March 22th, 2007

Finding a largest subset of rooted triples identifying a tree is an NP-hard task
Research Report LIRMM - RR-07010

Sylvain Guillemot and Vincent Berry

Equipe Méthodes et algorithmes pour la Bioinformatique, LIRMM Université Montpellier 2 – CNRS 161, rue Ada, 34392 Montpellier, France
{sguillem,vberry}@lirmm.fr
tel: 00 33 4 67 41 85 48
fax: 00 33 4 67 41 85 00

Abstract

Supertree methods are used to build comprehensive phylogenies from source trees with overlapping sets of leaves. Ranwez et al. (2007) recently proposed a polynomial-time method outputting supertrees that verify attractive mathematical properties to obtain a consensual summary of a collection of source trees. The topological information of the source trees that is considered by the method is the set of rooted triples they define on the leaf set. The method finds a subset of these rooted triples that *identify* a supertree (Semple and Steel, 2003) and return the corresponding supertree. Biologists are interested in supertrees that are as resolved as possible (Bininda-Emonds, 2004b). Thus, given a set of rooted triples on a set of taxa, Ranwez et al. (2007) asked whether it is possible to find in polynomial time a largest subset of rooted triples identifying a tree. In this short note, we show that solving this problem is an NP-hard task.

Keywords: computational biology, supertrees, trees, rooted triples, mathematical properties, NP-hardness

Contact: vberry@lirmm.fr

1 Introduction

Phylogenies are invaluable tools in various areas of biology to understand the evolution of genes and taxa. Trees that incorporate an exhaustive sampling of taxonomic biodiversity provide crucial information about systematics, genomics, and diversification patterns of species (Davies et al., 2004). Large trees can be built according to a divide-and-conquer approach, first inferring source trees from biological data sets covering only part of the taxa, then assembling the source trees in a comprehensive tree, called a *supertree* (Bininda-Emonds, 2004a).

To assess the quality of a supertree, it is most useful to have measures but also properties characterizing to what extent the supertree is a good representation of the input collection (Steel et al., 2000). Ranwez et al. (2007) defined formal properties requesting that supertrees do not contradict the source trees, singly or in combination, and that supertrees do not propose arbitrary resolutions. This implies that the supertree does not favor a particular resolution among several conflicting alternatives (*non-contradiction property*, denoted PC), but also that every piece of phylogenetic signal displayed in the supertree is induced by one or several source topologies (*induction* property, denoted PI). Indeed, "novel relationships displayed by a supertree (...) are worrying if they are not implied by combinations of the input trees" (Wilkinson et al., 2005).

Until now, supertree methods have focused on mathematical criteria defining how well a supertree represents a collection of source trees and trying to find the supertree optimizing such a fitness criterion. Yet, biologists also need inferred supertrees to be as informative as possible in order to elucidate phylogenetic relationships of the taxa they consider. However, just trying to maximize the amount of information in a supertree leads to infer a binary tree spanning the whole set of considered taxa, without regard to the amount of information present in the data. Such a supertree is not desirable if it contains arbitrary groups proposed for the sake of obtaining a maximally resolved supertree. Thus, a reasonable goal is to obtain a supertree that is as resolved as possible but that only displays defendable relationships with regards to the source trees. The PI and PC properties of Ranwez et al. (2007) define a formal framework in which this request from biologists can be investigated. We study here the hardness of the corresponding computational problem. We show that finding the most informative supertree satisfying PI and PC is an NP-hard problem, when the amount of information in a tree is measured by the number of rooted triples it induces.

