When and how the Perfect Phylogeny model explains Evolution

Paola Bonizzoni, Anna Paola Carrieri, Gianluca Della Vedova, Riccardo Dondi, and Teresa M. Przytycka

Abstract Character-based parsimony models have been among the most studied notions in computational evolution, but research in the field has stagnated until some important and recent applications, such as the analysis of data coming from proteins domains, protein networks and genetic markers, as well as haplotyping, have brought new life into this sector. The focus of this survey is to present the perfect phylogeny model and some of its generalizations. In particular, we develop the use of persistency in the perfect phylogeny model as a new promising computational approach in analyzing and reconstructing evolution. We show that, in this setting, some graph theoretical notions can provide a characterization of the relationships between characters or attributes, playing a crucial role in developing algorithmic solutions for the problem of reconstructing a maximum parsimony tree.

1 Introduction

Evolution is the lens that allows to study and understand a lot of phenomena in Molecular Biology [8]. The prototypical representation of any evolutionary history is a phylogeny, that is a labeled tree whose leaves are the extant species, or individuals, or simply data that we are currently able to analyze [10]. Phylogenetics is the research area of Computational Biology devoted to computing phylogenies. In this field, the focus has shifted through the years. The initial developments date back to the pioneering work by Cavalli-Sforza and Edwards [6, 9] in the 60s, where

Paola Bonizzoni, Anna Paola Carrieri, Gianluca Della Vedova
DISCo, Università degli Studi di Milano–Bicocca, Milan (Italy)

Riccardo Dondi
Dipartimento di scienze umane e sociali. Università degli Studi di Bergamo, Bergamo(Italy)

Teresa Przytycka
National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, USA,

some fundamental ideas of the study of phylogenies have been introduced, namely the fact that evolution is a branching process where characters are changing, that an intuitive argument is to find the minimum total amount of evolutionary events compatible with available data, and that of maximizing the likelihood of the proposed interpretation.

The limited computational resources of the time, together with the kind of data available (phenotypical data were much more frequent than genomic data), initially put emphasis on maximum parsimony character-based approaches. Successively new advances, also in the statistical modeling of evolution [10], made the approaches based on inferring maximum likelihood phylogenies more attractive.

More recently, the pendulum has swung again, as parsimony methods have found new relevance, mostly due to new applications and data provenance. The perfect phylogeny model, which is conceptually the simplest, is based on the infinite sites assumption, that is no character can mutate more than once in the whole tree. While this assumption is quite restrictive, bordering on the plainly wrong for some cases, the perfect phylogeny model turned out to be splendidly coherent within the haplotyping problem [4, 17], where we want to distinguish the two haplotypes present in each individual when given only genotype data. More precisely, the interest here is in computing a set of haplotypes and a perfect phylogeny such that the haplotypes (i) label the vertices of the perfect phylogeny and (ii) explain the input set of genotypes. This context has been deeply studied in the last decade, giving rise to a number of beautiful algorithms [2, 7]. Those algorithms (and others on the same topics) exploit a number of nice combinatorial properties of perfect phylogenies and graphs. In [2], a graph theoretical characterization of genotype matrices admitting a tree representation is given by using properties of partial orders and Dilworth's Theorem [11], while in its original formulation the haplotyping problem under the perfect phylogeny model [7, 17] has revealed interesting connection with the graph realization problem [28], a well known graph problem used to decide whether a matroid is graphic.

Still, the perfect phylogeny model and the assumptions that have been central in the previous decades cannot be employed without adaptations or improvements. A main open problem regarding the model is finding generalizations that retain the computational tractability of the original model but are more flexible in modeling biological data. Following this research direction, we explore some extensions of the perfect phylogeny model that are capable of modeling some processes whose study is motivated by some recent applications.

In particular, we present two recent applications that can find only a partial solution in perfect phylogenies. The first application is carcinogenesis, i.e., the factors and mechanisms that cause the onset of cancer in cells. The carcinogenesis can result from many combinations of mutations, but only a few sequences of mutations, called *progression pathways*, seem to account for most human tumors [27]. The main issue here is characterizing the common progression pathways as a first step towards identifying therapeutic targets and reliable diagnostic tests. The natural observation that tumors are evolving cell populations leads to phylogeny-based studies. At the same time the intrinsic nature of cancer cells, that is cell proliferating quickly and in a degenerate way, results in a relative high amount of sites with multiple mutations (i.e., in violations of the infinite sites assumption).

A second application regards the study of protein domains. A protein domain is a part of protein sequence and structure that can evolve, function, and exist independently of the rest of the protein chain. Many proteins consist of several structural domains, while a domain may appear in a variety of different proteins. In this case it is quite frequent to acquire a domain and then to lose it (much more frequent than acquiring and then losing the whole gene). Again, the infinite site assumption can be violated.

In this survey, we pay special attention to an approach proposed in [23] and based on the notion of persistent character inside the perfect phylogeny model, and on its use to exclude some characters from the construction of the phylogeny. The general focus will be on computational issues, such as efficient algorithms.

