J. Math. Biol. (2006) 53:340–364
10.1007/s00285-006-0011-4

Mathematical Biology

On the ancestral compatibility of two phylogenetic trees with nested taxa

Mercè Llabrés · Jairo Rocha · Francesc Rosselló · Gabriel Valiente

Received: 30 May 2005 / Revised: 9 May 2006 /
Published online: 6 July 2006

Abstract Compatibility of phylogenetic trees is the most important concept underlying widely-used methods for assessing the agreement of different phylogenetic trees with overlapping taxa and combining them into common supertrees to reveal the tree of life. The notion of ancestral compatibility of phylogenetic trees with nested taxa was recently introduced. In this paper we analyze in detail the meaning of this compatibility from the points of view of the local structure of the trees, of the existence of embeddings into a common supertree, and of the joint properties of their cluster representations. Our analysis leads to a very simple polynomial-time algorithm for testing this compatibility, which we have implemented and is freely available for download from the BioPerl collection of Perl modules for computational biology.

Keywords Phylogenetic tree · Compatibility · Topological embedding

Mathematics Subject Classification (2000) 05C05 · 92D15 · 92B10

1 Introduction

A rooted phylogenetic tree can be seen as a static description of the evolutionary history of a family of contemporary species: these species are located at the

M. Llabrés · J. Rocha · F. Rosselló
Department of Mathematics and Computer Science,
Research Institute of Health Science, University of the Balearic Islands,
07122 Palma de Mallorca, Spain

G. Valiente (✉)
Algorithms, Bioinformatics, Complexity and Formal Methods Research Group,
Technical University of Catalonia,
08034 Barcelona, Spain
e-mail: valiente@lsi.upc.edu

leaves of the tree, and their common ancestors are organized as the inner nodes of the tree. These interior nodes represent taxa at a higher level of aggregation or nesting than that of their descendents, ranging for instance from families over genera to species. Phylogenetic trees with nested taxa have thus all leaves as well as some interior nodes labeled, and they need not be fully-resolved trees and may have unresolved polytomies, that is, they need not be binary trees.

Often one has to deal with two or more phylogenetic trees with overlapping taxa, probably obtained through different techniques by the same or different researchers. The problem of combining these trees into a single supertree containing the evolutionary information of all the given trees has recently received much attention, and it has been identified as a promising approach to the reconstruction of the tree of life [2]. This information corresponds to evolutionary precedence, and hence it is kept when every arc in each of the trees becomes a path in the supertree.

It is well known that it is not always possible to combine phylogenetic trees into a single supertree: there are *incompatible* phylogenetic trees that do not admit their simultaneous inclusion into a common supertree. Compatibility for leaf-labeled phylogenetic trees was first studied in [14]. Incompatible phylogenetic trees can still be partially combined into a maximum agreement subtree [13]. Compatible phylogenetic trees, on the other hand, can be combined into a common supertree, two of the most widely used methods being matrix representation with parsimony [1,7] and mincut [5,11] and it is clear that, because of Occam's razor, one is interested in obtaining not only a common supertree of the given phylogenetic trees, but the smallest possible one. The relationship between the largest common subtree and the smallest common supertree of two leaf-labeled phylogenetic trees was established in [8] by means of simple constructions, which allow one to obtain the largest common subtree from the smallest common supertree, and vice versa.

While the question of whether or not two overlapping phylogenetic trees are compatible is a fundamental problem towards their combination into a common supertree, the presence of nested taxa introduces new forms of compatibility. As illustrated in Fig. 1, two compatible phylogenetic trees can become incompatible when nested taxa are taken into account.

The study of the compatibility of phylogenetic trees with nested taxa, also known as *semi-labeled trees*, was asked for in [6]. Polynomial-time algorithms were proposed in [3,9] for testing a weak form of compatibility, called *ancestral compatibility*, and a stronger form called *perfect compatibility*. Roughly, two or more semi-labeled trees are ancestrally compatible if they can be refined into a common supertree, and they are perfectly compatible if there exists a common supertree whose topological restriction to the taxa in each tree is isomorphic to that tree.

In this paper, we are concerned with the notion of ancestral compatibility of semi-labeled trees. In particular, we establish the equivalence between this notion and the absence of certain 'incompatible' pairs and triples of labels in the trees under comparison. We also prove the equivalence between ancestral compatibility and a certain property of the cluster representations of the

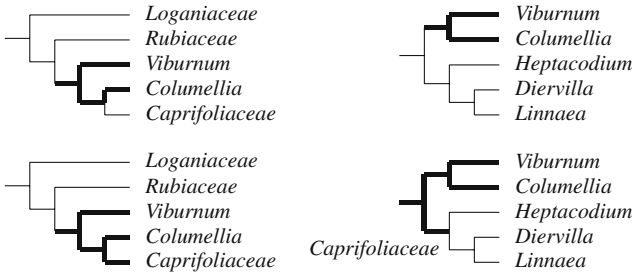

Fig. 1 Two compatible phylogenetic trees (*top*) obtained from studies S2x7x96c15c21c45 and S866 in TreeBASE, with their restriction to common taxa shown with *thick lines*. Two incompatible semi-labeled trees (*bottom*) obtained from the same studies, with incompatible clusters shown with *thick lines*. The incompatible triple of labels involves three taxa in one tree and two taxa plus one internal label in the other tree

trees. These equivalences lead to a new polynomial-time algorithm for testing ancestral compatibility of semi-labeled trees, which we have implemented and is freely available for download from the BioPerl collection of Perl modules for computational biology [12].