2 Definitions

Definitions and notations used for trees and their topological description mainly follow those used in the book of Semple and Steel (2003). We only consider rooted trees, due to the impossibility of supertree methods to fulfill different desirable properties listed in Steel et al. (2000) when considering unrooted trees. Given a tree T, we denote $L(T)$ the set of taxa associated to its leaves. More generally, given a collection $\mathcal{T}$ of trees, $L(\mathcal{T})$ denotes the set of taxa appearing in at least one tree of $\mathcal{T}$. Given two trees T, T' on the same leaf set ($L(T) = L(T')$), we say that T *refines* T', denoted $T \trianglerighteq T'$, whenever T contains all clades of T'. In other words, T can be transformed into T' by collapsing some of its internal edges (i.e., merging the respective extremities of these edges)

A rooted tree on three leaves a, b, c has only three possible binary shapes, called *rooted triples* (or *triplets*) and denoted $ab|c$, resp. $ac|b$, resp. $bc|a$, depending on the innermost clade (ab, resp. ac, resp bc). Given a rooted triple t, we denote $\bar{t}$ any of the two other rooted triples on the same set of leaves. Alternatively, a tree on three leaves can be a star tree, i.e. a unique internal node connected to the leaves. It is well known that a rooted tree T with more than three leaves can be equivalently described by a set of rooted triples: we note $rt(T)$ the set of rooted triples homeomorphic to subtrees of T connecting three leaves. Given a collection $\mathcal{T}$ of trees, we denote $rt(\mathcal{T}) = \bigcup_{T_i \in \mathcal{T}} rt(T)$ the set of rooted triples present in theses trees. Note that it is possible that $rt(\mathcal{T})$ contains two rooted triples t and $\bar{t}$, namely when $\mathcal{T}$ hosts two incompatible trees. Clearly, two such rooted triples can not be combined into a single supertree of the collection. Given a set $\mathcal{R}$ of triples, $L(\mathcal{R})$ denotes the set of taxa appearing in at least one tree in $\mathcal{R}$.

A tree T is said to *display* a set $\mathcal{R}$ of rooted triples when $\mathcal{R} \subseteq rt(T)$; moreover, T *strictly displays* $\mathcal{R}$ if additionally $L(T) = L(\mathcal{R})$. A set $\mathcal{R}$ of rooted triples is *compatible* if there is a tree T that displays $\mathcal{R}$. To find a tree displaying $\mathcal{R}$, it is useful to rely on the fact that some triples of the tree are *induced* by $\mathcal{R}$: a compatible set $\mathcal{R}$ of rooted triples *induces* a rooted triple t, denoted $\mathcal{R} \vdash t$, if and only if $\mathcal{R} \cup \{\bar{t}\}$ is not compatible, or equivalently if any tree T that displays $\mathcal{R}$ contains t. For instance, any tree displaying $\{ab|c, bc|d\}$ has to also display the triple $ac|d$, that is $\{ab|c, bc|d\} \vdash ac|d$. Bandelt and Dress (Bandelt and Dress, 1986) and Dekker (Dekker, 1986) were among the first ones to investigate such induction rules. The set of all rooted triples induced by a compatible set $\mathcal{R}$ is called the *closure* of $\mathcal{R}$ and is denoted $cl(\mathcal{R})$. Source trees considered for supertree building are sometimes incompatible, in which case the set of rooted triples considered is incompatible. Nonetheless, we can characterize the set of rooted triples induced by these collections by extending the preceding definition: we will say that a set $\mathcal{R}$ of rooted triples *induces* a rooted triple t when there is a compatible subset $\mathcal{R}' \subseteq \mathcal{R}$ that induces t. We now recall the definition of the PC and PI properties given in Ranwez et al. (2007).

Definition 1 *Let T be a tree, and $\mathcal{T}$ be a collection of trees,*

- *Let $\mathcal{R}(T,\mathcal{T})$ be the set of rooted triples in $rt(\mathcal{T})$ for which T proposes a resolution. More formally, $\mathcal{R}(T,\mathcal{T}) = \{ab|c \in rt(\mathcal{T})$ such that $\{ab|c, ac|b, bc|a\} \cap rt(T) \neq \emptyset\}$.*
- *We say that T verifies **PI** for $\mathcal{T}$ if and only if for all $t \in rt(T)$, it holds that $\mathcal{R}(T,\mathcal{T}) \vdash t$.*
- *We say that T verifies **PC** for $\mathcal{T}$ if and only if for all $t \in rt(T)$ and all $\bar{t}$, it holds that $\mathcal{R}(T,\mathcal{T}) \nvdash \bar{t}$.*