2 Maximum Parsimony and the Perfect Phylogeny

Parsimony models, just as all models, are characterized by specific constraints that are based on some biological assumptions. The first basic assumption states that each species or taxon is described by a set of attributes, called characters, where each character is inherited independently, and each character can assume one of a finite set of values, also called *states*. Alternatively, the input is a matrix whose rows are the taxa and the columns are the characters. Another basic assumption about the evolution of characters, called *homology*, assumes that characters that are present in more than one species must be inherited by a common ancestor.

The natural computational problem has, as input, a matrix M with n rows and m columns, where each row can be viewed as a m-vector over the set of states of characters. The matrix describes a set of n taxa (species or individuals) – corresponding to the rows of M – and a set of m characters – corresponding to the columns of M – and asks for a minimum cost tree explaining the input matrix M. In a tree T *explaining* a matrix M (i) the nodes are labeled by m-long vectors of states, (ii) each row of M labels exactly one node of T, (iii) the leaves are labeled by some rows of M (iv) each edge (r_1, r_2) of T is labeled by the character c of M whose state differs in r_1 from that in r_2 (see Figure 1 as an example).

The cost of a tree is the number of mutations in the tree or, more formally, the sum over all edges of the tree of the cost of each edge, given by the number of characters with different states in the two nodes composing the edge. In binary parsimony models – the most widely used – characters can take only the values (or states) zero or one, usually interpreted as the presence or absence of the attribute in the taxa.

We will now discuss how computing the maximum parsimony phylogeny can be framed as a Steiner tree problem, which is one of the most widely studied problem in Operations Research. Recall that each input taxon is viewed as a binary vector of length m. Then the set of all possible binary vectors of length m forms a hypercube H, whose edges are exactly the pairs of vertices (u, v) where u and v differ in exactly

M	c_1	c_2	c_3	c_4	c_5
s_1	1	1	0	0	0
s_2	0	0	1	0	0
s_3	1	1	0	0	1
s_4	0	0	1	1	0

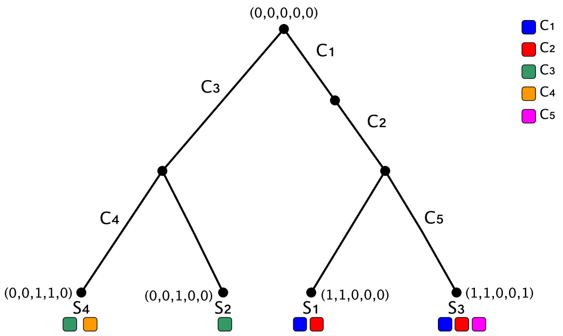

Fig. 1: Example of perfect phylogeny over a binary matrix M of 5 characters and 4 species.

one position. Let S be the set of input species of the phylogeny problem, and notice that S is also a subset of the vertices of the hypercube H. Then the Steiner tree problem asks for a minimum cost subtree T of H such that all vertices in S are also in T. The cost of the solution T is the number of edges of T. The Steiner tree problem is NP-hard [20], even in the case of a binary alphabet with the metric induced by the Hamming distance [13], which is the restriction deriving from the reduction from the maximum parsimony phylogeny to the Steiner tree on a hypercube. Extensive recent work, both experimental and theoretical, has focused on the binary character set with the Hamming metric [24, 26].

We can now introduce some specific parsimony models, starting from the simplest: the perfect phylogeny. A tree is called a *perfect* phylogeny if each character i mutates exactly once (i.e., there is exactly one edge such that the vertices are labeled by vectors differing in position i). Notice that a perfect phylogeny (if it exists) minimizes the overall cost, as any perfect phylogeny has cost m. We call the perfect phylogeny *directed* or *rooted* if there is a distinguished node corresponding to the $[0,\dots,0]$ vector. It is immediate to notice that we can transform a perfect phylogeny into a rooted perfect phylogeny, by choosing an arbitrary node x and flipping (in each species) the state of each character that initially has value 1 in x (those characters are also called *active* in x). There is a well-known linear time algorithm for computing a binary perfect phylogeny [16], if it exists, and some more involved fixed-parameter algorithms for the general perfect phylogeny problem, where the parameter is the maximum number of states for each character [1, 19]. In the following, unless specified differently by perfect phylogeny we mean a rooted perfect phylogeny, that is characters mutate only from state zero to one.

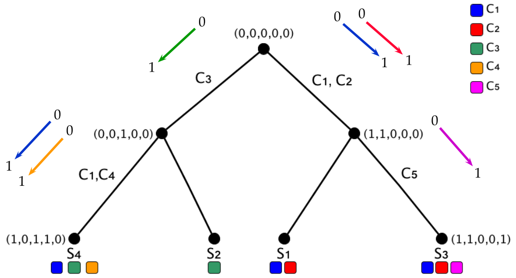

Fig. 2: Example of Camin-Sokal parsimony over the same set of characters as Fig. 3. Observe that character c_1 is gained twice in the tree.

3 Dollo Parsimony and its Variants

Unfortunately, there are some evolutionary phenomena, such as *homoplasy*, that violate the fundamental assumptions of perfect phylogeny [10]. Two kinds of homoplasies are *recurrent mutations* and *back mutations*. The first event occurs when a character changes state along divergent branches of the tree, while a back mutation implies that a character may go back to the ancestral state in descendant species after changing state. The two types of events justify the introduction of different models, differing mainly in the allowed homoplasies. While the perfect phylogeny model does not allow any homoplasy, some extended models have been introduced to allow recurrent or back mutations.

