The Binary Perfect Phylogeny with Persistent Characters

Paola Bonizzoni[1], Anna Paola Carrieri[1], Riccardo Dondi[3], and Gabriella Trucco[2]

[1] Dipartimento di Informatica Sistemistica e Comunicazione
Univ. degli Studi di Milano - Bicocca
bonizzoni@disco.unimib.it
[2] Dipartimento di Tecnologie dell'Informazione Univ. degli Studi di Milano, Crema
gabriella.trucco@unimi.it
[3] Dipartimento di Scienze dei Linguaggi, della Comunicazione e degli Studi Culturali
Univ. degli Studi di Bergamo, Bergamo
riccardo.dondi@unibg.it.

1 Introduction

The perfect phylogeny is one of the most investigated models in different areas of computational biology. This model derives from a restriction of the parsimony methods used to reconstruct the evolution of species (taxa) characterized by a set of characters that are gained and/or lost during the evolution. In this paper we focus on the binary characters that can take only the states 0 or 1, usually interpreted as the presence or absence of the attribute in the taxa. Restrictions on the type of changes from zero to one and vice versa lead to a variety of specific models [4]. The most restrictive parsimony assumption is perfect phylogeny: a tree in which each character state can change its state from 0 to 1 at most once. The algorithmic solution of the Perfect Phylogeny model has been investigated in [5], where a well known characterization of matrices admitting a perfect phylogeny[4] and a linear time algorithm are provided. The perfect phylogeny model has been successfully applied in the context of haplotype inference [6] and very efficient polynomial time solutions to this problem have been proposed, including linear-time algorithms [3], [10], [1]. However, this model is quite restrictive to explain the biological complexity of real data, where homoplasy events (such as recurrent mutations or back mutations) are present. Thus a central goal in this model is to extend its applicability, while retaining the computational efficiency where possible.

Following this research direction, in this paper we address the problem of constructing a perfect-phylogeny under the assumption that only a special type of back mutation may occur in the tree: a character may change state only twice in the tree from 0 to 1 and from 1 to 0. These characters have already been considered in the literature and called *persistent* by T. Przytycka [9] in a general framework of tree inference.

We consider a binary matrix M of size $n \times m$ that has columns associated with the set $C = \{c_1, \ldots, c_m\}$ of characters and rows associated with the set $S = \{s_1, \ldots, s_n\}$ of species, then $M[i,j] = 1$ if and only if species s_i has character c_j, otherwise $M[i,j] = 0$. The gain of a character c in a phylogenetic tree is usually represented by an edge labelled by the character c. In order to model the presence of persistent characters, the loss of a character c in the tree is represented by an edge that is labelled by the negated character, denoted by $\bar{c}$. Formally, we define the persistent perfect phylogeny model as follows.

Persistent Perfect Phylogeny Let M be a binary matrix of size $n \times m$. Then a *persistent perfect phylogeny*, in short *p-pp tree* for M, is a rooted tree T that satisfies the following properties:

1. each node x of T is labelled by a vector l_x of length m. The root of T is labelled by a vector of all zeros, while for each node x of T the value $l_x[j]$ represents the state, 0 or 1, of character c_j in tree T. Each row of M labels exactly one leaf of T;
2. for each character c_j there are at most two edges $e = (x, y)$ and $e' = (u, v)$ such that e, e' occur along the same path from the root to a leaf of T; if e is closer to the root than e', then the edge e is labelled c_j, while edge e' is labelled $\bar{c}_j$;

[4] A binary matrix M admits a rooted perfect phylogeny if and only if it does not contain a pair of columns and three rows inducing the configurations $(0,1), (1,0)$ and $(1,1)$, also known as *forbidden matrix*.

Let us state the main problem investigated in the paper.

The **Persistent Perfect Phylogeny problem (P-PP):** Given a binary matrix M, returns a p-pp tree for M if such a tree exists.

