Algorithms for Phylogeny Reconstruction in a New Mathematical Model

Gabriele Lenzini

Department of Computer Science
University of Pisa
Corso Italia, 40
56126 Pisa, Italy
e-mail lenzini@di.unipi.it

Silvia Marianelli

Department of Mathematics
University of Pisa
Via F. Buonarroti, 2
56127 Pisa, Italy

Abstract

The evolutionary history of a set of species is represented by a tree, called a *phylogenetic tree* or phylogeny, whose structure depends on precise biological assumptions about the evolution of species. Problems related to phylogeny reconstruction (i.e., finding a tree representation of information regarding a set of items) are widely studied in computer science. Most of these problems have been found to be *NP-hard*, but they can sometimes be solved polynomially if appropriate restrictions on the structure of the tree are fixed. The aim of this paper is to summarize the most recent problems and results in phylogeny reconstruction, and to introduce an innovative tree model, the Phylogenetic Parsimonious tree (*PP tree*), which is justified by a biologically significant hypothesis. Using the *PP tree*, two problems are studied: the existence and the reconstruction of a tree both when sequences of characters and partial order on interspecies distances, are given. We prove complexity results that confirm the hardness of this class of problems.

1 Introduction to phylogenetic problems

The evolutionary history of a set of species is usually described by a rooted tree called a *phylogenetic tree*, or *phylogeny*, formally defined as follows:

Definition 1.1 *Let S be a set of species. A* phylogenetic tree *T is a tree (V, E) where:*

- *V is a set $\{1, \ldots, m\}$ of nodes, such that $V \supseteq S$; S are at least the leaves of T, and nodes in $V - S$ stand for extinct species.*
- *E is a set $\{(i, j) : i, j \in V\}$ of edges, and the presence of an edge $(i, j) \in E$ means that j directly descends from i.*

Finding a tree-representation of information about a set S of items is an important algorithmical problem. It has been related to phylogeny reconstruction since the 1960s when, for the first time, computers were used to infer phylogenetic relationships in *numerical taxonomy* [25]. Many different instances of this problem have been defined and studied, and this experience has shown that:

1. most of them are *NP-complete* or *NP-hard*;
2. sometimes the existence of polynomial algorithms depends on restrictions imposed upon the structure of the tree.

1.1 Biological data models

Data can came directly from both DNA and protein analysis [21]; for this reason a species set can be mathematically described as

- a set of character sequences;
- a matrix of distances.

A *character* is a bit of information that assume a finite number of states. For example in a DNA sequence, a character is a single nucleotide: it has four possible states A, G, C, T. Each species is identified by a sequence (i.e., a vector) of characters.

Definition 1.2 *A* species *is a vector $(c_1, \ldots, c_k)$ where c_i is one of the $m+1$ states of the i-th character; we can assume that $c_i \in A = \{0, \ldots, m\}$, $\forall\, i \in \{1, \ldots, k\}$.*

Note: if the evolutionary age of each state, the *polarity*, is known the oldest of the state is associated with 0, the next with the 1, and so on up to the youngest.

An *interspecies distance* d_{ij} is a non-negative real number expressing a measure either of a genetic similarity or a genetic distance between i and j [21]. Distances are usually given by a square, species $\times$ species, symmetric matrix M, such that $M[i,j] = d_{ij}, \forall\ i,\ j$. [1] As any species i has always null distance from itself, we assume $M[i,i] = 0$.

1.2 Evolutionary hypotheses

Phylogenetic problems are usually based on biological hypotheses. Two of the most common ones are:

- *irreversibility of acquired characters* hypothesis: for each character, state i evolves only into the next $(i+1)$, and it cannot change back;
- *parsimony* hypothesis: the most reliable (i.e., likely) phylogenetic tree, called *parsimonious trees*, is that which has a minimum number of extinct species.

Usually the choice of a tree model is justified by the assumption that a certain hypothesis is true.

2 Background in phylogeny reconstruction

We now summarize same recent problems and results in the phylogeny reconstruction. A distinction can be made in terms of the data model used.

2.1 Character sequences

As we saw in Section 1.1, a species can be a vector $(c_1, \ldots, c_k)$ such that $\forall i \in \{1, \ldots, k\}$, $c_i \in A = \{0, \ldots, m\}$. Two classes of problem can be defined:

- *character compatibility* problems (or *perfect phylogeny* problems);
- *parsimony* problems.

In the *character compatibility* class we find the following problem:

[1] We indicate with d_{ij} the single distance between i and j, while $(d)_{ij}$ indicates the class of distances $d_{ij}, \forall\ i,\ j$.

Definition 2.1 ([29]) *A set S of species is* compatible *if it is possible to find a tree $T = (V, E)$, named* perfect phylogeny, *such that:*

1. *$S \subseteq V \subseteq A^k$, that is each node in V is labelled with a vector of k character states;*
2. *species in S lie on the leaves of T;*
3. *$\forall c_i$ and $\forall a \in A$, the set of nodes $n = (c_1, \ldots, c_k)$ such that $c_i = a$, induces a connected sub-graph of T (see Figure 1).*

Condition 2 in definition 2.1 describes a widely used model of a phylogenetic tree, but others are possible; condition 3 expresses the irreversibility of the acquired characters hypothesis.

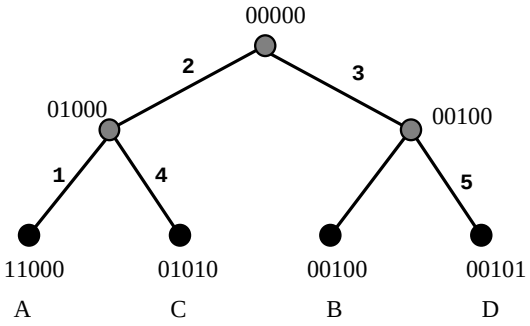

Figure 1: A perfect phylogeny for species $A = (11000)$, $B = (00100)$, $C = (01010)$ and $D = (00101)$. In the example only binary characters are used, and on the edges the indix of the character is reported that has evolved from 0 to 1.

Definition 2.2 (Perfect Phylogeny problem) *Given a set S of N species decide whether or not S is compatible.*

The Perfect Phylogeny problem is *NP-complete* ([2, 26]). An $O(\ m^{k+1}\ (k+1)^k + Nk^2\ )$ algorithm exists ([22]), which is polynomial when the number k of characters is bounded by a constant; in particular a linear-time algorithm is known when only three characters are used ([14, 13]). In the case of binary characters ($m = 2$), there exists an optimal $O(kN)$ time algorithm ([12]). An $O(Nk^2)$ time algorithm is known for trinary ($m = 3$) characters in [6], and an $O(N^2k)$ time algorithm for quaternary ($m = 4$) characters was found in [15].

