of C is at most 2. This implies that the sets $\mathrm{hma}_k(C)$ are empty for all $k \geq 3$. We make a case distinction, depending on whether $\mathrm{hma}_1(C)$ or $\mathrm{hma}_2(C)$ or both are non-empty.

If we assume $\mathrm{hma}_1(C) = \{c^*\}$ and $\mathrm{hma}_2(C) = \emptyset$, then we are done since the first statement is satisfied. For the second case we assume $\mathrm{hma}_1(C) = \emptyset$ and $\mathrm{hma}_2(C) = \{c_1, c_2\}$. Let (C_1, C_2) be a ppp-cover of C. Since c_1 and c_2 are incomparable, they lie on different chains. Without loss of generality we assume $c_1 \in C_1$ and $c_2 \in C_2$. One can see that c_1 is the maximal element of C_1 and that c_2 is the maximal element of C_2. Since (C_1, C_2) is a ppp-cover, c_1 and c_2 are separable. For the remaining case we assume $\mathrm{hma}_1(C) = \{c^*\}$ and $\mathrm{hma}_2(C) = \{c_1, c_2\}$. Again, let (C_1, C_2) be a ppp-cover of C. Since $\{c^*\}$ dominates all elements of C, it is the maximal element of either C_1 or C_2. Without loss of generality we assume $c^* \in C_1$. The columns c_1 and c_2 lie in different chains and we first assume $c_1 \in C_1$ and $c_2 \in C_2$. Then the columns c^* and c_2 are separable for the following reason: Let c be the maximal element of C_2. Since $c_2 \preceq c$ and c and c^* are separable, we know that c_2 and c^* are separable. If we assume $c_2 \in C_1$ and $c_1 \in C_2$, it follows that c_1 and c^* separable.

For the if-part, assume that the width of C is at most 2 and one of the three statements holds. First, if $\mathrm{hma}_1(C) = \{c^*\}$ and $\mathrm{hma}_2(C) = \emptyset$, then C is a chain and, therefore, $(C, \emptyset)$ is a ppp-cover of C. Second, let $\mathrm{hma}_1(C) = \emptyset$, $\mathrm{hma}_2(C) = \{c_1, c_2\}$, and c_1 and c_2 be separable. Since the width of C is 2, the set C can be covered by two chains C_1 and C_2. Again, c_1 and c_2 are the maximal elements of the chains and, therefore, the maximal elements of C_1 and C_2 are separable. Hence (C_1, C_2) is a ppp-cover of C. Third, let $\mathrm{hma}_1(C) = \{c^*\}$, $\mathrm{hma}_2(C) = \{c_1, c_2\}$, and c^* and c_1 be separable; for the case where c^* and c_2 are separable, an analogue argument holds. Let C_1 and C_2 be two chains that cover C and assume $c_1 \in C_1$ and $c_2 \in C_2$. The column c^* is the maximal element of either C_1 or C_2. In order to construct a ppp-cover for C, alter C_1 and C_2 as follow: If c^* is in C_1, we move it to C_2. If c_1 is the maximal element of C_1, then (C_1, C_2) is a ppp-cover of C. If this is not the case, we move each element of C_1 that dominates c_1 to C_2. This yields a further chain cover of C since the columns that dominate c_1 form a chain and also dominate c_2. Now, c_1 is the maximal element of C_1 and c^* is the maximal element of C_2. □

Theorem 4.3. DPPPH *is first-order definable.*

Proof. It suffices to show that the characterization of Lemma 4.2 can be tested using a first-order formula. For this, first note that a genotype matrix can contain identical columns while a set of columns contains each column only once. In order to decide properties about the column set of a genotype matrix, our formula considers only the leftmost column of multiple identical columns. Besides this, the formula works as follows: First, it tests whether the width of C is at most 2. For this we check whether there are three pairwise incomparable columns in C. Since the partial order $\succeq$ is first-order definable, the first part of the characterization can be described with a first-order formula. The second part of the formula tests whether one of the three statements of Lemma 4.2 is satisfied. These statements describe properties of the highest maximal antichains, which can be formalized with first-order logic as follows: For a column c^*, we have $\mathrm{hma}_1(C) = \{c^*\}$ if c^* dominates every other column. For columns c_1 and c_2, we have $\mathrm{hma}_2(C) = \{c_1, c_2\}$ if they are incomparable, there is no column c such that c, c_1 and c_2 are pairwise incomparable and there is no antichain of size two with a column that dominates c_1 or c_2. Since it is first-order definable whether two columns are separable, the whole characterization is first-order definable. □

Theorem 4.4. PPPH *is first-order definable.*

Proof. We present a first-order reduction from PPPH to DPPPH $\in$ FO. The reduction consists of two steps: The first is the reduction from the beginning of Section 3, which we already used to reduce PPH to DPPH. Recall that this reduction finds for each column of the genotype matrix the first non-2-entry and, if this entry is a 1-entry, exchanges the meaning of 0-entries and 1-entries for this column. In the second step we consider each column c and set all entries in c to 0 whenever there is a column c' with a smaller index that is identical to c. Since both steps are first-order queries, the whole construction is a first-order query. Note that after this construction each pair of columns induces 00 and there are no equal polymorphic columns. The next two claims imply that the construction reduces PPPH to DPPPH.

