Computational Complexity of Perfect-Phylogeny-Related Haplotyping Problems

Michael Elberfeld and Till Tantau

Institut für Theoretische Informatik
Universität zu Lübeck, 23538 Lübeck, Germany
{elberfeld,tantau}@tcs.uni-luebeck.de

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 $\mathrm{Mod}_2\mathrm{L}$, 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.

Keywords: bioinformatics, haplotyping, computational complexity, circuit classes, perfect phylogenies

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 on which chromosome a 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 [11, 12, 18] 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, 13, 16, 19] 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 [26]. This is remedied by Gusfield's perfect-phylogeny-based approach [17]. 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 [17]. 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, 27].

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, 23, 24] 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 two haplotypes for each individual, 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 [15] 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 $\in \mathrm{Mod}_2\mathrm{L}$, 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. [15] for directed perfect path phylogenies to the undirected case. We also show that MPPPH is complete for the uniform threshold circuit complexity class TC^0, which is the same as FO(COUNT).

Restricting the tree topology to a path is one way of simplifying the PPH problem. Another approach that has also 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 [17]. 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, 10]. A few years later three groups independently devised linear time algorithms for the problem [8, 23, 24]. 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. [15] 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 PPPH problem is somehow "easier" than PPH. 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, 27]. One might try to explain these contrasting complexities by arguing that "considering perfect *path* phylogenies rather than perfect phylogenies makes the problems 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 [15] just like IPPH.

Van Iersel et al. [27] 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. Due to lack of space, most proofs are omitted; the missing proofs can be found in the technical report version [9].

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 $i \in \{1,\ldots,n\}$ 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 $i \in \{1,\ldots,n\}$ 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 $\left[\begin{smallmatrix}0&0\\0&1\\1&0\\1&1\end{smallmatrix}\right]$. 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 [10] 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* $\mathrm{ind}_B(c, c') \subseteq \{00, 01, 10, 11\}$ contains all bitstrings $ab \in \{00, 01, 10, 11\}$ for which 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* $\mathrm{ind}_A(c, c') \subseteq \{00, 01, 10, 11\}$ is the intersection of all $\mathrm{ind}_B(c, c')$ where B is any haplotype matrix that explains A. This means, for instance, that the induce of the two columns of the genotype matrix $\left[\begin{smallmatrix}0&1\\2&0\end{smallmatrix}\right]$ is $\{01, 00, 10\}$, the induce of $\left[\begin{smallmatrix}2&0\\1&2\end{smallmatrix}\right]$ is $\{00, 10, 11\}$, and the induce of $\left[\begin{smallmatrix}0&0\\2&2\end{smallmatrix}\right]$ 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 $\left[\begin{smallmatrix}0&0\\1&1\end{smallmatrix}\right]$ or $\left[\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\right]$ 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 only needs to be $O(\log^2 n)$.

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 $\mathcal{A} = (I^{\mathcal{A}}, A_0^{\mathcal{A}}, A_1^{\mathcal{A}}, A_2^{\mathcal{A}}, n_r^{\mathcal{A}}, n_c^{\mathcal{A}})$ as follows: $I^{\mathcal{A}}$ is a finite set of indices, which are used both for rows and columns, and $n_r^{\mathcal{A}}$ and $n_c^{\mathcal{A}}$ are elements from $I^{\mathcal{A}}$ that are the maximum row and column indices, respectively. The relation $A_0^{\mathcal{A}} \subseteq I^{\mathcal{A}} \times I^{\mathcal{A}}$ indicates 0-entries, that is, $(r, c) \in A_0^{\mathcal{A}}$ holds exactly if the row r has a 0-entry in column c. The relations $A_1^{\mathcal{A}}$ and $A_2^{\mathcal{A}}$ are defined similarly, only for 1- and 2-entries. We assume that the universe $I^{\mathcal{A}}$ is ordered (we always have free access to a predicate $<$), but will not need the bit-predicate (see [20] 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 than PPH since Eskin, Halperin and Karp [10] 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 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 $\in \mathrm{Mod}_2\mathrm{L}$. 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 reductions.*

Proof (sketch). We show that there is a first-order reduction from the reachability problem for undirected graphs to the complement of PPH. Given a graph G and a start node s and a target node t we first modify the graph such that any path from s to t must have even length. Then we construct a genotype matrix such that the resolutions of certain column pairs are predetermined by induces. The matrix is setup in such a way that an edge joining two nodes enforces that two specific columns in the matrix are resolved unequally. This can be used to enforce that columns corresponding to s and to nodes at an even distance from s must be resolved equally. By adding further restrictions that enforce that the columns of s and t must be resolved unequally, we can ensure that (a) if s and t lie in different components there exists an explaining haplotype matrix that admits a perfect phylogeny and (b) otherwise every explaining haplotype matrix violates the four gamete property. □

Theorem 3.2. PPH *is in* $\mathrm{Mod}_2\mathrm{L}$.

Proof (sketch). 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 [10]. Solving systems of linear equations over $\mathbb{Z}/2\mathbb{Z}$ is in $\mathrm{Mod}_2\mathrm{L}$ as shown in [3] and since $\mathrm{Mod}_2\mathrm{L}$ 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 [26] 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 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 hold in practice and that make the problem tractable. In different papers parameters k and l have been determined such that MH(k, l) is either efficiently solvable or NP-complete [4, 21, 22, 26, 27]. 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 PPH$(3, *)$ is L-hard; and it is known [27] that MH$(3, *)$ and MPPH$(3, *)$ are NP-complete. We show PPH$(2, *) \in$ FO and PPH$(*, 1) \in$ FO; and it is known [4, 22, 27] that MH$(2, *)$, MPPH$(2, *) \in$ P and MH$(*, 1)$, MPPH$(*, 1) \in$ P. We do not know the complexity of PPH$(*, 2)$; and the complexity MH$(*, 2)$ and MPPH$(*, 2)$ are also open.