The problems defined in the *parsimony* class differently use the parsimony hypothesis: the general target is finding a *parsimonious tree* (Section 1.2). In order to define the class of parsimony problems more easily, we assume that a species is identified by one single character c instead of a vector.

Definition 2.3 (Parsimony problem) *Let S be a species set. The most* parsimonious tree, *is a tree $T = (V, E)$ such that:*

1. *$V \subseteq A$, that is every node of T is labelled with a character state;*
2. *species in S lie on the leaves of T;*
3. *it minimizes the following quantity:*

$$\sum_{(i,j)\in E} L(i,j)$$

where $L(i,j)$ is the function "length of the edge (i,j)", and usually expresses the cost of evolution from state i to state j. If this evolution is not biologically possible $L(i,j)$ is set to ∞.

Note: in order to minimize only the number of dead species, just set $L(i,j) = 1\ \forall i,j$.

Different instances of the parsimony problem arise when varying the function L ([28]). In general, if k characters are used, we can easily extend the definition, but a further condition is needed:

4. $\forall (i,j) \in E$, with $i = (c_1, \ldots, c_k)$ and $j = (d_1, \ldots, d_k)$, i and j differ exactly in one position.

All problems in this class are known to be *NP-complete* ([5]).

2.2 Distances

The target in this class is to find an edge-weighted tree which exactly, or approximately, *realizes* a given distance matrix.

Definition 2.4 *Let $T = (V, E)$ an edge-weighted tree. The* tree-distance *d_{ij}^T between two nodes i and j in V is the length of the path P_{ij} from i to j, calculated as the sum of edge-lengths on P_{ij}.*

Definition 2.5 (Phylogeny reconstruction from distances) *Let M be a square $n \times n$ matrix of non negative reals, and S a set of n species. Determine an edge-weighted phylogenetic tree $T = (V, E)$ for S, such that $\forall i, j \in S$, $d^T_{ij} = M[i, j]$. If such a T exists, we say that it* realizes *M.*

Given a matrix M, the existence of a tree T that realizes M is a property of the matrix M itself, known as *additivity*; in other words if the matrix is additive then there exists (a unique [30]) T such that $d^T_{ij} = M[i, j]$. It is known that, given an $n \times n$ matrix M, it is possible to verify whether it is additive, and build the tree in $O(n^2)$ time [4]. The problem, as defined in definition 2.5, is far from describing a real situation. In fact, biological distances are rarely additive and a tree T that realizes a matrix distance does not exist. For this reason it is usually requested that the tree-distance d^T_{ij} *best approximates* $M[i, j]$.

Definition 2.6 (Phylogeny approximated reconstruction) *Let M be a square $n \times n$ matrix of non negative reals, and S a set of n species. Determine an edge-weighted phylogenetic tree $T = (V, E)$ for S, such that $\forall i, j \in S$, $d^T_{ij} \approx M[i, j]$.*

Various mathematical definitions for $\approx$ have been proposed, and most of the related optimal problems are shown *NP-hard* (for a good survey see [7]).

In [16] a new data model has recently been proposed related with distances, where only a partial order on distances is known. Since the set of tree-distances $(d^T)_{ij}$ is totally ordered, a new class of problem can be defined.

Definition 2.7 (Phylogeny reconstruction from partial order) *Given a set S of species, and a partial order $<_p$ on $(d)_{ij}$ distances, find an edge weighted phylogeny T for S, such that the total order $<_t$ on $(d^T)_{ij}$ is a topological sort of $<_p$, that is $\forall i, j, h, k : d_{ij} <_p d_{hk} \Longrightarrow d^T_{ij} <_t d^T_{hk}$.*

If T exists we say that T is *consistent* with the partial order. In [16] the partial order is described by a set of *experiments*, and an experiment is conducted in the following model:

Definition 2.8 ([16]) *Given a triple species i, j and k an* Ordering Model (OM) experiment *is a partial or total order on the triple of distances d_{ij}, d_{ik} and d_{jk}, with $=$, $<$ and $>$ explicitly indicated.*

For example an *OM experiment* on i, j and k can give the result $d_{ij} < d_{ik} = d_{jk}$. In [16] the following tree models were defined:

- **unweighted edge tree** T (i. e., with edges of length 1), such that a species in S lies only on the leaves of T, and without internal nodes of degree 2 (except for the root).
- **weighted edge tree** T, such that species in S lies only on the leaves of T.

The choice of bounding the degree of nodes (those corresponding to dead species) is justified in [30], and its goal is to avoid paths of nodes of degree 2 which are the same as a weighted edge. The existence problem for this model is:

Definition 2.9 (Existence from partial orders [16]) *Given a set S of species, and a partial order $Exp(S)$ on $(d)_{ij}$ distances, decide whether exists an unweighted phylogeny T for S, consistent with $Exp(S)$, such that species in S lies only on the leaves of T, and without internal nodes of degree 2 (except for the root).*

Fact 2.10 ([16]) *The problem* existence from partial orders *of an unweighted tree is* NP-complete.

The reconstruction can be performed by an algorithm of time complexity $O(N^3)$, supposing that existence problem has affirmative answer.

Fact 2.11 ([16]) *The existence of a weigthed tree problem is* NP-complete, *while the relative reconstruction problem is still open.*

3 A new tree model

In this section we introduce a new tree model based on a hypothesis many biologists agree upon. This hypothesis, which we call the *natural death (ND) hypothesis*, is the following:

Hypothesis of natural death. A dead species cannot have a living ancestor, that is the oldest species must disappear first.

The concept of *living* species, versus *dead* species, refers to the present or past existence of a known species; on the other hand dead species are unknown a priori. The ND hypothesis justifies a new general tree model where living species can lie on internal nodes too; these nodes must also induct a forest of subtrees (Figure 2). On the basis of the ND hypothesis we define the following models of *parsimonious phylogenetic forest* and *parsimonious phylogenetic tree.*

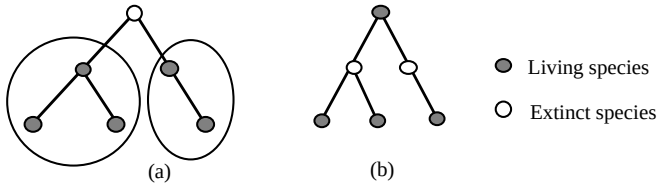

Figure 2: (a)A tree that agrees with the ND hypothesis; subtrees are shown in circles. (b) A tree that does not agree with the ND hypothesis (b).