An extended model is the Camin-Sokal parsimony [5], where characters are *directed*, that is only changes from zero to one are possible on any path from the root to a leaf. This fact means that the root is assumed to be labeled by the ancestral state with all zeroes, and no back mutation is allowed, but any character can be acquired more than once, i.e., recurrent mutations are possible.

Another possible way of extending the perfect phylogeny model is that of the Dollo parsimony, which allows any character to change state from zero to one only once, but gives no restriction on the number of times that it mutates from one to zero [10], that is back mutations are allowed, but recurrent mutations are not. The definition of Dollo parsimony implies that characters are acquired at most once in the tree, but may be lost multiple times.

An interesting application of the Dollo parsimony is the analysis of dynamic protein interactions [30], which has also shown an interesting connection with graph theory. Protein networks are graphs modeling protein interactions. More precisely, nodes of a graph are the proteins studied and edges represent the interactions. A functional module is a subset of the proteins that have a common biological function. Usually a functional module is not a generic graph, as it is made of overlapping cliques or quasi-cliques (that are called functional groups or complex). It is possible to represent those functional groups and complexes interactions with a tree, called *tree of complexes*, whose nodes are the functional groups (to be identified as cliques or quasi-cliques of an original protein network) and such that the set of nodes consisting of the functional groups containing any given common protein is connected.

Let us denote each complex or protein with a distinct symbol of the alphabet Σ. Then the Tree of complexes (TC) problem, over an instance consisting of a set $A = \{a_1, a_2, \cdots, a_m\}$ of subsets of Σ, asks for a tree T, if it exists, whose nodes are the input sets and such that for each $\sigma \in \Sigma$, the set of nodes to which σ belongs is a subtree of T. Clearly, the problem TC admits several solutions that may explain a set S. The following property has never been, to the best of our knowledge, explicitly pointed out previously.

Lemma 1. *Let $A = \{a_1, a_2, \cdots, a_m\}$ be an instance of the TC problem admitting a tree of complexes T. Then T is compatible with the Dollo parsimony model (i.e., no two characters are acquired more than once).*

Proof. Let σ be a generic symbol and let $N(\sigma)$ be the set of nodes of T with σ. By definition of the tree of complexes, $N(\sigma)$ induces a connected subtree of T. It is not limitative to assume $|N(\sigma)| > 1$. Let x be the least common ancestor of $N(\sigma)$, we claim that the arc incoming in x is the only one where σ is acquired. By definition of least common ancestor in a tree, (i) only nodes that are descendants of x can have the symbol σ, and (ii) there are two nodes $v_1, v_2 \in N(\sigma)$ such that all paths of T connecting v_1 and v_2 pass through x, therefore σ is active in x, for otherwise $N(\sigma)$ would be disconnected. Consequently σ is acquired in x. Assume now to the contrary that σ is also acquired in node x_1 which is a descendant of x, and let x_2 be the parent of x_1. Since σ is acquired in node x_1, σ is active in x_1 but not in x_2. Since all paths of T connecting x and x_1 must pass through x_2, $N(\sigma)$ is disconnected, contradicting the hypothesis that T is a tree of complexes.

The connection between trees of complexes and graph theory is deeper. For instance, when S is the set of cliques of a chordal graph, then the tree of complexes can be obtained from the clique tree associated to the chordal graph [30]. In fact, chordal graphs are exactly those admitting a clique tree representation. Let us recall that a graph is chordal if the only vertex-induced subgraphs that are also cycles have exactly three vertices [15].

One of the main open questions in [30] is providing a characterization of the protein networks that admit a tree of complexes representation. Lemma 1 points out the equivalence of this open problem with the question of finding the protein networks that admit an evolutionary representation of functional groups compatible with a Dollo parsimony.

As pointed out in the introduction, the perfect phylogeny model is too restrictive for some applications, since it cannot explain the evolution of characters in the presence of homoplasy events. On the other hand, the optimization problems associated to the Dollo and Camin-Sokal parsimony models are NP-hard [10]. Moreover they are too general to be useful in practical applications where interesting characters are usually affected only by a few back mutations or recurrent mutations. Therefore the research activity has focused on finding some models that couple computational tractability with the capability of adequately modeling actual phenomena,

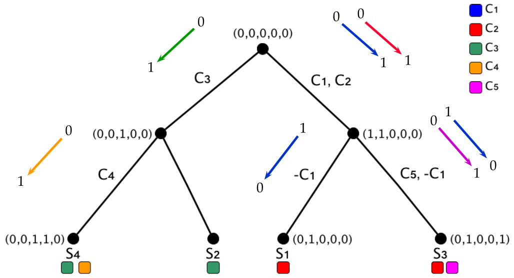

Fig. 3: Example of Dollo parsimony over a matrix of five characters. Observe that character c_1 is the only one that is lost in the tree.

for example in the context of proteomics when analyzing properties of multidomain proteins [22, 23].