The rest of the paper is organized as follows. Basic notions and notation are recalled in Sect. 2. A notion of local compatibility as the absence of incompatible pairs and triples of labels is introduced in Sect. 3, together with some basic results about a relaxed notion of semi-labeled trees. Weak topological embeddings, and the notion of ancestral compatibility that derives from them, are studied in Sect. 4. In Sect. 5, the equivalence between local compatibility in the sense of Sect. 3 and ancestral compatibility in the sense of Sect. 4 is established, as well as a characterization in terms of cluster representations. The BioPerl implementation of the algorithm for testing compatibility of two semi-labeled trees is described in Sect. 6. Finally, some conclusions and further work are outlined in Sect. 7.

2 Preliminaries

Throughout this paper, by a *tree* we mean a *rooted tree*, that is, a directed finite graph $T = (V, E)$ with V either empty or containing a distinguished node $r \in V$, called the *root*, such that for every other node $v \in V$ there exists one, and only one, path from the root r to v. Recall that every node in a tree has in-degree 1, except the root, which has in-degree 0.

Henceforth, and unless otherwise stated, given a tree T we shall denote its set of nodes by $V(T)$ and its set of arcs by $E(T)$. The *children* of a node v in a tree T are those nodes w such that $(v, w) \in E(T)$. The nodes without children are the *leaves* of the tree, and we shall call *elementary* the nodes with only one child.

Given a path $(v_0, v_1, \ldots, v_k)$ in a tree T, its *origin* is v_0, its *end* is v_k, and its *intermediate nodes* are $v_1, \ldots, v_{k-1}$. Such a path is *non-trivial* when $k \geqslant 1$. We shall represent a path *from v to w*, that is, a path with origin v and end w, by

$v \rightsquigarrow w$. When there exists a path $v \rightsquigarrow w$, we say that w is a *descendant* of v and also that v is an *ancestor* of w. Every node is both an ancestor and a descendant of itself, through a trivial path.

Two non-trivial paths $(a, v_1, \ldots, v_k)$ and $(a, w_1, \ldots, w_\ell)$ in a tree T are said to *diverge* when the only node they have in common is their origin a. Notice that, by the uniqueness of paths in trees, it is equivalent to the condition $v_1 \neq w_1$. For every two nodes v, w of a tree that are not connected by a path, there exists one, and only one, common ancestor a of v and w such that there exist divergent paths from a to v and to w. We shall call it the *most recent common ancestor* of v and w. When there is a path $v \rightsquigarrow w$, we say that v is the *most recent common ancestor* of v and w.

3 $\mathcal{A}$-trees

Let $\mathcal{A}$ be throughout this paper a fixed set of labels. In practice, we shall use the first capital letters, $A, B, C, \ldots$, as labels.

Definition 1 *A semi-labeled tree over $\mathcal{A}$ is a tree with some of its nodes, including all its leaves and all its elementary nodes, injectively labeled in the set $\mathcal{A}$.*

To simplify several proofs, we shall usually allow the existence of unlabeled elementary nodes. This motivates the following definition.

Definition 2 *An $\mathcal{A}$-tree is a tree with some of its nodes, including all its leaves, injectively labeled in the set $\mathcal{A}$.*

We shall always use the same name to denote an $\mathcal{A}$-tree and the (unlabeled) tree that *supports* it. Furthermore, for every $\mathcal{A}$-tree T, we shall use henceforth the following notations:

- $\mathcal{L}(T)$ and $\mathcal{A}(T)$ will denote, respectively, the set of the labels of its leaves and the set of the labels of all its nodes.
- For every $v \in V(T)$, we shall denote by $\mathcal{A}_T(v)$ the set of the labels of all its descendants, including itself, and we shall call it, following [10], the *cluster* of v in T; if T is irrelevant or clearly determined by the context, we shall usually write $\mathcal{A}(v)$ instead of $\mathcal{A}_T(v)$. Notice that if there exists a path $w \rightsquigarrow v$, then $\mathcal{A}(v) \subseteq \mathcal{A}(w)$.
- We shall set

$$\mathcal{C}_{\mathcal{A}}(T) = \{\mathcal{A}_T(v) \mid v \in V(T)\}.$$

  Notice that $\emptyset \notin \mathcal{C}_{\mathcal{A}}(T)$ unless T is empty. If T is a semi-labeled tree over $\mathcal{A}$, then $\mathcal{C}_{\mathcal{A}}(T)$ coincides with the cluster representation [10] of T, up to the trivial cluster for the root of T. Consequently, even for $\mathcal{A}$-trees, we shall call $\mathcal{C}_{\mathcal{A}}(T)$ the *cluster representation* of T.