Definition 3.1 *A* parsimonious forest *is a forest of living species with the minimum number of trees, that is where any species at the root of a tree can not have a living ancestor.*

Definition 3.2 *A* phylogenetic forest *is a forest of living species such that there exists a path of only dead species from each root to a particular species, called Root.*

Definition 3.3 *A* parsimonious-phylogenetic forest *(in short* PP forest*) is a phylogenetic and parsimonious forest.*

Figure 3 shows an example of each type of forest. Finally:

Definition 3.4 *A* parsimonious-phylogenetic tree *(in short* PP tree*) is a phylogenetic tree such that the set of nodes labelled with living species inducts a* PP forest*.*

By using the *PP tree* we stipulate the parsimony hypothesis as follows: before considering dead species to justify a descent, we look for a living ancestor. In addition, we are also interested in the *Optimal PP tree*, the one with the minimum number of dead species. Two fundamental problems arise:

Definition 3.5 (Existence of a *PP tree*) *Given a species set S, decide whether there exists a* PP tree *for S.*

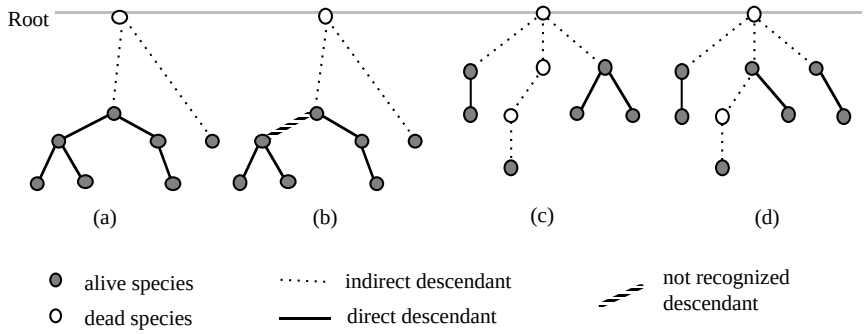

Figure 3: (a) Parsimony and (b) non parsimony forest. (c) Phylogenetic and (d) non phylogenetic forest.

Definition 3.6 (Reconstruction of a *PP tree*) *Supposing that the existence problem has an affirmative answer, give to give an algorithm that builds the* PP tree.

Note: deciding the existence of a *PP tree* is the same as deciding the existence of a *PP forest* for S.

Our models have general properties that are independent of the data that describes a species.

Proposition 3.7 *Let be S a species set. If there exists a phylogenetic forest F for S, then there exists a* PP forest *for S.*

Proof. Let F be a phylogenetic forest for S. If F is parsimonious then F is a *PP forest*; otherwise there is at least one root r in F which can be directly derived from a living species. Suppose F is composed of trees $T_1, \ldots, T_h$ whose roots are respectively $r_1, \ldots, r_h$. Let

$$L = \{i / r_i \text{derives from a living species}\}$$

we modify F by executing the following command:

for each $i \in L$ **do** connect the root r_i to its living father

This modification has no effect on the phylogenetic part of F. The forest F' we obtain is a *PP forest*, since F' is both phylogenetic (by hypothesis) and parsimonious (by construction). ■

Proposition 3.8 *Let S be a species set. If there exists a* PP forest *F for S, then every parsimonious forest is also a phylogenetic forest (*i.e.*, it is a* PP forest*).*

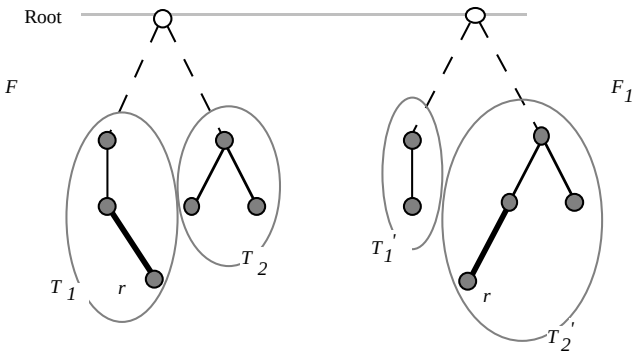

Figure 4: $F = \{T_1, T_2\}$ and $F_1 = \{T_1', T_2'\}$ are equally parsimonious: the species s may derive from two different living fathers.

Proof. Let F be a *PP forest* for S, and F_1 be a parsimonious forest for S, such that $F_1 \neq F$. F and F_1 must have the same number of trees. In addition a species in F is a root if and only if it is a root in F_1. In fact, a parsimonious forest has the minimum number of trees, which is equal to the number of those living species that have no living ancestors. In other words F_1 differs from F only in terms of those species that are internal nodes and can be derived from more than one living species (see Figure 4). Lastly, if F is a phylogeny forest then F_1 is phylogenetic too, and so F_1 is a *PP forest*. ■

Proposition 3.8 ensures that the choice of a living father, from a set of equally possible ones, has no influence on the next search for a path of dead species.

Proposition 3.9 *If there exists a phylogenetic forest for S, then every parsimonious forest is a phylogenetic forest (*i.e.*, it is a* PP forest*).*

Proof. Let F be a phylogenetic forest for S. From F we can obtain a *PP forest* F (proposition 3.7), and because a PP forest exists, every parsimonious forest is a *PP forest* (proposition 3.8). ■

Proposition 3.9 suggests a general algorithm to solve the existence problem. First we construct a parsimonious forest F by joining all couples of species s and t, where s derives directly from t; afterwards, when no more connections between living species are possible, we test whether there exists a path of dead species. F is a *PP forest* only if the test is successful.

3.1 Existence and reconstruction from characters

In this section, we study existence and reconstruction problems when species are vectors of k integers belonging to $\{0, 1, \ldots, m\}$. We assume that the irreversibility of acquired characters hypothesis is true, that is character states are ordered by age, from 0 (the oldest), till m (the youngest). We call that model $\{0 \to 1 \to \ldots \to m\}$, and we start our analysis in the simpler model $\{0 \to 1\}$.