Notice that, differently from the perfect phylogeny, a Dollo phylogeny always exists, by assuming a special internal node $[1, \ldots, 1]$ that is also the least common ancestor of all leaves. This fact implies that a rooted Dollo parsimony always exists for an input matrix. Since there is no restriction on mutations from 1 to 0, any binary vector can be generated. While there is no guarantee that such tree is optimal, it suffices to prove the existence of a Dollo phylogeny. However such a tree makes no sense from a biological point of view, because it implies the existence of an ancestral taxon that has all characters in the extant taxa.

We have already pointed out that the problem of constructing a maximum parsimony tree is a special case of the well studied Steiner tree on a hypercube problem, but the set of allowed homoplasies can influence in a fundamental way the computational complexity of the resulting problem. A first effort towards describing new and relevant variants of the Dollo parsimony has been been the introduction of the *conservative Dollo* and *static Dollo parsimony* models [22].

The *static Dollo Parsimony* is a Dollo parsimony where for each node x and for each active character c in x, there exists a leaf l that is a descendant of x and where c is active. The *conservative Dollo Parsimony* is a Dollo parsimony where for each node x and for each pair c_1 and c_2 of active characters, there exists a leaf l that is a descendant of x and where both c_1 and c_2 are active. Notice that both models forbid the presence of an ancestral active character that is not shared with some extant species. The main motivations for those models arise in the study of multi-domain protein evolution in terms of domain insertions and losses. A protein domain is a part of protein sequence and structure that can evolve, function, and exist independently of the rest of the protein chain; the approach followed represents domain structure as taxa, while the domains are the characters. A character active for a certain taxa represents the fact that a domain is part of a given architecture. Hence, a state change from 0 to 1 corresponds to the addition of a domain, while a change from 1 to 0 corresponds to a domain loss. A conservative Dollo parsimony for a protein family is a history where each domain pair that is observed in an extant taxon has been generated from a single merge event. Since the simultaneous presence of two domains in one protein often enhance the functionality of such protein, the model suggests it is highly unlikely that such pair is separated (and its enhanced functionality has not survived) in all extant species.

While the optimization problems, where the number of back mutations is minimized, associated with static and conservative Dollo parsimony are both NP-hard, there are two fast algorithms for testing if such a phylogeny exist [22]. Still, the experimental analysis [22] shows that a sizable minority of multidomain protein superfamilies does not admit a static Dollo parsimony (and, a fortiori, a conservative Dollo Parsimony). Hence an even less restrictive model is necessary to successfully model those cases.

4 Persistent phylogeny

An important ingredient that may affect the applicability and success of parsimony methods is given by the set of characters used to infer the phylogeny. The issue of selecting characters has been addressed in [23], where the notion of *persistent* or *stable* character has been proposed. Such characters are allowed to violate the properties of a perfect phylogeny, as a persistent character is gained exactly once but it can be lost at most once in the tree.

Based on this notion, a different model, which is intermediate between the perfect phylogeny and the Dollo parsimony, called the *persistent phylogeny* has been proposed in [3]. Notice that a persistent perfect phylogeny is also a Dollo phylogeny, or even a static Dollo parsimony. In fact a persistent phylogeny is a static Dollo parsimony where all but at most one of the descendant of a species with any given character must retain such character. Moreover, differently from Dollo Parsimony, some matrices may not admit a persistent perfect phylogeny. Therefore, the main computational problem we will discuss in this section is to compute (if it exists) a persistent perfect phylogeny compatible with a given matrix M. We notice that the computational complexity of the problem is still unsettled, while there exists an algorithm that is exponential in the number of characters, but polynomial in the number of species [3]. This time complexity makes the algorithm of practical interest for biological applications we have discussed before, as usually the number of species is large, while the number of characters is bounded.

The notion of *overlap graph*, which is a graph whose nodes are the characters and two characters are adjacent if and only if there exists a species with both characters, is useful in this context. In fact, if a matrix M admits a persistent phylogeny, then the corresponding overlap graph is chordal [23].

A first application of the Persistent phylogeny model is in the study of introns, which are sequences of non-coding DNA in eukaryotic genes. In fact the Dollo Parsimony model has led to an incorrect evolutionary tree for those data, while assuming the Persistent phylogeny model has resulted in an evolutionary tree consistent with the Coelomata hypothesis, that is a clade comprised of arthropods and chordates. On the contrary the analysis of more variable introns favored the Ecdysozoa topology, that is a clade of arthropods and nematodes [29]. The controversy about the Coelomata and Ecdysozoa topologies is one of the most discussed and persistent problems in animal phylogeny.

For sake of completeness we recall the definition of a persistent phylogeny given in [3]. Let M be a binary matrix of size $n \times m$. The *persistent phylogeny* for M, is a rooted tree T that satisfies the following properties:

1. each node x of T is labeled by a vector l_x of length m;
2. the root of T is labeled by a vector of all zeros, while for each node x of T the value $l_x[j] \in \{0,1\}$ is the state of character c_j at this node;
3. for each character c_j there are at most two edges $e = (x,y)$ and $e' = (u,v)$ such that $l_x[j] \neq l_y[j]$ and $l_u[j] \neq l_v[j]$ (representing a change in the state of c_j) such that e, e' occur along the same path from the root of T to a leaf of T; if e is closer to the root than e', then the edge e where c_j changes from 0 to 1 is labeled c_j^+, while edge e' is labeled c_j^-,
4. each row of M labels exactly one leaf of T.