Claim. *Let A be a genotype matrix and let A'' arise from A by the described reduction. Then $A \in$ PPPH if, and only if, $A'' \in$ PPPH.*

Proof. Just as the reduction consists of two steps, we prove the claim in two steps. Let A be the initial genotype matrix, let A' be the matrix after the first step, and let A'' be the matrix after the second step. We have $A \in$ PPPH if, and only if, $A' \in$ PPPH since a perfect path phylogeny for A can be transformed into a perfect path phylogeny for A' by inverting columns. It remains to prove $A' \in$ PPPH if, and only if, $A'' \in$ PPPH. Let B' be a haplotype matrix that explains A' and admits a perfect path phylogeny. In order to construct a haplotype matrix B'' for A'', we copy the columns that are not set to 0 and set each entry in the remaining columns to 0. A new perfect path phylogeny arises by displacing the columns that are set to 0 such that they do not lie on a path between two haplotypes. Conversely, let B'' be a haplotype matrix that explains A'' and admits a perfect path phylogeny. In order to obtain a haplotype matrix B' for A', we replace the columns that have been set to 0 by values in the unchanged column. We obtain a perfect path phylogeny by labeling edges multiple times. □

(Note that the matrix modifications described in the above proof does not alter the number of haplotypes. This additional property will be used in Section 4.2 to identify the complexity of MPPPH.)

The second claim is a property that is well-known for perfect phylogeny haplotyping [9]. We show that it also holds for the path variant.

Claim. *Let A be a genotype matrix such that for each column pair (c,c') we have $\{00\} \subseteq \mathrm{ind}_A(c,c')$. Then $A \in$ PPPH if, and only if, $A \in$ DPPPH.*

Proof. For the first direction, just note that every directed perfect path phylogeny is also a perfect path phylogeny. We prove the other direction by contraposition. Let A be a genotype matrix that does not admit a directed perfect path phylogeny. By Fact 4.1, we know that the column set C of A does not have the ppp-property and Lemma 4.2 implies that the width of C is at least 3 or the statements 1, 2 and 3 in Lemma 4.2 are not satisfied. If the width of C is at least 3, then there exist three pairwise incomparable columns c, c', and c''. Incomparable columns induce 01 and 10 and every column pair induces 00 by assumption. Thus, the column pairs (c,c'), (c',c''), and (c,c'') all induce 00, 01, and 10. No perfect path phylogeny exists for these induces. To see this, one can test each possible path for c, c' and c'' or use necessary properties for three columns that form a perfect path phylogeny from [24].

It remains to argue that if (a) the width of C is at most two and (b) neither statement 1, nor 2, nor 3 from Lemma 4.2 is satisfied, then no perfect path phylogeny exists. We make a case distinction depending on whether $\mathrm{hma}_1(C)$ or $\mathrm{hma}_2(C)$ or both are nonempty. First, $\mathrm{hma}_1(C) = \{c^*\}$ and $\mathrm{hma}_2(C) = \emptyset$ is not the case by (b). Second, consider the case $\mathrm{hma}_1(C) = \emptyset$ and $\mathrm{hma}_2(C) = \{c_1, c_2\}$. Again by (b), the columns c_1 and c_2 are not separable and therefore induce 11. Also they induce 01 and 10 since they are incomparable. Finally, we know that (c_1,c_2) induces 00 and therefore the size of its induced set is 4. For the remaining case let $\mathrm{hma}_1(C) = \{c^*\}$ and $\mathrm{hma}_2(C) = \{c_1, c_2\}$. By assumption (b) both c^* and c_1 and c^* and c_2 are not separable and therefore induce 11. Since c^* dominates c_1 and c_2, both (c^*,c_1) and (c^*,c_2) induce 10. Furthermore, we know that the columns c_1 and c_2 are incomparable and therefore induce 01 and 10. Finally, we obtain $\{00,01,10\} \subseteq \mathrm{ind}_A(c_1,c_2)$, $\{00,10,11\} \subseteq \mathrm{ind}_A(c^*,c_1)$ and $\{00,10,11\} \subseteq \mathrm{ind}_A(c^*,c_2)$. Again, no perfect path phylogeny exists for these induces. □

By the above two claims, the reduction described at the beginning of this proof is correct and we already argued that it is a first-order reduction. Thus, PPPH first-order reduces to DPPPH $\in$ FO, which proves the claim. □

4.2 Combining Perfect Path Phylogenies and Maximum Parsimony

In the present section we prove that MPPPH is TC^0-complete, in stark contrast to the fact that MPPH is NP-complete.