3.1.1 Model $\{0 \to 1\}$

The biological interpretation of this model is easy: each character can assume only two states 0 and 1, of which 0 is the older. Species are identified with vectors of k binary digits. As a Consequently we can represent at most 2^k different species, but the number N of living species (those from set S) is much less than 2^k, in real cases. All the 2^k possible species are considered as vertexes on a k-cube in a k-dimensional space, and they can be grouped into levels.

Definition 3.10 *Given a species* $s = (c_1, \ldots, c_k)$ *the* level *of* s, lev(s), *is* $\sum_{i=1}^{k} c_i$

This k-cube is the supporting structure on which to build our PP tree. The vertex $r = (0, \ldots, 0)$ at level 0 is the known *a priori* Root. To completely characterise a *PP tree* in this model we should define when one species derives from another.

Definition 3.11 *A species* s derives *from a species* t *if and only if* $lev(s) = lev(t) + 1$

The existence problem is defined as follows:

Definition 3.12 (Problem P_1) *Let be* $H = \{1, 0\}^k$ *a* k*-cube,* S *a set of* N *species such that* $S \subseteq \{1, 0\}^k$. *Decide whether there is a subtree* $T = (V_T, E_T)$ *of* H, *such that* T *is a* PP tree *for* S.

We have:

Theorem 3.13 *Problem* P_1 *can be solved in* $O(kN^2)$ *time, when* $N \in O(2^k)$.

Proof. The proof is given by the following algorithm. It is divided into two phases: the *construction* of a parsimonious forest and a test for the *existence* of a path of dead species. ■

Algorithm 1

Phase 1: Construction of a parsimonious forest F
given a set $S = \{s_1, \ldots, s_N\}$ of species.
let $root(T)$ **be**
    a function that returns the species at the root of tree T;
let $addSon(T_1, T_2)$ **be**
    a function that returns a tree obtained by connecting
    the root of the tree T_2 to the root of tree T_1;
begin
$F := S$;
(F is composed of N one–node trees, one for each species in S)
    let $maxL$, $minL$ **be**
    the maximum, minimum level of species in S;
    for $L := maxL$ **downto** $minL + 1$ **do**
        for each species s in S at level L **do**
        if there exists a species t at level $L - 1$ such
        that s *derives* from t **then**
            begin
            let T_1, T_2 **be**
                the tree in F such that $root(T_2) = s$ and $root(T_1) = t$;
            $F := F - \{T_1, T_2\}$;
            $T1 := addSon(T_1, T_2)$;
            $F := F \cup T_1$;
            end
        else skip
(s is a root of some tree in F);
end

Proposition 3.14 *The time complexity of phase 1 is* $O(kN^2)$

Algorithm 2

Phase 2: Determination of the existence of a subset of dead species
that connects each root of F to $r = (0, \ldots, 0)$.
begin
let S' **be**
    the set of roots of forest F built in phase 1;
(basic cases)
    case F **of**
    (F is a phylogeny forest)
        • is composed of only one tree,
        or all the roots are at the same level,**success**;
        • is composed of more than one tree and
        the species $r = (0, \ldots, 0)$ is in S,

```
        there does not exist any phylogeny forest; fail;
    otherwise
(looking for dead species connections)
        begin
    [1] <built the k-cube from level 0 to
        the level of the root at maximum level in F >
        if there exist a path that connect each root to r = (0, ..., 0)
        then (F is a phylogeny forest) success;
        else (F is not a phylogeny forest) fail;
        end;
    endcase;
end.
```

Proposition 3.15 *Time complexity of phase 2 is $O(k2^k)$. The 2^k factor is due to instruction* [1]*; only if $N \in O(2^k)$ is the cost of phase 2 polynomial in N.*

If phase 2 is successful then F is a *PP forest*, that is a PP tree exists. Of all possible *PP trees*, we are interested in the one with the minimum number of dead species, the *Optimal* PP tree, because it satisfies the parsimony hypothesis (Section 1.2). Let us define the related decisional problem:

Definition 3.16 (Problem P_2) *Let be $H = \{1, 0\}^k$ a k-cube, S a set of N species such that $S \subseteq 1, 0^k$, and b a positive integer. There exists a subtree $T = (V_T, E_T)$ of H, such that T is a PP tree for S, and $|V_T - S| \leq b$?*

We now demonstrate that the problem is *NP-complete* by building a polynomial reduction from the following problem known as Vertex Cover, and shown *NP-complete* in [10].

Definition 3.17 (Vertex Cover (VC)) *Let $G = (V_G, E_G)$ a graph, $d \leq |V_G|$ an integer value. There exists a subset V of V_G, such that $|V| \leq d$, and for each $(i, j) \in E_G$, we have $i \in V$ or $j \in V$?*

We have:

Theorem 3.18 *P_2 is* NP-complete.

Proof. First of all P_2 is in NP. In fact suppose that $T = (V_T, E)$ is a solution of the P_2 problem. We can easily test if it is a solution by visiting the tree T and verifying that every living node has living children. This test can obviously be performed in $O(N^2)$ time. Let $x = [G = (V_G, E_G), d]$ an instance of the VC problem; let us construct an instance $f(x) = [S, k, b]$ of P_2 as follows (Figure 5):

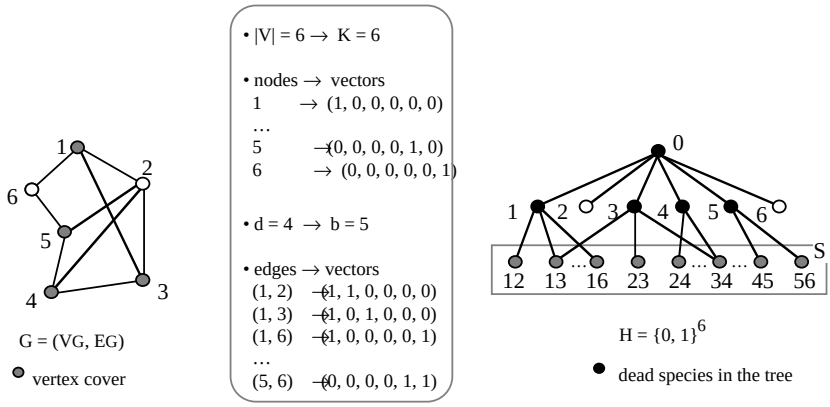

Figure 5: Reduction from an instance of the VC problem to an instance of the P_2 problem.