We say that two positive characters c, c' of matrix M are in *conflict* in matrix M, if and only if the pair of columns c, c' of M induces the four gametes $(0,1)$, $(1,1)$, $(1,0)$ and $(0,0)$. Then the *conflict graph*[5] associated with a binary matrix M is the undirected graph $G_c = (C, E \subseteq C \times C)$ where a pair $(u,v) \in E$ if and only if u, v are in conflict in matrix M. Notice that a conflict graph with no edges (called *e-empty*) does not necessarily imply the existence of a rooted perfect phylogeny, because of the occurrence of the forbidden matrix with only the three configurations $(1,1)$, $(1,0)$ and $(1,0)$. However, by allowing a character to be persistent, the matrix admits a rooted persistent perfect phylogeny.

In this paper we propose a graph-based solution of the problem of the reconstruction of the persistent perfect phylogeny (in short P-PP problem) that is obtained by restating the problem as a variant of the Incomplete Directed Perfect Phylogeny [8], called Incomplete Perfect Phylogeny with Persistent Completion (IP-PP problem). We show a polynomial-time algorithm that finds a solution for the input matrices described by an e-empty conflict-graph. Then we use it to develop an optimized version of the exact algorithm, that has been presented in [2] and having a a worst time complexity that is polynomial in the number n of rows of the matrix and exponential in the number m of characters. An experimental analysis shows that the new optimized version outperforms the previously proposed algorithm and has a wider applicability, since it can solve all input matrices within fixed time bounds, while the previous algorithm was not able to finish on some of them.

2 Solving the Persistent Perfect Phylogeny problem

Let M be a binary $n \times m$ matrix which is an instance of the P-PP problem. The *extended matrix* associated with M is a matrix M_e of size $n \times 2m$ over alphabet $\{0, 1, ?\}$ which is obtained by replacing each column c of M by a pair of columns $(c, \bar{c})$, where c is the positive character, while $\bar{c}$ is the negated character, moreover for each row s of M, it holds:

if $M[s,c] = 1$, then $M_e[s,c] = 1$ and $M_e[s,\bar{c}] = 0$,

if $M[s,c] = 0$, then $M_e[s,c] = ?$ and $M_e[s,\bar{c}] = ?$.

Informally, the assignment of the pair $(?,?)$ in a species row s for the pair of columns $(c,\bar{c})$ means that character c could be persistent in species s, i.e. it is gained and then lost. On the contrary, the pair $(1,0)$ assigned in a species row s for the pair $(c,\bar{c})$, means that character c is only gained by the species s. A *completion of a character* c of matrix M_e is obtained by solving the pair $(?,?)$ given in the pair $(c, \bar{c})$ by the value $(0,0)$ or $(1,1)$. A *completion of matrix* M_e is a completion of all characters of M_e, while a *partial completion of* M_e is a completion of zero or more characters of M_e. We introduce below a problem to which we reduce P-PP, as shown in Theorem 1.

Incomplete Perfect Phylogeny with Persistent Completion Problem (IP-PP): given an extended matrix M_e over $\{0, 1, ?\}$ return a completion M' of M_e such that M' admits a pp tree, if it exists.

Thus we state the first result of the paper.

Theorem 1. *Let M be a binary matrix and M_e the extended matrix associated with M. Then M admits a p-pp tree if and only if there exists a completion of M_e admitting a pp tree.*

The notion of red-black graph G_{RB} for a matrix M has been introduced to find a completion of a matrix M_e. It consists of the edge colored graph (V, E) where $V = C \cup S$, given $C = \{c_1, \cdots, c_m\}$ and $S = \{s_1, \cdots, s_n\}$ the set of positive characters and species of matrix M_e, while E is defined as follows: $(s, c) \in E$ is a black edge if and only if $M_e[s,c] = 1$ and $M_e[s,\bar{c}] = 0$.

Realization of a character c and its canonical completion

Let $\mathcal{C}(c)$ be the connected component of graph G_{RB} containing node c. The *realization of character c* in graph G_{RB} consists of:

[5] The conflict graph is a well known concept that has been used several times in the framework of the perfect phylogeny is a graph representation of the four gametes $(0,1)$, $(1,1)$, $(1,0)$ and $(0,0)$.