Theorem 3.3.

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

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. [15] and later studied in [14]. 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 not. 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 an *(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 [15] 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. [15]) *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.*

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

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

To prove Theorem 4.3, it suffices to show that the characterization given in Lemma 4.2 can be tested using a first-order formula. Theorem 4.4 can be proved by a first-order reduction from PPPH to DPPPH.

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 under* AC^0*-reductions.*

Proof (sketch). First, we show that $(A, d) \in$ MPPPH if, and only if, $A \in$ PPPH and d is greater than the number of distinct polymorphic columns in A. Due to this characterization, MPPPH has nearly the same complexity as PPPH, we only need to add counting quantifiers, which are used to count the number of distinct polymorphic columns in an input matrix. This implies MPPPH $\in TC^0$. To prove the TC^0-hardness of MPPPH, we present an AC^0-reduction from MAJORITY, where a binary string is given and the question is whether at least half of the input bits are 1. We construct unique genotypes for bits that equal 0 and use an MPPPH oracle gate with an appropriate budget value to count them. □

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 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 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. *J. Comput. Biol.*, 11(5):858–866, 2004.
2. V. Bafna, D. Gusfield, G. Lancia, and S. Yooseph. Haplotyping as perfect phylogeny: A direct approach. *J. Comput. Biol.*, 10(3–4):323–340, 2003.
3. G. Buntrock, C. Damm, U. Hertrampf, and C. Meinel. Structure and importance of logspace-MOD-classes. *Math. Syst. Theor.*, 25(3):223–237, 1992.
4. R. Cilibrasi, L. van Iersel, S. Kelk, and J. Tromp. On the complexity of several haplotyping problems. In *Proc. WABI 2005*, volume 3692 of *Lecture Notes in Comput. Sci.*, pages 128–139. Springer, 2005.
5. A. G. Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. *J. Mol. Biol. and Evol.*, 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. *Nat. Genet.*, 29:229–232, 2001.

7. R. P. Dilworth. A decomposition theorem for partially ordered sets. *Ann. Math.*, 51(1):161–166, 1950.
8. Z. Ding, V. Filkov, and D. Gusfield. A linear-time algorithm for the perfect phylogeny haplotyping (PPH) problem. *J. Comput. Biol.*, 13(2):522–553, 2006.
9. M. Elberfeld and T. Tantau. Computational complexity of perfect-phylogeny-related haplotyping problems. Tech. Rep. SIIM-TR-A-08-02, Universität zu Lübeck, 2008.
10. E. Eskin, E. Halperin, and R. M. Karp. Efficient reconstruction of haplotype structure via perfect phylogeny. *J. Bioinform. and Comput. Biol.*, 1(1):1–20, 2003.
11. L. Excoffier and M. Slatkin. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. *Mol. Biol. and Evol.*, 12(5):921–7, 1995.
12. D. Fallin and N. Schork. Accuracy of haplotype frequency estimation for biallelic loci via the expectation-maximation algorithm for unphased diploid genotype data. *Am. J. Hum. Genet.*, 67:947–959, 2000.
13. 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. *Am. J. Hum. Genet.*, 69:831–843, 2001.
14. J. Gramm, T. Hartman, T. Nierhoff, R. Sharan, and T. Tantau. On the complexity of SNP block partitioning under the perfect phylogeny model. *Discrete Math.*, 2008. to appear, doi:10.1016/j.disc.2008.04.002.
15. J. Gramm, T. Nierhoff, R. Sharan, and T. Tantau. Haplotyping with missing data via perfect path phylogenies. *Discrete and Appl. Math.*, 155(6–7):788–805, 2007.
16. D. Gusfield. Inference of haplotypes from samples of diploid populations: complexity and algorithms. *J. Comput. Biol.*, 8(3):305–23, 2001.
17. D. Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In *Proc. RECOMB 2002*, pages 166–175. ACM Press, 2002.
18. M. Hawley and K. Kidd. Haplo: A program using the EM algorithm to estimate the frequency of multi-site haplotypes. *J. Hered.*, 86:409–41, 1995.
19. L. Helmuth. Map of the human genome 3.0. *Science*, 293(5530):582–585, 2001.
20. N. Immerman. *Descriptive Complexity*. Springer-Verlag, New York, 1999.
21. G. Lancia, M. C. Pinotti, and R. Rizzi. Haplotyping populations by pure parsimony: Complexity of exact and approximation algorithms. *INFORMS J. Comput.*, 16(4):348–359, 2004.
22. G. Lancia and R. Rizzi. A polynomial case of the parsimony haplotyping problem. *Oper. Res. Lett.*, 34(3):289–295, 2006.
23. Y. Liu and C.-Q. Zhang. A linear solution for haplotype perfect phylogeny problem. In *Proc. Int. Conf. Adv. in Bioinfor. and Appl.*, pages 173–184. World Scientific, 2005.
24. R. Vijaya Satya and A. Mukherjee. An optimal algorithm for perfect phylogeny haplotyping. *J. Comput. Biol.*, 13(4):897–928, 2006.
25. R. Vijaya Satya and A. Mukherjee. The undirected incomplete perfect phylogeny problem. *IEEE/ACM T. Comput. Biol. and Bioinfor.*, 2008. to appear, doi:10.1109/TCBB.2007.70218.
26. R. Sharan, B. V. Halldórsson, and S. Istrail. Islands of tractability for parsimony haplotyping. *IEEE/ACM T. Comput. Biol. and Bioinfor.*, 3(3):303–311, 2006.
27. 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 T. Comput. Biol. and Bioinfor.*, 5(2):301–312, 2008.