Thus the main problem investigated in this section, called Persistent Phylogeny problem given in input a binary matrix M asks for finding a persistent phylogeny for M if such a tree exists.

We will devote the remainder of the section to the discussion of the algorithm of [3] for determining whether an input matrix M admits a persistent phylogeny and, in case, to compute such a phylogeny (even though the solution computed might not be the most parsimonious).

First of all, we recall that there exists a very simple test to determine if M admits an unrooted perfect phylogeny. Two characters c_1 and c_2 are in *conflict* in matrix M, if and only if the two corresponding columns of M contain the four possible rows – $(0,0)$, $(0,1)$, $(1,1)$, $(1,0)$ – the so called four gametes. A matrix M has an unrooted perfect phylogeny if and only if no two of its characters are in conflict. The test for a matrix M in the rooted case consists of verifying that M has no induced matrix consisting of the three configurations $(0,1)$, $(1,1)$, $(1,0)$.

Conflicting characters in a matrix can be represented by the undirected *conflict graph* graph $G_c = (C, E \subseteq C \times C)$, where nodes are the characters and two characters are adjacent if they are in conflict in M. Clearly, having an edgeless conflict graph is a necessary but not a sufficient condition for having a rooted perfect phylogeny, but it implies that in that case a persistent phylogeny exists [3]. Moreover the conflict graph is also a measure of the complexity of an instance of the persistent perfect phylogeny reconstruction.

M	a	b	c	d	e
s_1	0	0	0	1	0
s_2	0	0	1	1	1
s_3	0	1	1	0	0
s_4	1	1	0	0	0
s_5	1	1	1	0	1

(a) Binary matrix M.

M'	a^+	a^-	b^+	b^-	c^+	c^-	d^+	d^-	e^+	e^-
s_1	?	?	?	?	?	?	1	0	?	?
s_2	?	?	?	?	1	0	1	0	1	0
s_3	?	?	1	0	1	0	?	?	?	?
s_4	1	0	1	0	?	?	?	?	?	?
s_5	1	0	1	0	1	0	?	?	1	0

(b) Extended matrix M_e.

Fig. 4: An example of binary matrix M which is the input of the persistent phylogeny problem, and its associated extended matrix.

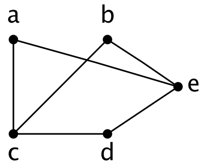

Fig. 5: The conflict graph G_c associated with the binary matrix M of Fig. 4a.

4.1 A graph theoretical solution of the Persistent Phylogeny problem

We can associate an *extended matrix* M_e to the input matrix M, by replacing each column c of M by a pair of columns (c^+, c^-), where c^+ is called the *positive* character, and c^- is called the *negated* character. Moreover for each row s of M, $M_e[s,c^+] = 1$ and $M_e[s,c^-] = 0$ whenever $M[s,c] = 1$, while $M_e[s,c^+] = M_e[s,c^-] = ?$ otherwise. We want to complete the extended matrix M_e, obtaining a new matrix M_f which is equal to M_e for all species s and characters c such that $M_e[s,c] = 1$, while $M_f[s,c^+] = M_f[s,c^-]$ whenever $M_e[s,c] = 0$ (in this case we can interpret $M_f[s,c^-] = 1$ as the fact that the species s does not have the character c, but some of its ancestors used to have it). The idea of completing a matrix with missing data in order to obtain a perfect phylogeny has been introduced in [21], but in our case the completion has some constrains, making the algorithm of [21] inapplicable. Finding such a matrix M_f that admits a perfect phylogeny is equivalent to computing a persistent phylogeny on the original matrix M. The following Theorem has been proved in [3].

Theorem 1. *Let M be a binary matrix and M_e the extended matrix associated with M. Then M admits a a persistent phylogeny if and only if there exists a completion M_f of M_e admitting a perfect phylogeny.*

Figure 4b provides an example of extended matrix M_e, with respect to Figure 4a, whose conflict graph is given in Figure 5.

The red-black graph and the realization of a character

In order to find a completion of the input matrix M_e, another graph representation of the input matrix, called *red-black graph*, in short G_{RB}, is given. The latter consists of the edge colored graph (V,E) where $V = C \cup S$, with $C = \{c_1, \cdots, c_m\}$ 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 $M_e[s,c^-] = 0$. The algorithm for finding a persistent phylogeny basically determines a sequence of *character realizations* which are represented as very specific operations on the red-black graph. The graph operation is called *realization of a character* and consists of removing black edges and adding or removing red edges.

In fact let c be a character and let $\mathscr{C}(c)$ be the connected component of graph G_{RB} containing the node c. Then realizing the character c on G_{RB} consists of the following steps:

(i) adding the red edges (c,s) for all species $s \in \mathscr{C}(c)$ such that (c,s) is not an edge of G_{RB},

(ii) removing all black edges (c,s) – in this case c is called *active* –

(iii) if an active character c_1 is connected by red edges to all species of $\mathscr{C}(c_1)$, then all edges incident of c_1 are deleted and c is called *free*.