1. adding red edges connecting character c to all species nodes s that are in $\mathcal{C}(c)$ and such that (c, s) is not an edge of G_{RB},
2. removing all black edges (c, s) in graph G_{RB}. Then c is labelled *active*.
3. if an active character c' is connected by red edges to all species of that are in $\mathcal{C}(c')$, then its outgoing red edges are deleted from the graph and c is labelled c' *free*.

The realization of a character c is associated with a *canonical completion* of character c in matrix M_e that is defined by completing each pair $(?, ?)$ occurring in the pair $(c,\bar{c})$ as follows: the pair $(?, ?)$ is completed by $(1, 1)$ in every species s that is in the connected component $\mathcal{C}(c)$ of graph G_{RB}, (s is connected with c by a red edge) while value $(0, 0)$ is assigned in the remaining rows. We call *e-empty* a red-black graph without edges. Since we are interested in computing canonical completions of M_e that admit a pp tree, only canonical completions that are obtained by the realization of special sequences of characters of the red-black graph are considered, as defined below.

Definition 1. *Given a graph G_{RB} for an extended matrix M_e, a* successful reduction *of G_{RB} is an ordering $r =< c_{i_1}, \cdots, c_{i_m} >$ of the set of all positive characters of M_e such that the consecutive realization of each character in r leaves an e-empty red-black graph.*

In [2] we show that finding a solution to an instance of the IP-PP problem is equivalent to computing the existence of a successful reduction for the red-black graph for the input matrix. Furthermore by the Theorem 1 a solution to the IP-PP instance M_e is equivalent to a solution to the P-PP instance M.

Theorem 2. *Let M_e be an extended matrix. Then M_e admits a perfect phylogeny, if and only if there exists a successful reduction of the graph G_{RB} for M_e.*

We propose an algorithm, called **Decide-pp-opt**, for the P-PP problem that is based on Theorems 1 and 2. It builds a decision tree that explores all permutations of the set C of characters of M_e in order to find one that is a successful reduction, if it exists.

The following result is a consequence of two technical Lemmas that are omitted for lack of space.

Theorem 3. *Let M be a binary matrix that has an e-empty conflict graph. Then matrix M admits a persistent perfect phylogeny and there exists a polynomial time algorithm to build the p-pp tree for M.*

We give a polynomial time algorithm, in the size of the input matrix M, to find a successful reduction of graph G_{RB}, thus showing that a p-pp tree for M always exists. Given c, c' columns of M, then $c < c'$ if and only if for each species s of M, it holds that $M[s, c] \leq M[s, c']$. Then given M a binary matrix, the *partial order graph* for M is the partial order P obtained by ordering columns of M under the $<$ relation.

The algorithm constructs the partial order graph P for M. Then it iterates the following step to build a successful reduction r: - add to sequence r all element in the set C_M consisting of the maximal ones in P. Remove characters C_M from P.

Furthermore we propose a optimized version of the exact algorithm presented in [2] that uses the polynomial time algorithm for an e-empty conflict graph.

Algorithm Decide-pp-opt(M, M', x, T)
Input: a binary matrix M of size $n \times m$, a partial depth-first visit tree T of the decision tree $\mathcal{T}$ and a leaf node x of $\mathcal{T}$, a partial completion M' of the extend matrix M_e obtained by the realization of the characters labelling a path π from r to node x of the tree T;
Output: the tree T extended with the depth-first visit of T from node x. The procedure eventually outputs a successful reduction r or fails to find such a successful reduction.

- Step 1: if the incident edge to node x is labelled c, then realize c in G_{RB} and complete the pair of columns (c, c') in M'. If the matrix M' has a forbidden matrix, then label x as a fail node.

- Step 2: compute the conflict graph G_c for the matrix M updated after the realization of the characters along the path π from the root r to node x (i.e. M is obtained after eliminating the rows that correspond to species-nodes that are singletons in G_{RB}),
- Step 3: if the conflict graph G_c is e-empty, then apply the polynomial-time algorithm for an empty conflict-graph and return a successful reduction. Else for each node x_i that is a child of node x in tree T and is labelled by a non-active character in G_{RB}, apply Decide-pp-opt(M, M', x_i, $T \cup \{x_i\}$).