– For every $X \subseteq \mathcal{A}(T)$, we shall denote by $v_{T,X}$ the most recent common ancestor of the nodes of T with labels in X; when T is irrelevant or clearly determined by the context, we shall usually write v_X instead of $v_{T,X}$. Moreover, when X is given by the list of its members between brackets, we shall usually omit these brackets in the subscript. So, in particular, for every $A \in \mathcal{A}(T)$, we shall denote the node of T labeled A by $v_{T,A}$ or simply v_A. Notice that $\mathcal{A}(v_{T,X}) = X$ if and only if $X \in \mathcal{C}_{\mathcal{A}}(T)$.

We shall often use the following easy results, usually without any further mention.

Lemma 1 *Let T be an $\mathcal{A}$-tree, and let $x, y \in V(T)$. If $\mathcal{A}(x) \cap \mathcal{A}(y) \neq \emptyset$, then x is a descendant of y or y is a descendant of x.*

Proof Let $A \in \mathcal{A}(x) \cap \mathcal{A}(y)$, so that there exist paths $x \rightsquigarrow v_A$ and $y \rightsquigarrow v_A$, and let r be the root of T. Then, both x and y appear in the path $r \rightsquigarrow v_A$. This entails that either x appears in the path $y \rightsquigarrow v_A$ or y appears in the path $x \rightsquigarrow v_A$, meaning that there is a path either from y to x or from x to y. □

Corollary 1 *Let T be an $\mathcal{A}$-tree, and let $x, y \in V(T)$. If $\mathcal{A}(x) \subsetneq \mathcal{A}(y)$, then there is a non-trivial path $y \rightsquigarrow x$.*

Proof By Lemma 1, if $\mathcal{A}(x) \subsetneq \mathcal{A}(y)$, then either x is a descendant of y or y is a descendant of x. But, being the inclusion strict, y cannot be a descendant of x. □

Corollary 2 *Let T be an $\mathcal{A}$-tree, and let $x, y \in V(T)$ be two different nodes. If $\mathcal{A}(x) = \mathcal{A}(y)$, then there is a path $x \rightsquigarrow y$ or a path $y \rightsquigarrow x$, such that its origin and all its intermediate nodes are unlabeled and elementary.*

Proof By Lemma 1, if $\mathcal{A}(x) = \mathcal{A}(y)$, there is either a path $x \rightsquigarrow y$ or a path $y \rightsquigarrow x$. If the origin or some intermediate node in this path is labeled or if any one of these nodes has more children than the one appearing in this path, then the set of labels will decrease from this node to its child in the path, and *a fortiori* from the origin to the end of the path. □

In particular, in a semi-labeled tree over $\mathcal{A}$, $\mathcal{A}(x) = \mathcal{A}(y)$ if and only if $x = y$, and $\mathcal{A}(x) \subsetneq \mathcal{A}(y)$ if and only if there exists a non-trivial path $y \rightsquigarrow x$. This entails that the cluster representation $\mathcal{C}_{\mathcal{A}}(T)$ of a semi-labeled tree T over $\mathcal{A}$ determines T up to isomorphism [10, Theorem 3.5.2].

Definition 3 *The restriction $T|\mathcal{X}$ of an $\mathcal{A}$-tree T to a set $\mathcal{X} \subseteq \mathcal{A}$ of labels is the subtree of T supported on the set of nodes*

$$V(T|\mathcal{X}) = \{v \in V(T) \mid \textit{there exists a path } v \rightsquigarrow v_A \textit{ for some } A \in \mathcal{X}\}$$
$$= \{v \in V(T) \mid \mathcal{A}(v) \cap \mathcal{X} \neq \emptyset\},$$

and where a node is labeled when it is labeled in T and this label belongs to $\mathcal{X}$, in which case its label in $T|\mathcal{X}$ is the same as in T.

If $\mathcal{X} \cap \mathcal{A}(T) = \emptyset$, then $T|\mathcal{X}$ is the empty $\mathcal{A}$-tree, while if $\mathcal{X} \cap \mathcal{A}(T) \neq \emptyset$, then $T|\mathcal{X}$ has the same root as T and leaves the nodes of T with labels in $\mathcal{X}$ that do not have any descendant with label in $\mathcal{X}$.

Now we introduce the notion of *locally compatible $\mathcal{A}$-trees* as the absence of *incompatible pairs and triples* of labels.

Definition 4 *Two $\mathcal{A}$-trees T_1 and T_2 are locally compatible when they satisfy the following two conditions:*

(C1) *For every two labels $A, B \in \mathcal{A}(T_1) \cap \mathcal{A}(T_2)$, there is a path $v_A \rightsquigarrow v_B$ in T_1 if and only if there is a path $v_A \rightsquigarrow v_B$ in T_2.*

(C2) *For every three labels $A, B, C \in \mathcal{A}(T_1) \cap \mathcal{A}(T_2)$, if there exists a non-trivial path $v_{B,C} \rightsquigarrow v_{A,B}$ in T_1, then there does not exist any non-trivial path $v_{A,B} \rightsquigarrow v_{B,C}$ in T_2.*

Any pair of labels A, B violating condition (C1) and any triple of labels A, B, C violating condition (C2) in a pair of trees T_1 and T_2 are said to be incompatible.

Two $\mathcal{A}$-trees T_1 and T_2 are locally incompatible when they are not locally compatible, that is, when they contain an incompatible pair or triple of labels.