Theorem 4.5. MPPPH *is* TC^0*-complete.*

Proof. First, we show that MPPPH has nearly the same complexity as PPPH, we only need to add counting quantifiers. This implies MPPPH $\in \mathrm{TC}^0$. As in other proofs, we start with a characterizing claim. Later on we argue that this characterization can be tested using a first-order formula with counting quantifiers.

Claim. *Let A be a genotype matrix with $\{00\} \subseteq \mathrm{ind}_A(c,c')$ for each column pair (c,c') and no duplicate polymorphic columns. Let m be the number of polymorphic columns in A. Then every haplotype matrix that explains A and admits a perfect path phylogeny contains exactly $m+1$ different haplotypes.*

Proof. Let A and m be as above, let B be a haplotype matrix that explains A, and let T_B be a perfect path phylogeny for B. We first prove that B contains at least $m+1$ different haplotypes, then we prove that B contains at most $m+1$ different haplotypes.

Suppose, for the sake of contradiction, that B contains $d < m+1$ different haplotypes. Let $h_1, \dots, h_d$ be the sequence of node labels on the path T_B. Note that these haplotypes $h_1, \dots, h_d$ are exactly the d different haplotypes from B. Each one of the m polymorphic columns of B must label one of the $d-1 < m$ edges of T_B, so two difference polymorphic columns c and c' must label the same edge $\{h_i, h_{i+1}\}$. The haplotypes $h_1, \dots, h_i$ have the same value in column c, which we denote by a; and the haplotypes $h_{i+1}, \dots, h_d$ have the inverted value $1-a$. The same holds for the column c' with values b and $1-b$. Since each column pair in A induces 00, there exists a haplotype with a 0-entry in both columns c and c'. Thus, either $a = b = 0$ and $1-a = 1-b = 1$ or $a = b = 1$ and $1-a = 1-b = 0$. Consequently, we know that for each haplotype in T_B the entries of c and c' are equal. Hence, c and c' are identical columns of B and, therefore, of A, a contradiction.

Now assume that B contains $d > m+1$ different haplotypes. Again, let $h_1, \dots, h_d$ be the sequence of different haplotypes in T_B. First note that only polymorphic columns occur on the path from h_1 to h_d and since there are only m polymorphic columns, there exists an i such that no column labels the path from h_i to h_{i+1}. Thus, at least one edge is unlabeled, and T_B is no perfect path phylogeny. □

From the above claim we can conclude that a genotype matrix A with $\{00\} \subseteq \mathrm{ind}_A(c,c')$ for each column pair (c,c') and no equal polymorphic columns has the property $(A,d) \in$ MPPPH if, and only if, $A \in$ PPPH and d is greater than the number of polymorphic columns of A. We know already that PPPH $\in$ FO. Thus, in light of the observation made in the proof of Theorem 4.4 that the reduction does not change the number of haplotypes, all that remains to be shown is that the number of polymorphic columns of A can be counted using a counting quantifier. However, this is clearly the case since being a polymorphic column is a first-order property.

Next, we prove the TC^0-hardness of MPPPH.

Claim. MPPPH *is* TC^0*-hard under* AC^0*-reductions.*

Proof. We prove the claim via an AC^0-reduction from the TC^0-complete problem MAJORITY, where the input is a binary string $x = x_1, \dots, x_n$ and the question is whether at least half of the input bits in x are 1. We construct an $(n+1) \times n$ genotype matrix A as follows: If $x_i = 0$, we set the ith genotype in A to $1^i 0^{n-i}$ and, otherwise, we set it to 0^n. The last genotype is always set to 0^n. Since A admits a perfect path phylogeny, one can easily verify that at least half of the input bits in x are 1 if, and only if, there exists a perfect path phylogeny with at most $n - \lceil \frac{n}{2} \rceil + 1$ different haplotypes for A. □

Altogether, we obtain the TC^0-completeness of MPPPH.

□

5 Conclusion

The three main results of the present paper are that (a) the complexity of PPH lies between L and Mod_2L, (b) while PPPH lies in AC^0 and MPPPH is TC^0-complete, and (c) restricted variants of PPH are either L-hard or they lie in AC^0. Concerning the latter results, the complexity of a few restricted variants is still open. In particular, what is the complexity of $\text{PPH}(3,2)$?