Realizing a character c is associated with a *canonical completion* of c in matrix M_e by completing incomplete pairs of characters c^+, c^- as $M_f(c^+,s) = M_f(c^-,s) = 1$ for each species $s \in \mathscr{C}(c)$, while $M_f(c^+,s) = M_f(c^-,s) = 0$ for the other species — we recall that in a completion $M_f(c^+,s) = M_e(c^+,s)$ and $M_f(c^-,s) = M_e(c^-,s)$ if $M_e(c^+,s) \neq M_e(c^-,s)$.

Consequently, any ordering $\langle c_{i_1}, \ldots, c_{i_m} \rangle$ of the character set represents a possible solution, obtained by realizing the characters according to the ordering. Not all orderings lead to actual feasible solution though, but only those whose resulting red-black graph is edgeless [3]. Still, the fundamental result of [3] is the following.

Theorem 2. *Let M be a binary matrix and G_{RB} the red-black graph for matrix M. Then M admits a persistent phylogeny if and only if there exists an ordering of the characters of M such that the realization of characters in that ordering in graph G_{RB} results in an edgeless red-black graph.*

The main consequence of Theorem 2 is that an algorithm for finding a persistent phylogeny, if it exists, is to enumerate all possible orderings of the character set and to compute the red-black graph resulting from realizing the characters in such order. In fact, the algorithm of [3] builds a decision tree that explores all orderings of the set C of characters. An experimental analysis of the computational performance of the algorithm for building a persistent phylogeny has been done in [3].

Example 1. Consider the matrix M given in Figure 4a. In Fig. 7 you can see an example of the realization of characters in the red-black graph according to the ordering $\langle d,c,e,b,a \rangle$. The binary matrix M has associated the conflict graph G_c represented in Fig. 5. The pairs of characters in conflict are (a,c), (b,c), (c,d), (a,e), (b,e), (d,e). The ordering $\langle d,c,e,b,a \rangle$ leads to the canonical completion M'

M'	a^+	a^-	b^+	b^-	c^+	c^-	d^+	d^-	e^+	e^-
s_1	0	0	0	0	0	0	1	0	0	0
s_2	0	0	0	0	1	0	1	0	1	0
s_3	1	1	1	0	1	0	1	1	1	1
s_4	1	0	1	0	1	1	1	1	1	1
s_5	1	0	1	0	1	0	1	1	1	0

Fig. 6: A completion M' of the extended matrix M_e of Fig. 4b.

of Fig. 6. The perfect phylogeny compatible with M' is also a persistent phylogeny for the input matrix M and is represented in Figure 8.

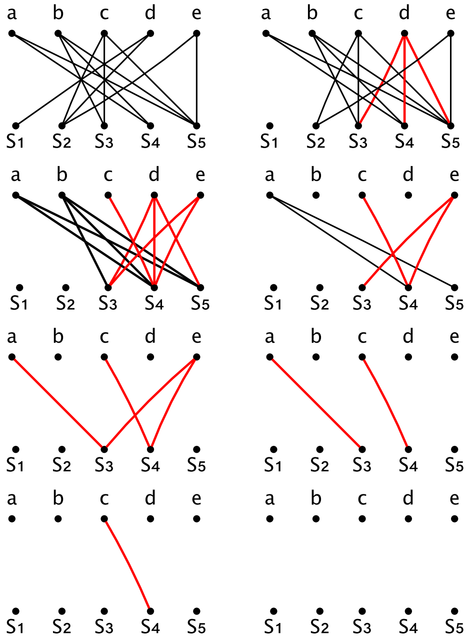

Fig. 7: The realization of $\langle d, c, e, b, a \rangle$ on the red-black graph $G_{R,B}$

5 The near perfect phylogeny

We recall that there are instances M of the perfect phylogeny problem that cannot be solved, motivating the need for different models. Another approach to ensure the construction of a most likely phylogeny according with the input data is

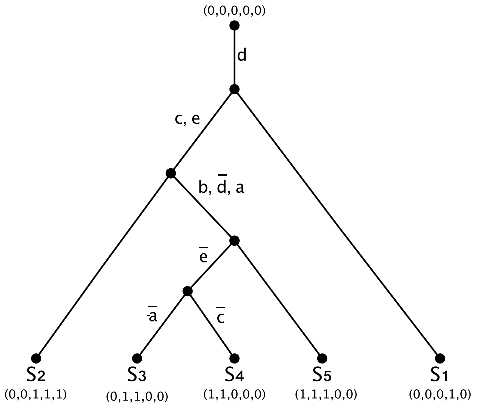

Fig. 8: Realizing the characters in the ordering $\langle d, c, e, b, a \rangle$ results in a persistent phylogeny for M.

moving towards some optimization problems, such as identifying a largest subset of characters admitting a perfect phylogeny or, equivalently, to remove the minimum number of columns from the input matrix M so that the resulting matrix has a perfect phylogeny. This problem is also called the CHARACTER COMPATIBILITY problem. Unfortunately those optimization problems are intractable, as identifying the largest subset of characters admitting a perfect phylogeny is equivalent to MAX CLIQUE [14], sharing also its inapproximability [18]. Consequently different versions must be sought.