- $k = |V_G|$;
- $S = \{h \in \{0,1\}^k /$
  $\forall (i,j) \in E_G$, h has 1's in position i and position j, and 0's elsewhere$\}$;
- $b = d + 1$;

The function f can obviously be computed in polynomial time. Suppose now that x was a *yes* instance for the VC problem, and let V be the vertex cover of G, such that $|V| \leq k$. The PP Optimal tree $T = (V_T, E_T)$ can be constructed from $f(x)$ as follows:

(V_T) First $V_T = S$; then for each vertex $v \in V$, add to V_T the vector h at level 1 with a 1 in the position v. Finally add to V_T the root $r = (0, \ldots, 0)$;

(E_T) $E_T = \{(i,j)/j \in S$ and $i = (c_1, \ldots, c_k)$ is a vector at level 1 such that it has 1 in the same position as j has (if more than one choice for i is possible, choose arbitrarily)$\}$. Finally add to E_T the edges which connect each considered vector at level 1 to the root $r = (0, \ldots, 0)$.

The obtained tree $T = (V_T, E_T)$ is a *PP tree* such that $|V_T - S| \leq d+1 = b$; $f(x)$ is thus a *yes* instance of the P_2 problem. Suppose now that $f(x)$ was a *yes* instance of P_2 problem, and $T = (V_T, E_T)$ was the PP tree such that $|V_T - S| \leq b$. Since every vector in S at level 2, their fathers should lie at level 1, and they are, in turn, connected to the root $r = (0, \ldots, 0)$. These vectors at level 1 are at most d. Defining V as the set of vertexes

corresponding to these vectors, V is a vertex cover of G such that $|V| \leq d$, and x is a *yes* instance of VC. ■

Corollary 3.19 *Reconstructing a PP Optimal tree is an* NP-hard *problem.*

Corollary 3.19 conjectures that any algorithm to reconstruct a *PP Optimal tree* has an exponential time complexity unless P = NP. This confirms the difficulty of phylogeny problems: even in the simple model $\{0 \rightarrow 1\}$, the reconstruction problem is intractable. Even if we relax the target of finding an optimal solution, we still have to resort to heuristics. We now describe an algorithm which uses a heuristic function of local search, which tries to minimize the overall number of dead species, by minimizing the insertion of dead species at each step. The *PP tree* is constructed bottom-up and, at each step, a couple of species is connected to the common ancestor selected by the heuristic. Phase 1 of the algorithm is the same as algorithm 2; we report only Phase 2 assuming that F is a *PP forest* and $S' = \{$all the species at root of trees in $F\}$ of cardinality r.

Algorithm 3

Phase 2: reconstruction of a *PP tree*.
let S' *be* $\{s_1, \ldots, s_r\}$;
begin
initialisation
$h := 1$; $S_h := S'$, $NmDeadSp := 0$, $N_h := r$;
while $N_h > 1$ **do**
begin
determining the common ancestor
for i, $j := 1$ **to** N_h **do**
let D_{ij} **be** the common ancestor of s_i, s_j
obtained as follows: $D_{ij}[r] := s_i[r] \vee s_j[r]$, $\forall r = 1, \ldots, k$;
let D_u **be** the ancestor D_{uv} at maximum level among those in D_{ij};
updating variables
$S_h = S_h - \{s_u, s_v\} \cup D_h$;
calculating the number of dead species inserted depending on D_h
case D_h **of**
$\bullet D_h \in \{s_u, v_j\} : N_h := N_h - 1$;
$NmDeadSp := NmDeadSp + \max\{lev(s_u), lev(s_v)\} - lev(D_h) - 1$;
$\bullet D_h \in S_h : N_h := N_h - 2$;
$NmDeadSp := NmDeadSp + (lev(s_u) - lev(D_h) - 1)$
$+(lev(s_v) - lev(D_h) - 1)$;
$\bullet D_h \notin S_h : N_h := N_h - 1$;
$NmDeadSp := NmDeadSp + 1$
$+(lev(s_u) - lev(D_h) - 1) + (lev(s_v) - lev(D_h) - 1)$;

```
            endcase;
        endwhile;
    end.
```

Fact 3.20 *The algorithm has $O(kr^3)$ time complexity.*

The error committed by algorithm 4 was studied in [19], but with no definitive results. An interesting test was done using that algorithm with sequences obtained from a PCR analysis of DNA on Primates [3]. the correct relationship was inferred, and this encouraging practical result made us decide to report the algorithm.

3.1.2 Model $\{0 \rightarrow 1 \rightarrow \ldots \rightarrow m\}$

This model is the natural extention of the previous $\{0 \rightarrow 1\}$. Each character can assume $m+1$ different states of which 0 is the oldest. Both consistence and reconstruction problems can be defined in this model using a generalization of k-cube ([19]). We also reached similar *NP-complete* results with similar methods, which we omit.

3.2 Existence and reconstruction from partial orders

We now perform the same analysis as in Section 3.1 using a partial order on distances. We recall from Section 2.2 that a partial order on distances is given by a set $Exp(S)$ of OM experiments on triple species in S. Let $N = |S|$; although the input size is $O(N^3)$ we refer to N as the main parameter.

3.2.1 Existence of a PP tree

We now consider the *PP tree* model introduced in Section 3. As was done in [16], we bound the degree of internal nodes standing for dead species to be greater than 2. We define the following existence problem of a *PP tree*:

Definition 3.21 (Problem P_3) *Given a set S of species, and a partial order $Exp(S)$ on $(d)_{ij}$ distances, decide whether exists an unweighted PP tree T for S, consistent with $Exp(S)$, such that T has no nodes standing for dead species of degree 2 (except for the root).*

We have:

Theorem 3.22 *Problem P_3 is* NP-complete

Proof. The reduction is from the problem P which was shown to be *NP-complete* in [16]. First we prove that the problem is in NP. We can test whether a tree $T = (V, E)$ is a solution for the problem. We can test in $O(|V|)$ whether T is a PP tree visiting T, and in $O(N^4)$ whether T is consistent with $Exp(S)$. In fact, for each i, j and k we calculate, in $O(N)$, the length of paths P_{ij}, P_{jk} , P_{ik} and compare their order with the OM experiment on triple i, j and k which are at most $N(N-1)(N-2)$. Let $x = [S, Exp(S)]$ be an instance of P. We construct an instance $f(x) = [S', Exp'(S')]$ of P_3 as follows:

- $S' = S \cup \{x_1, \ldots, x_N\}$, where x_i are new living species;
- $Exp'(S') = Exp(S) \cup (\bigcup_{i=1}^{N} A_i)$, where A_i are new OM Experiments on species x_i, s_i and y, $\forall y \in S' - \{x_i, s_i\}$, with the result $d_{s_i x_i} < d_{s_i y} < d_{y x_i}$.