So, if T_1 and T_2 represent phylogenetic trees with nested taxa, an incompatible pair of labels in T_1 and T_2 corresponds to a pair of taxa whose evolutionary precedence is different in both trees, while an incompatible triple of labels in T_1 and T_2 corresponds to three taxa whose evolutionary divergence is different in both trees.

Example 1 Let T_1, T_2 be two locally compatible $\mathcal{A}$-trees, and let $A, B, C \in \mathcal{A}(T_1) \cap \mathcal{A}(T_2)$. If T_1 contains a structure above v_A, v_B, v_C as the one shown in the left-hand side of Fig. 2,[1] then T_2 contains either the same structure above v_A, v_B, v_C as T_1 or the one shown in the right-hand side of the same figure.

Indeed, since no two among v_A, v_B, v_C are connected in T_1 by a path, condition (C1) implies that no two among the nodes in T_2 labeled A, B, C are connected by a path, either. Beside the structures shown in Fig. 2, only the structures T_2' and T_2'' shown in Fig. 3 satisfy this property. Now, T_1 contains a non-trivial path $v_{A,C} \rightsquigarrow v_{A,B}$, while T_2' contains a non-trivial path $v_{A,B} \rightsquigarrow v_{A,C}$; and T_1 contains a non-trivial path $v_{B,C} \rightsquigarrow v_{A,B}$, while T_2'' contains a non-trivial path $v_{A,B} \rightsquigarrow v_{B,C}$. So, in both cases we find incompatible triples of labels. On the other hand, in the $\mathcal{A}$-tree T_2 shown in Fig. 2, $v_{A,B} = v_{A,C} = v_{B,C}$, and therefore T_1 and this $\mathcal{A}$-tree clearly satisfy condition (C2) far as the labels A, B, C go.

[1] In this figure, as well as in Figs. 3, 4 and 5, edges may represent actually non-trivial paths.

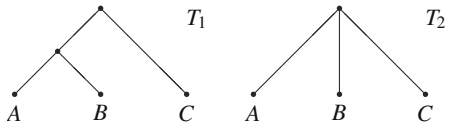

Fig. 2 T_1 and T_2 are locally compatible

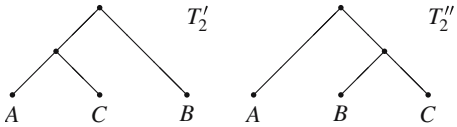

Fig. 3 T_2' and T_2'' are locally incompatible with T_1 in Fig. 2

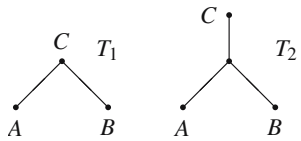

Fig. 4 T_1 and T_2 are locally compatible

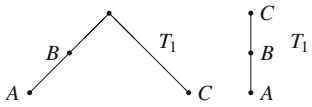

Fig. 5 These two $\mathcal{A}$-trees are only locally compatible with themselves

Example 2 Let T_1, T_2 be two locally compatible $\mathcal{A}$-trees, and let $A, B, C \in \mathcal{A}(T_1) \cap \mathcal{A}(T_2)$. If T_1 contains a structure above v_A, v_B, v_C as the one shown in the left-hand side of Fig. 4, then condition (C1) implies that T_2 contains either the same structure above v_A, v_B, v_C as T_1 or the one shown in the right-hand side of the same figure.

Example 3 Let T_1, T_2 be two locally compatible $\mathcal{A}$-trees, and let $A, B, C \in \mathcal{A}(T_1) \cap \mathcal{A}(T_2)$. If T_1 contains above v_A, v_B, v_C one of the structures shown in Fig. 5, then condition (C1) implies that T_2 must contain the same structure above v_A, v_B, v_C.

The following construction will be used henceforth several times.

Definition 5 *For every pair of $\mathcal{A}$-trees T_1 and T_2, let*

$$\bar{T}_1 = T_1|\mathcal{A}(T_1) \cap \mathcal{A}(T_2), \quad \text{and} \quad \bar{T}_2 = T_2|\mathcal{A}(T_1) \cap \mathcal{A}(T_2).$$

Notice that, by construction, every leaf of each $\bar{T}_i$ is labeled, and therefore $\bar{T}_1$ and $\bar{T}_2$ are $\mathcal{A}$-trees. Notice also that if $\mathcal{A}(T_1) = \mathcal{A}(T_2)$, then $\bar{T}_1 = T_1$ and $\bar{T}_2 = T_2$. In general,

$$\mathcal{A}(\bar{T}_1) = \mathcal{A}(\bar{T}_2) = \mathcal{A}(T_1) \cap \mathcal{A}(T_2).$$

Since local compatibility of two $\mathcal{A}$-trees refers to labels appearing in both $\mathcal{A}$-trees, we clearly have the following result.

Lemma 2 *Two $\mathcal{A}$-trees T_1 and T_2 are locally compatible if and only if $\bar{T}_1$ and $\bar{T}_2$ are so.* □

4 Weak topological embeddings

Compatibility of phylogenetic trees is usually stated in terms of the existence of simultaneous embeddings of some kind into a common supertree. In this section we introduce the embeddings that will correspond to local compatibility.

First, recall from [9] the definition of ancestral displaying, which we already present translated into our notations.