An interesting problem stems from the observation that the perfect phylogeny is the minimum cost Steiner tree where the set of species sharing a common state for any given character forms a connected subtree, and that the minimum cost is exactly equal to m (i.e., the number of characters). The NEAR PERFECT PHYLOGENY problem (NPPP) has a matrix M as input and asks for a minimum cost Steiner tree whose leaves are taken from the species (i.e., the rows of M) and all species label some vertices of the tree. By the previous argument on the optimum, the cost of any solution can be expressed as $m+q$, where q (which is always positive) is called the *penalty*. Notice that a penalty can be a back or a recurring mutation, since we have no way to distinguish or prioritize among those.

The first result in this setting is the $O(nm^q 2^{q^2 r^2})$ time algorithm [12] which draws upon some of the idea of the first fixed-parameter algorithm for the perfect phylogeny [1] to find a solution with penalty at most q, if such a tree exists. Unfortunately such time complexity makes the algorithm impracticable; especially the m^q factor limits the usefulness only to very small values of q. From a theoretical point of view, the main question left open in [12] is whether the NPPP admits an FPT algorithm when the parameters are q and r, r being the maximum number of states of any character.

The question has been answered positively in [26] for the binary perfect phylogeny, that is in the case $r = 2$. That case is especially important in theory and in practice. In fact, the study of the perfect phylogeny problem has shown that ideas originating from the 2-state case have, in due time, percolated up to 3-state and 4-state cases and then up to r-state case, for any fixed r. Therefore, the binary state algorithm is a strong hint that an FPT algorithm exists for any fixed r. From a practical point of view, most of available data are binary or can be transformed into binary characters via opportune clustering, therefore the algorithm of [26], which has $O(72^q + 8^q nm^2)$ time complexity, can be applied.

We will briefly sketch the main ideas of the algorithm of [26], which follows a randomized divide-and-conquer approach where, at each stage, a conflicting character c is picked at random, and then c is allowed to mutate only once in the tree. Since c mutates only once, the Steiner tree instance is partitioned into two subtrees T_0 and T_1, according the state that the species assumes in c. Then two vertices r_0, r_1 are chosen at random respectively from T_0 and T_1 (notice that r_0, r_1 might be Steiner vertices, so we cannot sample directly from the leaves or the species). A new edge (r_0, r_1) is created and labeled by the character c. Then the algorithm recurses on T_0 and T_1, by guessing no more than q edges overall and checking that, at the end, the conflict graph is sufficiently small to be solved via exhaustive enumeration.

The correctness of the algorithm mainly derives from the observation that at most q characters can mutate more than once. Therefore when the conflict graph is large the random choice of c has a high probability of being correct (i.e., there is exists a solution where c mutates once), while if the conflict graph is small then the optimal solution can be computed via brute force. The analysis of the time complexity is quite involved, as efficiently computing the vertices r_0, r_1 requires some combinatorial properties of Buneman graphs [25] (which are related to Steiner trees). The aforementioned $O(72^q + 8^q nm^2)$ time complexity is for the derandomized version of the algorithm. If we settle for finding the optimal solution with probability at least 8^{-q}, then the time complexity can be lowered to $O(18^q + 8nm^2)$.

A related problem studied in [24] is $H(p,q)$-NPP, where the input is a set of genotypes and we want to compute a phylogeny where the vertices are labeled with haplotypes so that (i) at most p sites can mutate, each at most q times (i.e. have at most q homoplasy events) and (ii) the set of haplotypes labeling the vertices are able to explain the input genotypes. In [24] is presented an algorithm for $H(1,q)$-NPP, that is when only one character is allowed to have at most q recurrent mutations. That algorithms nicely complements that of [26], where no restriction on the number of affected character exists, and is based on the analysis of the conflict graph, mainly on the property that the character with recurrent mutations must be the only one with two adjacent characters in the conflict graph.

6 Open problems

This work presents some generalizations of the perfect phylogeny model motivated by recent biological applications that investigate evolution as a character-based process. The availability of many genomic and proteomic data makes the use of genetic attributes or biological markers quite appealing in evolution analysis, thus giving even more importance to applying computationally efficient parsimony models. On the other hand, there is a huge gap between tractable and NP-hard parsimony models that needs to be filled. In fact, one extreme is the perfect phylogeny, which has linear time solution but has only a few specific biological applications. On the other hand we have models such as Dollo or Camin-Sokal parsimony models, which are often too generic from a biological viewpoint and computationally impracticable. A middle ground is occupied by the persistent perfect phylogeny model, for which some efficient and practical algorithms have been recently presented [3], while finding specific applications such as protein networks, domains analysis [23, 30]. However this research direction still needs to be explored. In particular, finding a polynomial time algorithm for the persistent phylogeny is still an open problem and the novelty of the algorithm of [3] hints that even more practical approaches are possible, even for some optimization versions of the problem deserving to be investigated. It must be pointed out that the persistent model is useful in detecting persistent characters that could be excluded from the evolutionary reconstruction process. In fact, having computational tools to detect characters that should be included or not in the parsimony model analysis lead to improve the correctness of the tree that is built from such characters [23]. From a theoretical point of view, the investigation of variants of the perfect phylogeny or restrictions of the Dollo parsimony, different from those presented here, is still an important research direction. In particular, the tree of complexes problem discussed in this work, reveals that there may be interesting and strong connections between graph theory and parsimony models representing the evolutionary relationships between functional modules in a protein network. To conclude, characterization of structural properties of protein networks as well as of the overlap graph of characters seems to be a promising novel direction for building parsimony models in a more efficient and biologically meaningful way.