Function $f(x)$ can obviously be calculated in polynomial time. Suppose that x is a yes instance for P, and $T = (V, E)$ the phylogeny that satisfies $Exp(S)$. The tree $T' = (V', E')$ solution of $f(x)$ is the following (Figure 6):

- $V' = V \cup \{x_1, \ldots x_N\}$; i.e. we add the species x_i as new leaves;
- $E' = E \cup (\bigcup_{i=1}^{N} A_i)$; i.e. we connect each x_i to s_i.

T' is consistent with $Exp'(S')$. In fact distances $(d'_T)_{ij}$ concerning the new leaves x_i are consistent with the new experiment A_i, while the remaining $(d'_T)_{ij}$ concerns only species in S, which are consistent with experiments in $Exp(S)$ by hypothesis.

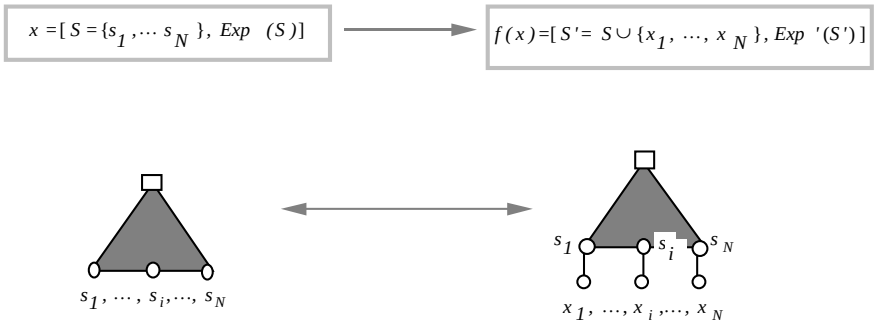

Figure 6: Example of reduction from an instance of problem P_1 to problem P.

Suppose now that $f(x)$ is a yes instance for P_3, and $T' = (V', E')$ the phylogeny that satisfies $Exp'(S')$. T' has all x_i at leaves, and each x_i is

connected to the node s_i. The tree $T = (V, E)$ solution of x is the following (see Figure 6): $V = V' - \{x_1, \ldots, x_N\}$; that is we cut off the x_i leaves; $E = E' - \cup(\bigcup_{i=1}^{N} A_i)$; we cut off all edges $(s_i,\ x_i)$ which are redundant. T is consistent with $Exp(S)$: in fact the distances $(d'_T)_{ij}$, concerning the remaining species (those in S), are consistent with experiments in $Exp(S)$ by hypothesis. ■

3.2.2 Reconstruction of a PP tree

We now study the reconstruction problem of a *PP tree* by assuming that the related consistency problem has an affirmative answer. We have:

Theorem 3.23 *Assuming that the consistency problem has an affirmative answer, the reconstruction problem for a PP tree T – such that it has no nodes (except for the root) standing for dead species has degree 2 – can be solved with an algorithm of $O(N^4)$ time complexity .*

The proof is given by the following algorithm. The frame of the algorithm is the same as algorithm 2 in Section 3.1. It is divided into two phases: first we construct a parsimonious forest F, and then we connect each root of F to Root, thus obtaining a *PP tree*. We have:

Definition 3.24 *A* supernode *V is either a tree, or a single node. We refer to a supernode both as a tree and as the set of species on its nodes.*

Algorithm 4

Phase 1: construction of a parsimonious forest F,
given a set $S = s_1, \ldots, s_N$ of species.
let $Exp(S)$ **be**
    a set of OM experiments on triple species in S;
let $root(T)$ **be**
    a function that returns the species at the root of the supernode T;
let $derive(T_1, T_2)$ **be**
    a function that uses information from $Exp(S)$ to test whether
    a supernode T_1 *derives* from a supernode T_2;
we discuss this function in the next section
let $addSon(T_1, T_2)$ **be**
    a function that returns a new supernode obtained by connecting
    the root of supernode T_2 to the root of the supernode T_1;
begin
$F := S$;
the forest F is composed of a one-node supernode, one for each species in S

main loop
    repeat
looking for living-living connections
        for each $i \in F$ **do**
            let $SonSet_i$ **be**
            the set $\{T/\text{derive}(T, i), \forall T \in F - \{i\}\}$;
        for each $SonSet_i \neq \emptyset$ **do**
        begin
            $F := F - \{i\}$;
            foreach $T \in SonSet_i$ **do**
                $i := addSon(T, i)$;
            $F := F - SonSet_i \cup \{i\}$;
        end;
    until $SonSet_i = \emptyset, \forall i$;
end.

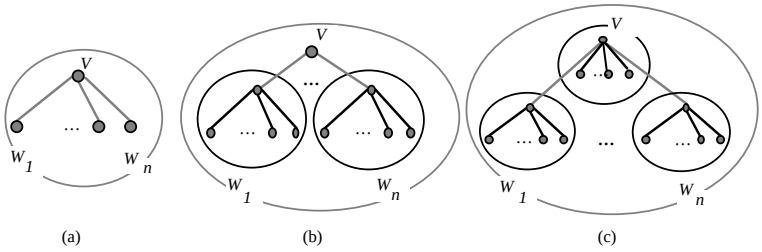

Figure 7: Examples of connections of supernodes of living species W_i and V using $addSon$, when $\text{derive}(W_i, V)$ is true.

Algorithm 5

Phase 2. Reconstruction of a PP tree T;
let F **be**
    $\{T_1, \ldots, T_h\}$ the forest (of supernodes) from phase 1;
let $areBrth(T_1, T_2)$ **be**
    a function that uses information from $Exp(S)$
    to test whether the supernode T_1 and the supernode T_2 are brothers,
    that is whether they are descendants of a common dead ancestor;
(we discuss this function in the next section)
let $mkNewSpNd(T)$ **be**
    a function that returns a new supernode whose root r is a new node,
    and each supernode in T is connected to the root r;
begin
    if F is a tree **then return** F
    else

```
        repeat
(looking for root-root connections)
            for each i ∈ F do
                let SonSet_i be the set
                {T/derive(T, i)∀T ∈ F − {i}};
(looking for connections to a new dead species)
            for each i ∈ F do
                let SiblSet_i be
                the set obtained by transitive closure on
                areBrth(i, T), where T ∈ F − i;
(root-root connections have priority over those involving a new dead species)
(in this way we introduce less dead species)
            if ∃SonSet_i ≠ ∅
            then
                for each SonSet_i ≠ ∅ do
                begin
                    F := F − {i};
                    foreach a ∈ SonSet_i do
                    i := addSon(a, i);
                    F := F − SonSet_i ∪ {i};
                end
            else
                for each SiblSet_i ≠ ∅ do
                begin
                    F := F − SiblSet_i;
                    s := mkNewSpNd(SiblSet_i);
                    F := F ∪ {s};
                end
        until ∀i (SonSet_i = ∅)and(SiblSet_i = ∅);
    return F
end.
```

3.2.3 How to implement *derive()* and *areBrth()* functions

To define the boolean functions derive and *areBrth* between two supernodes we use three binary relations on supernodes EQ, LT and GT.