Definition 6 *An $\mathcal{A}$-tree T ancestrally displays an $\mathcal{A}$-tree S if the following properties hold:*

- $\mathcal{A}(S) \subseteq \mathcal{A}(T)$.
- *For every $A, B \in \mathcal{A}(S)$, there is a path $v_A \rightsquigarrow v_B$ in S if and only if there is a path $v_A \rightsquigarrow v_B$ in T.*
- *S is refined by $T|\mathcal{A}(S)$, that is, $\mathcal{C}_{\mathcal{A}}(S) \subseteq \mathcal{C}_{\mathcal{A}}(T|\mathcal{A}(S))$.*

We introduce now the following, more algebraic in flavour, definition of embedding that will turn out to be equivalent to ancestral displaying, up to the removal of elementary unlabeled nodes: cf. Proposition 1 below.

Definition 7 *A weak topological embedding of trees $f : S \to T$ is a mapping $f : V(S) \to V(T)$ satisfying the following conditions:*

- *It is injective.*
- *It preserves labels: for every $A \in \mathcal{A}(S)$, $f(v_A) = v_A$.*
- *It preserves and reflects paths: for every $a, b \in V(S)$, there is a path from a to b in S if and only if there is a path from $f(a)$ to $f(b)$ in T.*

When a weak topological embedding of $\mathcal{A}$-trees $f : S \to T$ exists, we say that S is a *weak $\mathcal{A}$-subtree* of T and that T is a *weak $\mathcal{A}$-supertree* of S.

Example 4 Let S, T_1, T_2 be the $\mathcal{A}$-trees described in Fig. 6. The mapping $f_1 : V(S) \to V(T_1)$ that sends the root r of S to the root r_1 of T_1, and every leaf of S to the leaf of T_1 with the same label, is injective, preserves labels, and preserves and reflects paths. Therefore, it defines a weak topological embedding

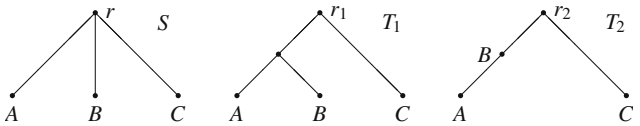

Fig. 6 The $\mathcal{A}$-trees in Example 4

$f_1 : S \to T_1$. On the contrary, the mapping $f_2 : V(S) \to V(T_2)$ that sends the root r of S to the root r_2 of T_2, and every leaf of S to the node of T_2 with the same label, is injective and preserves labels and paths, but it does not reflect paths: there is a path from v_B to v_A in T_2 that does not come from a path in S. Therefore, this mapping does not define a weak embedding.

Example 5 For every $\mathcal{A}$-tree T and for every $\mathcal{X} \subseteq \mathcal{A}(T)$, the inclusion of the restriction $T|\mathcal{X}$ into T is a weak topological embedding.

Remark 1 It is straightforward to prove that a mapping $f : V(S) \to V(T)$ preserves paths if and only if it *transforms arcs into paths*, that is, for every $a, b \in V(S)$, if $(a, b) \in E(S)$, then there exists a path $f(a) \rightsquigarrow f(b)$ in T. We shall sometimes use this alternative formulation without any further mention.

The following lemmas will be used several times in the sequel.

Lemma 3 *Let $f : S \to T$ be a weak topological embedding. Then, for every $v \in V(S)$, $\mathcal{A}(v) = \mathcal{A}(f(v)) \cap \mathcal{A}(S)$.*

Proof The inclusion $\mathcal{A}(v) \subseteq \mathcal{A}(f(v)) \cap \mathcal{A}(S)$ is a direct consequence of the fact that f preserves labels and paths, while the converse inclusion is a direct consequence of the fact that f preserves labels and reflects paths. □

Lemma 4 *Let $f : S \to T$ be a weak topological embedding of $\mathcal{A}$-trees. Then:*

(i) $\mathcal{L}(S) = \mathcal{L}(T|\mathcal{A}(S))$.
(ii) *f induces a weak topological embedding $f : S \to T|\mathcal{A}(S)$.*

Proof Notice first of all that $\mathcal{A}(S) \subseteq \mathcal{A}(T)$, because f preserves labels, and therefore it makes sense to define the restriction $T|\mathcal{A}(S)$; actually, the nodes of T with labels in $\mathcal{A}(S)$ are exactly the images of the labeled nodes of S. To simplify the notations, we shall denote in the rest of this proof $T|\mathcal{A}(S)$ by T'.

To prove (i), it is enough to check that the leaves of T' are exactly the images of leaves of S under f. And recall that $w \in V(T')$ is a leaf of T' if and only if $w = f(v_{S,A})$ for some $A \in \mathcal{A}(S)$ and $\mathcal{A}_T(w) \cap \mathcal{A}(S) = \{A\}$. Since, by the previous lemma, $\mathcal{A}(f(v_{S,A})) \cap \mathcal{A}(S) = \mathcal{A}(v_{S,A})$, we deduce that $w \in V(T')$ is a leaf of T' if and only if $w = f(v_{S,A})$ for some $A \in \mathcal{A}(S)$ such that $\mathcal{A}(v_{S,A}) = \{A\}$, that is, if and only if $w = f(v_{S,A})$ for some leaf $v_{S,A}$ of S, as we wanted to prove.