The set $\mathcal{R}(T,\mathcal{T})$ corresponds to all topological information present in the collection $\mathcal{T}$, or induced by it, that is related to the topological information present in the supertree T. Note that when the source trees are incompatible, it is possible that this subset of rooted triples contains two different rooted triples for the same three taxa. For some compatible collections of trees, it is possible to find a supertree that both displays all the rooted triples of the collection and is refined by all other possible supertrees. More formally, a set $\mathcal{R}$ of rooted triples is said to *identify* a tree T if and only if T strictly displays $\mathcal{R}$ and T is refined by every tree T' that strictly displays $\mathcal{R}$. In this case, $\mathcal{R}$ is said to be *identifying.* A set $\mathcal{R}$ can identify at most one tree. Thus when $rt(\mathcal{T})$ identifies a tree T, this tree can be seen as a *canonical* representation of all possible supertrees of the collection $\mathcal{T}$.

Proposition 1 (Ranwez et al. (2007)) *[Ranwez et al. (2007)] A tree T verifies PI and PC for a collection $\mathcal{T}$ of trees if and only if $\mathcal{R}(T,\mathcal{T})$ identifies T.*

Ranwez et al. (2007) proposed a polynomial-time method that outputs supertrees verifying PI and PC and showing a non-trivial level of resolution on several biological data sets. But the question remains to know whether a method can be designed that produces supertrees satisfying PI and PC and containing *as much resolution as possible*, e.g. resolves as much rooted triples as possible. More precisely, using Proposition 1, we ask for a method that, given any collection $\mathcal{T}$, proposes a supertree T such that $\mathcal{R}(T,\mathcal{T})$ identifies T and $\mathcal{R}(T,\mathcal{T})$ has maximum size over all such subsets of $rt(\mathcal{T})$. Such a subset of $rt(\mathcal{T})$ is called a maximum identifying subset of rooted triples (MIST). The difficulty of this problem can not be simply deduced from previously known results for optimization problems on rooted triples. Indeed, the MIST problem is a middle term between the NP-hard problem that consists of finding a maximum-sized *compatible* subset of rooted triples (Bryant, 1997) and the polynomial-time problem that asks for the maximum-sized *tree-like* subset of rooted triples[1] (Berry and Gascuel, 2000; Bryant and Berry, 2001).

3 Finding the largest subset of rooted triples identifying a tree is an NP-hard problem

Given a collection $\mathcal{T}$ of source trees, we show here that finding the largest subset of $rt(\mathcal{T})$ that identifies a tree is an NP-hard problem. More formally, we consider the following problem:

> MAXIMUM IDENTIFYING SUBSET OF ROOTED TRIPLES (MIST)
> *Input:* a set $\mathcal{R}$ of rooted triples.
> *Output:* a subset $\mathcal{R}'$ of $\mathcal{R}$ that is identifying and that is of maximum size among the identifying subsets of $\mathcal{R}$.

[1]"tree-like" means that the sought subset of triples must *exactly* be the set of rooted triples of a tree. When the input set of triples contains only one triple for each set of 3-taxa the mentioned problem is solvable in polynomial-time.

We consider the particular case where $rt(\mathcal{T})$ contains at most one rooted triple for any set of 3 leaves in $L(\mathcal{T})$. The difficulty of MIST in this particular case implies its difficulty in the general case. To show that MIST is NP-hard, we first consider a variant of the problem on somewhat simpler objects: ordered partitions (defined in Sect. 3.1). Then we show that the identification problem for ordered partitions (MISG) is NP-hard (Sect. 3.2). Finally, we prove the NP-hardness of MIST by reduction from MISG (Sect. 3.3).

3.1 Ordered partitions

An *ordered partition* on L is a tuple $P = (L, \leq_P)$, where $\leq_P$ is a total preorder on L (that is a binary relation satisfying reflexivity, transitivity and totality). P can be seen as a sequence $s(P) = (L_1, ..., L_k)$, where the L_i's are the equivalence classes of the preorder, and such that we have $x <_P y$ whenever $x \in L_i, y \in L_j$ with $i < j$. The sequence $s(P)$ will be called the *representation* of P. The ordered partition on L with only one class will be denoted by 0_L, i.e. $s(0_L) = (L)$. The set of ordered partitions on L will be denoted by $\mathcal{O}(L)$.