The algorithm **Decide-pp-opt** has been implemented and tested over simulated data produced by the tool *ms* by Hudson [7]. We have implemented the algorithm in C++ and the experiments have been run on a standard Windows workstation with 4 GB of main memory.

Table 1 reports the computation time to solve sets of 50 matrices for each dimension $(50, 15)$, $(100, 15)$, $(200, 15)$, and $(500, 15)$ with a recombination rate 1 over 15. The sets contain only matrices that are solved within 5 minutes. Another experiment has been done with 10 matrices of the same size 50×15 and different number of edges in the conflict graph. The average time was 0.015, 0.031 and 0.051, respectively for the case of $1, 5$ and 10 conflicts. Clearly, the number of unsolved matrices increases with the size of the input matrices but also with the number of conflicts that are present in the conflict graph. In order to test the performance of the algorithm for large matrices in terms of number of species we have processed a matrix of size 1000×15 with a conflict graph having 9 conflicts (edges). It took 35.5 seconds to find the solution to the matrix. We also compared the execution times of the exact algorithm and the optimized algorithm on sets of matrices with fixed number of columns and different numbers of rows. The **Decide-pp-opt** algorithm is able to find a solution for all matrices in contrast to the **Decide-pp** algorithm that in some cases takes more than 10 minutes to find a solution for a single matrix.

nxm	no P-PPH	tot conflicts	average conflicts	solved matrices		total time in s		average time in s	
				algorithm	opt. algorithm	algorithm	opt. algorithm	algorithm	opt. algorithm
50x15	6	236	4.72	47	50	89.12	32.32	1.90	0.65
100x15	4	175	3.5	48	50	436.02	194.63	9.08	3.89
200x15	3	147	2.94	48	50	1583.50	43.21	32.99	0.86
500x15	7	219	4.38	44	50	888.59	889.43	20.20	17.79

Table 1. The table has entries to specify the average time to solve a single matrix (in seconds shortened as s), the number of matrices that do not admit a p-pp tree, the total number of conflicts, measured as the number of edges in the graph G_c of the matrices of each set, and the average number of conflicts. Each considered matrix has a conflict graph G_c that consists of a single non trivial component.

References

1. P. Bonizzoni. A linear time algorithm for the Perfect Phylogeny Haplotype problem. *Algorithmica*, 48(3):267–285, 2007.
2. P. Bonizzoni, R. Dondi, C. Braghin, and G. Trucco. The persistent perfect phylogeny model. *Theoretical Computer Science*, to appear, 2012.
3. Z. Ding, V. Filkov, and D. Gusfield. A linear time algorithm for Perfect Phylogeny Haplotyping (pph) problem. *Journal of Computational Biology*, 13(2):522–553, 2006.
4. J. Felsenstein. *Inferring Phylogenies.* Sinauer Associates, 2004.
5. D. Gusfield. Efficient algorithms for inferring evolutionary trees. *Networks*, pages 19–28, 1991.
6. D. Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In *Proc. 6th Annual Conference on Research in Computational Molecular Biology (RECOMB 2002)*, pages 166–175, 2002.
7. R. R. Hudson. Generating samples under a wright-fisher neutral model of 31 genetic variation. *Bioinformatics*, 18(2):337–338, 2002.
8. I. Peer, T. Pupko, R. Shamir, and R. Sharan. Incomplete directed perfect phylogeny. *Siam Journal on Computing*, 33(3):590–607, 2004.
9. T. M. Przytycka. An important connection between network motifs and parsimony models. In *Proc. 10th Annual Conference on Research in Computational Molecular Biology (RECOMB 2006)*, pages 321–335, 2006.
10. V. Satya and A. Mukherjee. An optimal algorithms for perfect phylogeny haplotyping. *Journal of Computational Biology*, 13(4):897–928, 2006.