Acknowledgments

PB and APC are supported by the Fondo di Ateneo 2011 grant “Metodi algoritmici per l’analisi di strutture combinatorie in bioinformatica”. GDV are supported by the Fondo di Ateneo 2011 grant “Tecniche algoritmiche avanzate in Biologia Computazionale”. PB, GDV and RD are supported by the MIUR PRIN 2010-2011 grant “Automi e Linguaggi Formali: Aspetti Matematici e Applicativi”, code H41J12000190001. TMP is supported by the Intramural Research Program of the National Institutes of Health, National Library of Medicine.

References

1. R. Agarwala and D. Fernández-Baca. A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed. *SIAM J. Comput.*, 23(6):1216–1224, 1994.
2. P. Bonizzoni. A linear time algorithm for the Perfect Phylogeny Haplotype problem. *Algorithmica*, 48(3):267–285, 2007.
3. P. Bonizzoni, C. Braghin, R. Dondi, and G. Trucco. The binary persistent perfect phylogeny. *Theoretical Computer Science*, 454:51–63, 2012.
4. P. Bonizzoni, G. Della Vedova, R. Dondi, and J. Li. The haplotyping problem: An overview of computational models and solutions. *J. of Computer Science and Technology*, 18(6):675–688, 2003.
5. J. Camin and R. Sokal. A method for deducting branching sequences in phylogeny. *Evolution*, 19:311—-326, 1965.
6. L. L. Cavalli-Sforza and A. W. F. Edwards. Phylogenetic analysis. models and estimation procedures. *American journal of human genetics*, 19(3 Pt 1):233, 1967.
7. 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.
8. T. Dobzhansky. Nothing in biology makes sense except in the light of evolution. *American Biology Teacher*, 35(3):125–129, 1973.
9. A. W. F. Edwards and L. L. Cavalli-Sforza. The reconstruction of evolution. *Heredity*, 18, 1963.
10. J. Felsenstein. *Inferring Phylogenies*. Sinauer Associates, 2004.
11. S. Felsner, V. Raghavan, and J. Spinrad. Recognition algorithms for orders of small width and graphs of small dilworth number. *Order*, 20:351–364, 2003.
12. D. Fernández-Baca and J. Lagergren. A polynomial-time algorithm for near-perfect phylogeny. *SIAM J. Comput.*, 32(5):1115–1127, 2003.
13. L. Foulds and R. Graham. The steiner problem in phylogeny is np-complete. *Advances in Applied Mathematics*, 3(1):43 – 49, 1982.
14. M. Garey and D. Johnson. *Computer and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, 1979.
15. M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.
16. D. Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge, 1997.
17. 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.
18. J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182:105–142, 1999. 10.1007/BF02392825.
19. S. Kannan and T. Warnow. A fast algorithm for the computation and enumeration of perfect phylogenies. *SIAM J. on Computing*, 26(6):1749–1763, 1997.
20. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.
21. I. Peer, T. Pupko, R. Shamir, and R. Sharan. Incomplete directed perfect phylogeny. *Siam Journal on Computing*, 33(3):590–607, 2004.
22. T. Przytycka, G. Davis, N. Song, and D. Durand. Graph theoretical insights into dollo parsimony and evolution of multidomain proteins. *Journal of Computational Biology*, 13(2):351—-363, 2006.
23. 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.

24. R. V. Satya, A. Mukherjee, G. Alexe, L. Parida, and G. Bhanot. Constructing near-perfect phylogenies with multiple homoplasy events. In *ISMB (Supplement of Bioinformatics)*, pages 514–522, 2006.
25. C. Semple and M. Steel. *Phylogenetics*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, USA, 2003.
26. S. Sridhar, K. Dhamdhere, G. Blelloch, E. Halperin, R. Ravi, and R. Schwartz. Algorithms for efficient near-perfect phylogenetic tree reconstruction in theory and practice. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 4(4):561 –571, oct.-dec. 2007.
27. A. Subramanian, S. Shackney, and R. Schwartz. Inference of tumor phylogenies from genomic assays on heterogeneous samples. *Journal of Biomedicine and Biotechnology*, 2012:1–16, 2012.
28. W. T. Tutte. An algorithm for determining whether a given binary matroid is graphic. *Proceedings of the American Mathematical Society*, 11(6):905–917, 1960.
29. J. Zheng, I. B. Rogozin, E. V. Koonin, and T. M. Przytycka. Support for the Coelomata Clade of Animals from a Rigorous Analysis of the Pattern of Intron Conservation. *Mol. Biol. Evol.*, 24(11):2583–2592, 2007.
30. E. Zotenko, K. S. Guimarães, R. Jothi, and T. M. Przytycka. Decomposition of overlapping protein complexes: A graph theoretical method for analyzing static and dynamic protein associations. *Algorithms for Molecular Biology*, 7(1):1–11, 2006.