We can define the relation of *refinement* on ordered partitions: given $P, P' \in \mathcal{O}(L)$, we say that P refines P', denoted by $P \trianglerighteq P'$, iff for every $x, y \in L$, $x <_{P'} y$ implies $x <_P y$. As in the case of trees, we can endow $\mathcal{O}(L)$ with a structure of join-semilattice: the join of two partitions P, P' is the partition $P'' = P \sqcap P'$ s.t. $\leq_{P''}$ is the transitive closure of $\leq_P \cup \leq_{P'}$.

When $\mathcal{S}$ is a set of ordered partitions, we will denote by $\mathcal{S} \uparrow$ the upward closure of $\mathcal{S}$ by the refinement relation, i.e. $\mathcal{S} \uparrow = \{P : \exists P' \in \mathcal{S}, P \trianglerighteq P'\}$. Likewise, we will denote by $\mathcal{S} \downarrow$ the downward closure of $\mathcal{S}$, i.e. $\mathcal{S} \downarrow = \{P : \exists P' \in \mathcal{S}, P' \trianglerighteq P\}$. We will also use the notations $\mathcal{S} \uparrow$ and $\mathcal{S} \downarrow$ when $\mathcal{S}$ is a set of trees.

In the context of trees, given a set of triples $\mathcal{R}$, we denote by $D(\mathcal{R})$ the set of trees T that strictly display $\mathcal{R}$, i.e. such that $L(\mathcal{R}) = L(T)$ and $\mathcal{R} \subseteq \mathcal{R}_T$.

In the context of ordered partitions, we will redefine the notions of "display" and "identifies" by analogy with the case of trees. The ordered partitions will play the role of trees, and the directed graphs will play the role of sets of rooted triples. The definitions are as follows. Let $G = (V, A)$ be a directed graph, and let $P \in \mathcal{O}(V')$. We say that P *strictly displays* G iff $V = V'$ and for every $(x, y) \in A$, we have $x <_P y$. We will denote by $D(G)$ the set of ordered partitions P which strictly display G. We say that G *identifies* P iff $P \in D(G)$ and is refined by every partition $P' \in D(G)$. We say that G is *identifying* if G identifies some $P \in \mathcal{O}(V)$.

Observe that the set $D(G)$ enjoys the following properties (which also hold in the case of trees):

- $D(G)$ is closed upwards: if $P \in D(G)$ and $P' \trianglerighteq P$ then $P' \in D(G)$; thus, there exists a set $\mathcal{S}$ st $D(G) = \mathcal{S} \uparrow$.
- saying that G identifies P is equivalent to saying that $D(G) = \{P\} \uparrow$; in this case, $D(G)$ is closed by $\sqcap$.

3.2 The MISG problem

The problem Maximum Identifying Subgraph (MISG) asks: given a directed graph $G = (V, A)$, find an identifying subgraph $G' \subseteq G$ with the largest number of edges. We proceed to show that:

Proposition 2 MISG *is* NP*-hard.*

Proof.[Sketch] We reduce from the problem Maximum Edge Biclique which asks: given a bipartite graph G, find a biclique (i.e. a complete bipartite graph) $G' \subseteq G$ with the largest number of edges. This problem was shown NP-hard in Peeters (2003).