As far as (ii) goes, let us prove first that $f(V(S)) \subseteq V(T')$. Let $v \in V(S)$. If it is a leaf of S, then, as we have just seen, $f(v) \in V(T')$. If v is not a leaf of S, then there is a path in S from v to some leaf v'. Since f preserves paths, there is a path in T from $f(v)$ to $f(v')$, and $f(v')$ is labeled in $\mathcal{A}(S)$. Therefore, by the definition of restriction of an $\mathcal{A}$-tree, $f(v) \in V(T')$, too.

This proves that $f(V(S)) \subseteq V(T')$. And then it is straightforward to deduce that $f : S \to T'$ is injective, preserves labels, and that it preserves and reflects paths, from the corresponding properties for $f : S \to T$. □

Now we can prove that, as we announced, weak topological embeddings capture ancestral displaying.

Definition 8 *The largest semi-labeled weak subtree of an $\mathcal{A}$-tree S is the semi-labeled tree obtained from S by removing the elementary unlabeled nodes in it and replacing by arcs the maximal paths with all their intermediate nodes elementary and unlabeled.*

It is straightforward to check that the largest semi-labeled weak subtree of an $\mathcal{A}$-tree as defined above is, indeed, a semi-labeled tree that is a weak $\mathcal{A}$-subtree of S and it is the largest possible one.

Proposition 1 *Let S and T be two $\mathcal{A}$-trees, and let S' be the largest semi-labeled weak subtree of S. Then, T ancestrally displays S if and only if there exists a weak topological embedding $f : S' \to T$.*

Proof Assume that T ancestrally displays S, and in particular that $\mathcal{A}(S) \subseteq \mathcal{A}(T)$ and $\mathcal{C}_{\mathcal{A}}(S) \subseteq \mathcal{C}_{\mathcal{A}}(T|\mathcal{A}(S))$. Since elementary unlabeled nodes do not contribute any new member to the cluster representation, $\mathcal{C}_{\mathcal{A}}(S) = \mathcal{C}_{\mathcal{A}}(S')$ and therefore, $\mathcal{C}_{\mathcal{A}}(S') \subseteq \mathcal{C}_{\mathcal{A}}(T|\mathcal{A}(S))$. Then, it turns out that the mapping

$$\begin{aligned} f : V(S') &\to V(T|\mathcal{A}(S)) \\ v &\mapsto v_{T|\mathcal{A}(S),\mathcal{A}(v)} \end{aligned}$$

defines a weak topological embedding $f : S' \to T|\mathcal{A}(S)$, which composed with the weak embedding $T|\mathcal{A}(S) \hookrightarrow T$ yields a weak embedding $S' \to T$. We delay the proof of this fact, as well as that of the converse implication, until an Appendix at the end of the paper where we gather several proofs that we have omitted from the main body of the paper to make easier its reading. □

Now, recall from [9] the notion of ancestral compatibility.

Definition 9 *Two $\mathcal{A}$-trees T_1, T_2 are ancestrally compatible when there exists an $\mathcal{A}$-tree that ancestrally displays both of them. If two $\mathcal{A}$-trees are not ancestrally compatible, we say that they are ancestrally incompatible.*

The definition of weak topological embeddings has been chosen in order for ancestral compatibility to be exactly the same as 'compatibility for weak topological embeddings.'

Proposition 2 *Two $\mathcal{A}$-trees T_1, T_2 are ancestrally compatible if and only if they have a common weak $\mathcal{A}$-supertree, that is, if and only if they admit a weak topological embedding into a same $\mathcal{A}$-tree.*

Proof For every $i = 1, 2$, let T_i' be the largest semi-labeled weak subtree of T_i. If there exist weak topological embeddings $f_1 : T_1 \to T$ and $f_2 : T_2 \to T$ of T_1 and T_2 into a same $\mathcal{A}$-tree T, then, since each T_i' is a weak $\mathcal{A}$-subtree of the corresponding T_i, each one of these weak topological embeddings induces a weak topological embedding $f_i' : T_i' \to T$, showing that T ancestrally displays T_1 and T_2.

Conversely, assume that there exist weak topological embeddings $g_1 : T_1' \to T$ and $g_2 : T_2' \to T$ of T_1' and T_2' into a same $\mathcal{A}$-tree T. Let $\tilde{T}$ be the $\mathcal{A}$-tree obtained from T in the following way. For every arc $(v, w) \in E(T)$, if there exists an arc (v_i, w_i) in one T_i' such that $g_i(v_i) = v$ and $g_i(w_i) = w$, we split the arc (v, w) in T into a path $v \rightsquigarrow w$, with all its intermediate nodes elementary and unlabeled, of length equal to the length of the path $v_i \rightsquigarrow w_i$; if there are arcs $(v_1, w_1) \in E(T_1')$ and $(v_2, w_2) \in E(T_2')$ such that $g_1(v_1) = g_2(v_2) = v$ and $g_1(w_1) = g_2(w_2) = w$, then we split the arc (v, w) in T into a path $v \rightsquigarrow w$ as before, but now of length the maximum of the lengths of the paths $v_1 \rightsquigarrow w_1$ and $v_2 \rightsquigarrow w_2$. It is clear then that each $g_i : T_i' \to T$ can be extended to a weak topological embedding $\tilde{g}_i : T_i \to \tilde{T}$. □

From now on, we shall use this characterization of ancestral compatibility as the working definition of it.