A much broader, still largely open research field is the complexity of these problems when data may be missing. Typically, the resulting problems are NP-complete, so we need to look for approximation algorithms, fixed-parameter algorithms, or moderately exponential time algorithms. Specialized results are known in this context, but there are still only very few precise complexity-theoretic results in this setting.

References

[1] V. Bafna, D. Gusfield, S. Hannenhalli, and S. Yooseph. A note on efficient computation of haplotypes via perfect phylogeny. *Journal of Computational Biology*, 11(5):858–866, 2004.

[2] V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogeny: A direct approach. *Journal of Computational Biology*, 10(3–4):323–340, 2003.

[3] G. Buntrock, C. Damm, U. Hertrampf, and C. Meinel. Structure and importance of logspace-MOD-classes. *Mathematical Systems Theory*, 25(3):223–237, 1992.

[4] R. Cilibrasi, L. van Iersel, S. Kelk, and J. Tromp. On the complexity of several haplotyping problems. In *Proceedings of the 5th International Workshop on Algorithms in Bioinformatics (WABI 2005)*, volume 3692 of *Lecture Notes in Computer Science*, pages 128–139. Springer, 2005.

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

[6] M. Daly, J. Rioux, S. Schaffner, T. Hudson, and E. Ladner. High-resolution haplotype structure in the human genome. *Nature Genetics*, 29:229–232, 2001.

[7] R. P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51(1):161–166, 1950.

[8] Z. Ding, V. Filkov, and D. Gusfield. A linear-time algorithm for the perfect phylogeny haplotyping (PPH) problem. *Journal of Computational Biology*, 13(2):522–553, 2006.

[9] E. Eskin, E. Halperin, and R. M. Karp. Efficient reconstruction of haplotype structure via perfect phylogeny. *Journal of Bioinformatics and Computational Biology*, 1(1):1–20, 2003.

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

[11] D. Fallin and N. Schork. Accuracy of haplotype frequency estimation for biallelic loci via the expectation-maximation algorithm for unphased diploid genotype data. *American Journal of Human Genetics*, 67:947–959, 2000.

[12] L. Friss, R. Hudson, A. Bartoszewicz, J. Wall, T. Donfalk, and A. Di Rienzo. Gene conversion and differential population histories may explain the contrast between polymorphism and linkage disequilibrium levels. *American Journal of Human Genetics*, 69:831–843, 2001.

[13] J. Gramm, T. Hartman, T. Nierhoff, R. Sharan, and T. Tantau. On the complexity of snp block partitioning under the perfect phylogeny model. *Discrete Mathematics*, 2008. to appear, doi:10.1016/j.disc.2008.04.002.

[14] J. Gramm, T. Nierhoff, R. Sharan, and T. Tantau. Haplotyping with missing data via perfect path phylogenies. *Discrete and Applied Mathematics*, 155(6–7):788–805, 2007.

[15] D. Gusfield. Inference of haplotypes from samples of diploid populations: complexity and algorithms. *Journal of Computational Biology*, 8(3):305–23, 2001.

[16] D. Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In *Proceedings of the Sixth Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 166–175. ACM Press, 2002.

[17] M. Hawley and K. Kidd. Haplo: A program using the EM algorithm to estimate the frequency of multi-site haplotypes. *Journal of Heredity*, 86:409–41, 1995.

[18] L. Helmuth. Map of the human genome 3.0. *Science*, 293(5530):582–585, 2001.

[19] N. Immerman. *Descriptive Complexity*. Springer-Verlag, New York, 1999.

[20] G. Lancia, M. C. Pinotti, and R. Rizzi. Haplotyping populations by pure parsimony: Complexity of exact and approximation algorithms. *INFORMS Journal on Computing*, 16(4):348–359, 2004.

[21] G. Lancia and R. Rizzi. A polynomial case of the parsimony haplotyping problem. *Operations Research Letters*, 34(3):289–295, 2006.

[22] Y. Liu and C.-Q. Zhang. A linear solution for haplotype perfect phylogeny problem. In *Proceedings of the International Conference on Advances in Bioinformatics and its Applications*, pages 173–184. World Scientific, 2005.

[23] R. Vijaya Satya and A. Mukherjee. An optimal algorithm for perfect phylogeny haplotyping. *Journal of Computational Biology*, 13(4):897–928, 2006.

[24] R. Vijaya Satya and A. Mukherjee. The undirected incomplete perfect phylogeny problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2008. to appear, doi:10.1109/TCBB.2007.70218.

[25] R. Sharan, B. V. Halldórsson, and S. Istrail. Islands of tractability for parsimony haplotyping. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(3):303–311, 2006.

[26] L. van Iersel, J. Keijsper, S. Kelk, and L. Stougie. Shorelines of islands of tractability: Algorithms for parsimony and minimum perfect phylogeny haplotyping problems. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):301–312, 2008.