The reduction is as follows. Suppose that G has bipartition X, Y, then we view G as a directed graph $G = (V, A)$, with $A \subseteq X \times Y$. We show that: for every $G' \subseteq G$, G' is identifying $\Leftrightarrow$ G' is a biclique or G' is empty. Fix a graph $G' \subseteq G$, then $G' = (V', A')$; let $X' = V' \cap X$ and $Y' = V' \cap Y$.
($\Leftarrow$): if G' is a biclique, then G' identifies P with representation (X', Y'): P displays G' obviously, and for any P' displaying G', we have $x <_{P'} y$ whenever $x \in X', y \in Y'$ (since G' is a biclique), and thus P' must refine P. If G' is empty, then G' identifies P with representation (V').
($\Rightarrow$): suppose that G' is identifying. To derive a contradiction, suppose that G' is neither a biclique nor an empty graph. Since G' is not empty, we have $X' \neq \emptyset$ and $Y' \neq \emptyset$. Consider the ordered partition P with representation (X', Y'): since G' is bipartite, obviously P displays G'. Now, since G' is not a biclique, there exists $u \in X', v \in Y'$ s.t. $(u, v) \notin A'$. Consider the ordered partition P' with representation $(X' - \{u\}, \{u, v\}, Y' - \{v\})$: then P' displays G'. Thus, we obtain two partitions P, P' which both display G'. Let $P'' = P \sqcap P'$, then P'' displays G', since G' is identifying. But we have $P'' = 0_{V'}$ by definition of $\sqcap$. We thus obtain that $0_{V'}$ must display G', which is impossible since G' is not empty. $\square$

3.3 The MIST problem

Proposition 3 *MIST is NP-hard.*

Proof. We reduce from MISG. Consider an instance I of MISG, this is a directed graph $G = (V, A)$. We construct an instance I' of MIST as follows: (i) the label set is $L := V \cup \{x\}$; (ii) for every $a = (u, v) \in A$, we define the triple $t_a = xu|v$. For every $F \subseteq A$, we define $\mathcal{R}_F = \{t_a : a \in F\}$ and $G_F = (V_F, F)$, where V_F is the set of vertices incident to an arc of F. We set $\mathcal{R} := \mathcal{R}_A$.

Obviously, the construction can be done in polynomial time. It remains to justify that the reduction is correct. To this end, we need to define a correspondence between the solutions of MISG and the solutions of MIST. Noting $\mathcal{P}(L)$ the set of rooted phylogenies with leaf set L, we say that a tree $T \in \mathcal{P}(L)$ is a *brush* iff there exists sets $V_1, ..., V_k$ s.t. $T = brush(x, V_1, ..., V_k)$, where $brush(x, V_1, ..., V_k)$ denotes the tree having k internal nodes $u_1, u_2, ..., u_k$ s.t. $\forall i = 1..k-1$ the children of u_i are $V_i \cup \{u_{i+1}\}$ and those of u_k are $V_k \cup \{x\}$.

We define a straightforward correspondence between ordered partitions and brushes:

- given an ordered partition $P \in \mathcal{O}(V)$ with representation $(V_1, ..., V_k)$, we define $Tree(P) := brush(x, V_1, ..., V_k)$.
- given a brush $T \in \mathcal{P}(L)$ s.t. $T = brush(x, V_1, ..., V_k)$, we define $Partition(T)$ as the partition with representation $(V_1, ..., V_k)$.

The correctness of the reduction is based on the the following observations: for any $F \subseteq A$,

1. for every $P, P' \in \mathcal{O}(V)$: let $T = Tree(P), T' = Tree(P')$, then $P \trianglerighteq P'$ iff $T \trianglerighteq T'$;
2. for every $P \in \mathcal{O}(V)$: let $T = Tree(P)$, then P displays G_F iff T displays $\mathcal{R}_F$;
3. let T be an element of $D(\mathcal{R}_F)$ minimal for $\trianglerighteq$: then T is a brush.

Lemma 1 *For every $F \subseteq A$, G_F is identifying $\Leftrightarrow$ $\mathcal{R}_F$ is identifying.*

Proof. ($\Rightarrow$): suppose that G_F identifies an ordered partition P. Let $T = Tree(P)$, we claim that $\mathcal{R}_F$ identifies T. Indeed, since $P \in D(G_F)$, by Observation 2, $T \in D(\mathcal{R}_F)$. Moreover, suppose that $T'' \in D(\mathcal{R}_F)$. Let $\mathcal{S} = \{T''\} \downarrow \cap D(\mathcal{R}_F)$, and let T' be an element of $\mathcal{S}$ minimal for $\trianglerighteq$. Observe that T' must also be a minimal element of $D(\mathcal{R}_F)$. Thus, Observation 3 applies, and we obtain