The main result of this paper will establish that ancestral compatibility is equivalent to local compatibility. To prove it, we shall use the following proposition, which establishes that the ancestral compatibility of two $\mathcal{A}$-trees can be checked at the level of $\bar{T}_1$ and $\bar{T}_2$, as it was also the case for local compatibility.

Proposition 3 *Let T_1 and T_2 be $\mathcal{A}$-trees and let $\bar{T}_1$ and $\bar{T}_2$ be their $\mathcal{A}$-subtrees described in* Definition 5. *Then, T_1 and T_2 are ancestrally compatible if and only if $\bar{T}_1$ and $\bar{T}_2$ are ancestrally compatible.*

Proof As we have seen in the proof of Proposition 2, if T_1 and T_2 are ancestrally compatible, then $\bar{T}_1$ and $\bar{T}_2$ are also so. Conversely, let $f_1 : \bar{T}_1 \to \bar{T}$ and $f_2 : \bar{T}_2 \to \bar{T}$ be two weak topological embeddings. Then, a simple construction applied to $\bar{T}$, consisting on first restricting it to $\mathcal{A}(\bar{T}_1) = \mathcal{A}(\bar{T}_2)$, next "splitting in two" every node that is the image under f_1 and f_2 of two nodes with different labels in T_1 and T_2, and finally "adding $(T_1 - \bar{T}_1) \sqcup (T_2 - \bar{T}_2)$" to the resulting $\mathcal{A}$-tree, yields a common weak $\mathcal{A}$-supertree of T_1 and T_2. The details of this construction, and an example, are given again in the Appendix. □

5 Main results

In this section we establish that local compatibility is the same as ancestral compatibility. We also provide a characterization of the ancestral, or local, compatibility of a family of $\mathcal{A}$-trees in terms of joint properties of their cluster representations.

Definition 10 *Let T_1 and T_2 be two $\mathcal{A}$-trees.*

(a) Assume that $\mathcal{A}(T_1) = \mathcal{A}(T_2)$. In this case, the join of T_1 and T_2 is the $\mathcal{A}$-labeled graph $T_{1,2}$ defined as follows:
For every $\ell = 1, 2$ and for every $Y \in \mathcal{C}_{\mathcal{A}}(T_\ell)$, let

$$m_{\ell,Y} = \#\{v \in V(T_\ell) \mid \mathcal{A}_{T_\ell}(v) = Y\}$$

and $n_Y = \max\{m_{1,Y}, m_{2,Y}\}$. Set $\mathcal{C} = \mathcal{C}_{\mathcal{A}}(T_1) \cup \mathcal{C}_{\mathcal{A}}(T_2)$. Then:

- *Its nodes are*

$$w_{Y,j} \quad \text{with } Y \in \mathcal{C} \text{ and } j = 1, \ldots, n_Y.$$

- *Its arcs are:*

$$\begin{array}{ll} (w_{Y,j}, w_{Y,j-1}) & j = 2, \ldots, n_Y \\ (w_{Y,1}, w_{Z,n_Z}) & \text{if } Z \subsetneq Y \text{ and there is no } Z' \in \mathcal{C} \text{ such that } Z \subsetneq Z' \subsetneq Y. \end{array}$$

- *If there exists some $Y \in \mathcal{C}$ such that*

$$Y = \left(\bigcup \{Z \in \mathcal{C} \mid Z \subsetneq Y\}\right) \sqcup \{A\}$$

for some label $A \in \mathcal{A}$, then the node $w_{Y,1}$ is labeled with this A. In particular, the nodes $w_{A,1}$, with $\{A\}$ any singleton in $\mathcal{C}$, are labeled with the corresponding label A.

Now, for every $\ell = 1, 2$, we define a mapping $f_\ell : V(T_\ell) \to V(T_{1,2})$ as follows. For every $Y \in \mathcal{C}_{\mathcal{A}}(T_\ell)$, let $\{x^{(\ell)}_{Y,1}, \ldots, x^{(\ell)}_{Y,m_{\ell,Y}}\} \in V(T_\ell)$ be the set of nodes of T_ℓ with cluster Y, ordered as follows: $x^{(\ell)}_{Y,1} = v_{T_\ell,Y}$, and $(x^{(\ell)}_{Y,i+1}, x^{(\ell)}_{Y,i}) \in E(T_\ell)$ for every $i = 1, \ldots, m_{\ell,Y} - 1$.
With these notations, $f_\ell : V(T_\ell) \to V(T)$ is defined by

$$f_\ell(x^{(\ell)}_{Y,i}) = w_{Y,i} \text{ forevery } Y \in \mathcal{C}_{\mathcal{A}}(T_\ell) \text{ and } i = 1, \ldots, m_{\ell,Y}.$$

Since $\mathcal{C}_{\mathcal{A}}(T_\ell) \subseteq \mathcal{C}$ and, for every $Y \in \mathcal{C}_{\mathcal{A}}(T_\ell)$, $m_{\ell,Y} \leqslant n_Y$, it is clear that f_ℓ is well defined and injective.