Definition 3.25 *Let V be a supernode.*

root(V) *is the root of V;*

rep(V) *is the representative of V, is a node labelled with the closest living species to the root;*

$\delta(V)$ *is the distance in number of edges from* root(V) *and* rep(V).

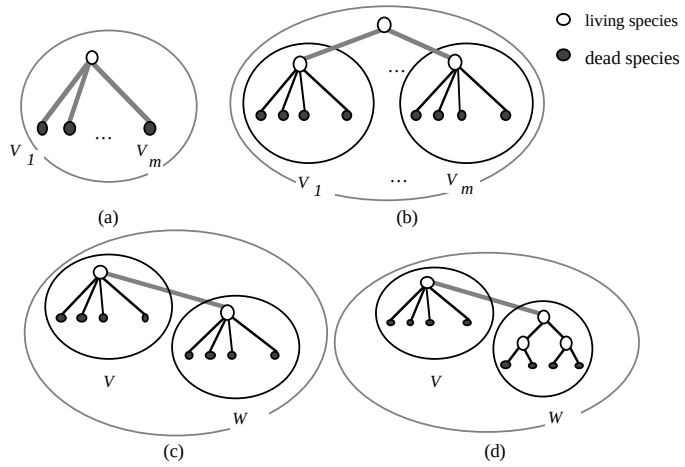

Figure 8: Examples of connections of supernodes V_i, by *mkNewSpNd* when *areBrth* is true, (a) and (b). Connection root-root by *addSon* when $\mathsf{derive}(W, V)$ is true, (c) and (d).

Definition 3.26 *Let V_1 and V_2 be two supernodes and r_1, r_2 their representatives.*

LT $(V_1, V_2) \in LT$ *if and only if there exists a node labelled with an living species j, $j \in S - (V_1, V_2)$ such that in the OM experiment on triple j, r_1, and r_2 we have $d_{jr_1} < d_{jr_2}$;*

EQ $(V_1, V_2) \in EQ$ *if and only if there exists a node labelled with an living species j, $j \in S - (V_1, V_2)$ such that in the OM experiment on triple j, r_1, and r_2 we have $d_{jr_1} = d_{jr_2}$;*

GT $(V_1, V_2) \in GT$ *if and only if there exists a node labelled with an living species j, $j \in S - (V_1, V_2)$ such that in the OM experiment on triple j, r_1, and r_2 we have $d_{jr_1} > d_{jr_2}$.*

We have:

Lemma 3.27 ([16]) *Let V_1, V_2 be two supernodes. It is possible to calculate* $\mathsf{areBrth}(V_1, V_2)$ *in $O(N)$ time.*

The test consists in assuming that $\mathsf{areBrth}(V_1, V_2)$ is true, connecting V_1 and V_2 as in Figures 8 (a) and 8 (b), and looking for confirmation or contradiction in the results of available experiments. If an inconsistency is found $\mathsf{areBrth}(V_1, V_2)$ becames false.

Lemma 3.28 *Let W, V be two supernodes. If $\delta(W) = \delta(V)$ then* $\mathsf{derive}(W, V)$ *if and only if $(W, V) \notin EQ \cup LT$.*

Proof. Assuming that W really derives from V; then they are connected by one edge in T (Figures 7 and 8 (c)). Thus every OM experiment on j, $\mathsf{rep}(W)$, and $\mathsf{rep}(V)$, with $j \in S - (W, V)$ should give:

$$d_{j\,\mathsf{rep}(W)} > d_{\mathsf{rep}(V)\mathsf{rep}(W)}$$

$$d_{j\,\mathsf{rep}(V)} > d_{\mathsf{rep}(V)\mathsf{rep}(W)}$$

$$d_{j\,\mathsf{rep}(V)} > d_{j\,\mathsf{rep}(W)}.$$

The opposite can be deduced by absurd assuming that $(W, V) \notin EQ \cup LT$, and that W does not derive from V: it is always possible to find a species j, $j \in S - (W, V)$ that makes $(W, V) \in EQ \cup LT$. ■

Lemma 3.29 *Let W, V be two supernodes. If $\delta(W) > \delta(V)$ then* $\mathsf{derive}(W, V)$ *can be tested in $O(N)$ time.*

Proof. If $(W, V) \in EQ \cup LT$ then $\mathsf{derive}(W, V)$ is false: in fact by absurd if W derives from V , they are connected by one edge in the tree, and thus for every species $j \in S - (W \cup V)$ we have:

$$d_{j\,\mathsf{rep}(W)} > d_{j\,\mathsf{rep}(V)}$$

that is $(W, V) \in EQ \cup LT$. If $(W, V) \in EQ \cup LT$ we assume that $\mathsf{derive}(W, V)$ is true, and we connect their roots with an edge as in figure 8 (d): we look for confirmation or contradiction in the available experiments. Let be L the length of path P from $\mathsf{rep}(W)$ to $\mathsf{rep}(V)$:

- if L is an even number then let x be the node in the middle of P, and j be a species in the subtree with root x that does not contain $\mathsf{rep}(V)$. Then $\mathsf{derive}(W, V)$ is true if and only if: $d_{j\,\mathsf{rep}(W)} = d_{j\,\mathsf{rep}(V)}$.
- if L is an odd number then let (x, y) be the edge in the middle of P, and j_1, j_2 be two leaves, in the subtree whose root is x that does not contain $\mathsf{rep}(V)$, and in the subtree whose root is y that does not contain $\mathsf{rep}(W)$ respectively; then $\mathsf{derive}(W, V)$ is true if and only if $d_{j_1\,\mathsf{rep}(W)} > d_{j_1\,\mathsf{rep}(V)}$ and $d_{j_2\mathsf{rep}(W)} > d_{j_2\mathsf{rep}(V)}$.