that T' is a brush. Now, we can define $P' = Partition(T')$: then $P' \in D(G_F)$ (by Observation 2). Since G_F identifies P, this implies that $P' \trianglerighteq P$. We obtain that $T' \trianglerighteq T$ by Observation 1. On the other hand, we have $T'' \trianglerighteq T'$ by definition of T', and we conclude that $T'' \trianglerighteq T$.
($\Leftarrow$): suppose that $\mathcal{R}_F$ identifies a tree T. Then T must be an element of $D(\mathcal{R}_F)$ minimal for $\trianglerighteq$. Thus, by Observation 3, T is a brush. We can then define $P = Partition(T)$, and we claim that G_F identifies P. Indeed, since $T \in D(\mathcal{R}_F)$, by Observation 2 $P \in D(G_F)$. Moreover, suppose that $P' \in D(G_F)$, and let $T' = Tree(P')$. Then by Observation 2, we have $T' \in D(\mathcal{R}_F)$. Since $\mathcal{R}_F$ identifies T, we have $T' \trianglerighteq T$. We conclude that $P' \trianglerighteq P$ by Observation 1. □

The result stated in this lemma suffices to end the proof of the proposition. □

References

Bandelt, H.-J. and A. W. M. Dress. 1986. Reconstructing the shape of a tree from observed dissimilarity data. Adv. in Appl. Math. 7:309–343.

Berry, V. and O. Gascuel. 2000. Inferring evolutionary trees with strong combinatorial evidence. Theor. Comput. Sci. 240:217–298.

Bininda-Emonds, O. 2004a. The evolution of supertrees. Trends Ecol. Evol. 19:315–322.

Bininda-Emonds, O. 2004b. Phylogenetic supertrees (combining information to reveal the tree of life) vol. 4 of *computational biology series*. Kluwer academic publishers.

Bininda-Emonds, O. and H. Bryant. 1998. Properties of matrix representation with parsimony analyses. Syst. Biol. 47:497–508.

Bryant, D. 1997. Building trees, hunting for trees and comparing trees: theory and method in phylogenetic analysis. Ph.D. thesis University of Canterbury, Department of Mathemathics.

Bryant, D. and V. Berry. 2001. A structured family of clustering and tree construction methods. Adv. in Appl. Math. 27:705–732.

Cotton, J., C. Slater, and M. Wilkinson. 2006. Discriminating supported and unsupported relationships in supertrees using triplets. Syst. Biol. 55:345–350.

Davies, T., T. Barraclough, M. Chase, P. Soltis, D. Soltis, and V. Savolainen. 2004. Darwin's abominable mystery: Insights from a supertree of the angiosperms. Proc. Natl. Acad. Sci. USA 101:1904–1909.

Dekker, M. 1986. Reconstruction methods for derivative trees. Master's thesis University of Amsterdam.

Goloboff, P. A. and D. Pol. 2002. Semi-strict supertrees. Cladistics 18:514–525.

Grunewald, S., M. Steel, and M. Swenson. 2006. Phylogenetic closure operations in phylogenetics submitted to Mathematical Biosciences.

Peeters, R. 2003. The maximum edge biclique is NP-complete. Discrete Appl. Math. 131:651–654.

Ranwez, V., V. Berry, S. Guillemot, P.-H. Fabre, and E. J. P. Douzery. 2007. Physic : a method for supertree inference combining induction and non-contradiction properties submitted to Systematic Biology.

Semple, C. and M. Steel. 2003. Phylogenetics vol. 24 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press.

Steel, M. A., A. W. M. Dress, and S. Böcker. 2000. Simple but fundamental limitations on supertree and consensus tree methods. Syst. Biol. 49:363–368.

Wilkinson, M., D. Pisani, J. Cotton, and I. Corfe. 2005. Measuring support and finding unsupported relationships in supertrees. Syst. Biol. 54:823–831.

Acknowledgments

This work has been supported by the "ACI Informatique-Mathématique-Physique en Biologie Moléculaire [ACI IMP-Bio]" and by the "action incitative BIOSTIC-LR".