Because these tests consist in looking for a path in a tree of $O(N)$ nodes they can be done in $O(N)$ time (the tree could be totally unbalanced). ■

Finally we have:

Theorem 3.30 *Algorithm 6 has time complexity* $O(N^4)$

Proof. From the list of algorithm 6 we can see that at each step the number of tests using derive() or areBrth() is $O(N^2)$, that is as many tests as there are couple of supernodes. Because each test can be performed in $O(N)$ time the resulting complexity is $O(N^3)$. We now estimate the number of steps in the worst case, which occurs when the tree searched for is the following:

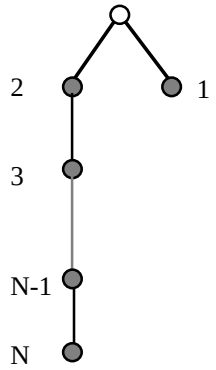

Figure 9: The tree whose reconstruction needs the greatest number of steps.

In fact at each step (except for the last one) we add only one edge, thus having O(N) steps. The resulting complexity is $O(N^4)$. ■

4 Conclusion

The *PP tree* model represents a biologically significant generalization of the usual phylogenetic tree, but similar NP-completeness results arise, when sequences and partial orders are used. Our results confirm the hardness of problems related to phylogeny reconstruction.

References

[1] R. AGARWALA and D. FERNANDEZ-BACA. A polynomial-time algorithms for the perfect phylogeny problem when the number of character is fixed. In *proc. of IEEE Symposium on Foundations of Computer Science*, pages 140–147, 1993.

[2] H. BODLAENDER, M. FELLOW, and T. WARNOW. Two strikes against the perfect phylogeny. *Lecture Notes in Computer Science*, pages 273–283, 1992.

[3] C. CARLÀ CAMPA, G. LENZINI, S. MARIANELLI, S. CROVELLA, G. ARDITO, M. BUIATTI, F. LUCCIO, and L. GALLENI. Phylogeny reconstruction in primates under a new mathematical model using pcr amplification sequences. manuscript.

[4] C. CULBERSON and R. RUDNICKI. A fast algorithm for construction trees from distances metrics. *Information Processing Letters*, (30):215–220, 1989.

[5] W. H. E. DAY. Computationally difficult of parsimony problems in phylogenetic systematics. *journal of Theoretical Biology*, (103):429–438, 1983.

[6] A. DRESS and M. A. STEEL. Convex tree realizations of partitions. *Applied Mathematical Letter*, (5):3–6, 1992.

[7] M. FARAK, S. KANNAN, and T. WARNOW. A robust model for finding optimal evolutionary trees. In *proc. of ACM SIAM Symposium on Theory of Computing*, pages 137–145, 1993.

[8] M. FARAK, L. LAWER, and T. WARNOW. Determining the evolutionary tree. In *proc. of ACM SIAM Symposium on Discrete Algorithms*, pages 475–484, 1990.

[9] J. FELSENSTEIN. *User's Manual for PHYLIP (Phylogeny Inference Package)*, 1988. v.v. 3.1.

[10] M. R. GAREY and D. S. JOHNSON. *Computers and intractability. A guide to the theory of NP-completeness.* W. H. Freeman & C., 1979. San Francisco.

[11] D. GIANNOTTA. Problemi computazionali nella ricostruzione di alberi filogenetici. Master's thesis, Università di Pisa, ITALY, 1993. Tesi di Laurea.

[12] D. GUSFIELD. Efficient algorithms for inferring evolutionary trees. *Networks*, (21):19–28, 1991.

[13] R. IDURY and A. SCHAFFER. Triangulating three-colored graphs in linear time and linear space. *SIAM Journal of Discrete Mathematics*, (6):289–293, 1993.

[14] S. KANNAN and T. WARNOW. Triangulating 3-coloured graphs. *SIAM Journal of Discrete Mathematics*, 5(2):249–256, 1992.

[15] S. KANNAN and T. WARNOW. Inferring evolutionary history from dna sequences. *SIAM Journal of Computing*, 2(23):713–737, 1994.

[16] S. KANNAN and T. WARNOW. Tree reconstruction from partial orders. *SIAM Journal of Computing*, 3(24):511–519, 1995.

[17] G. LENZINI. Ricostruzione di filogenesi da ordinamenti parziali su distanze interspecie. Master's thesis, Università di Pisa, ITALY, 1994. Tesi di Laurea.

[18] S. E. LURIA, S. J. GOULD, and S. SINGER. *A view of life*. Cummings, Publ. Comp., 1981. California.

[19] S. MARIANELLI. Alcuni problemi compuatazionali nell'evoluzione biologica. Master's thesis, Università di Pisa, ITALY, 1993. Tesi di Laurea.

[20] R. D. MARTIN. *Primate Origin and Evolution: a Phylogenetic Reconstruction*. Cambrige University Press, 1990. G.B.

[21] M. M. MAYAMOTO and J. CRACRAFT, editors. *Phylogeny Analysis of DNA Sequences*. Oxford University Press, 1991. USA.

[22] F. R. McMORRIS, T. WARNOW, and T. WIMER. Triangulating vertes-coloured graphs. *SIAM Journal of Computing*, 7(2):296–306, 1994.

[23] C. H. PAPADIMITRIOU and K. STEIGLITZ. *Combiantorial Optimization: Algorithms and Complexity*. Prentice Hall Inc., 1982. New Jersey.

[24] M. RIDLEY. *Evolution*. Blackwer Scientific Publication, 1993.

[25] S. S. SNEATH and R. R. SOKAL. *Numerical Taxonomy*. W. H. Freeman & C., 1973. San Francisco.

[26] M. A. STEEL. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*, (9):91–116, 1992.

[27] D. L. SWOFFORD. *User's Manual for PAUP (Phylogenetic Analysis Using Parsimony)*, 1989. v.v. 3.0.

[28] D. L. Swofford and G. J. Olsen. Phylogenetic inference. In D. M. Hillis, C. Mortiz, and B. K. Mable, editors, *Molecular Systematics*. Sinauer Associates Inc., Sunderland, Massachussetts, 1996.

[29] T. WARNOW. Tree compatibility and inferring evolutionay history. *Journal of Algorithms*, (16):388–407, 1994.

[30] P. WINKLER. The complexity of metric realization. *SIAM Journal of Discrete Mathematics*, 1(4):552–559, 1988.