(b) If $\mathcal{A}(T_1) \neq \mathcal{A}(T_2)$, let $\bar{T}_1$ and $\bar{T}_2$ be the $\mathcal{A}$-subtrees of T_1 and T_2 described in Definition 5. Then, the join $T_{1,2}$ of T_1 and T_2 is the result of applying the construction in the proof of Proposition 3 to the join $\bar{T}_{1,2}$ of $\bar{T}_1$ and $\bar{T}_2$ and the mappings $f_\ell : V(T_\ell) \to V(T_{1,2})$, $\ell = 1, 2$, are obtained by extending the mappings $f_\ell : V(\bar{T}_\ell) \to V(\bar{T}_{1,2})$ also in the way described in that proof.

Notice that, by construction, the mappings $f_l : V(T_l) \to V(T_{1,2})$, $l = 1, 2$, are always *jointly surjective*, that is, every node of $T_{1,2}$ belongs to the image of one or the other.

Theorem 1 *Let T_1 and T_2 be two $\mathcal{A}$-trees with $\mathcal{A}(T_1) = \mathcal{A}(T_2)$. Then, the following assertions are equivalent:*

(i) *T_1 and T_2 are ancestrally compatible.*
(ii) *T_1 and T_2 are locally compatible.*
(iii) *$\mathcal{C}_{\mathcal{A}}(T_1)$ and $\mathcal{C}_{\mathcal{A}}(T_2)$ satisfy jointly the following two conditions:*
  - *For every $A \in \mathcal{A}(T_1) = \mathcal{A}(T_2)$, the smallest member of $\mathcal{C}_{\mathcal{A}}(T_1)$ containing A is equal to the smallest member of $\mathcal{C}_{\mathcal{A}}(T_2)$ containing this label.*
  - *For every $X \in \mathcal{C}_{\mathcal{A}}(T_1)$ and $Y \in \mathcal{C}_{\mathcal{A}}(T_2)$,if $X \cap Y \neq \emptyset$, then $X \subseteq Y$ or $Y \subseteq X$.*
(iv) *The join $T_{1,2}$ of T_1 and T_2 is an $\mathcal{A}$-tree and the mappings $f_1 : V(T_1) \to V(T_{1,2})$ and $f_2 : V(T_2) \to V(T_{1,2})$ are weak topological embeddings.*

We give the proof of this theorem in the Appendix.

Corollary 3 *Let T_1 and T_2 be $\mathcal{A}$-trees. Then, the following assertions are equivalent:*

(i) *T_1 and T_2 are ancestrally compatible.*
(ii) *T_1 and T_2 are locally compatible.*
(iii) *Their $\mathcal{A}$-subtrees $\bar{T}_1$ and $\bar{T}_2$ described in* Definition 5 *satisfy condition* (iii) *in Theorem 1.*
(iv) *The join $T_{1,2}$ of T_1 and T_2 is an $\mathcal{A}$-tree and the mappings $f_1 : V(T_1) \to V(T_{1,2})$ and $f_2 : V(T_2) \to V(T_{1,2})$ are weak topological embeddings.*

Proof By Lemma 2, T_1 and T_2 are locally compatible if and only if $\bar{T}_1$ and $\bar{T}_2$ are so, and by Proposition 3, T_1 and T_2 are ancestrally compatible if and only if $\bar{T}_1$ and $\bar{T}_2$ are so. These facts, together with the last theorem, prove the implications (i)⇒(ii) and (ii)⇒(iii). As far as (iii)⇒(iv) goes, it is a direct consequence of the corresponding implication in the last theorem together with the proof of Proposition 3. □

Corollary 4 *Let T_1 and T_2 be semi-labeled trees over $\mathcal{A}$. Then, the following assertions are equivalent:*

(i) *T_1 and T_2 admit simultaneous weak topological embeddings into a same semi-labeled tree over $\mathcal{A}$.*
(ii) *T_1 and T_2 are ancestrally compatible.*
(iii) *T_1 and T_2 are locally compatible.*
(iv) *Their $\mathcal{A}$-subtrees $\bar{T}_1$ and $\bar{T}_2$ described in* Definition 5 *satisfy condition (iii) in Theorem 1.*
(v) *The join $T_{1,2}$ of T_1 and T_2 is a semi-labeled tree and the mappings $f_1 : V(T_1) \to V(T_{1,2})$ and $f_2 : V(T_2) \to V(T_{1,2})$ are weak topological embeddings.*

Proof It only remains to prove (iv)$\Longrightarrow$(v). And to do that, it is enough to notice that if T_1 and T_2 are semi-labeled trees over $\mathcal{A}$ such that $\bar{T}_1$ and $\bar{T}_2$ satisfy condition (iii) in Theorem 1, then their join $T_{1,2}$ is not only an $\mathcal{A}$-tree, but a semi-labeled tree, because, since $f_1 : T_1 \to T_{1,2}$ and $f_2 : T_2 \to T_{1,2}$ are jointly surjective, no elementary node in it remains unlabeled. □

6 Algorithmic details

The equivalence between ancestral compatibility and the properties of the cluster representations of the trees established in Theorem 1, leads to a very simple polynomial-time algorithm for testing ancestral compatibility of two semi-labeled trees. The detailed pseudo-code of the algorithm is shown in Fig. 7.

We have implemented in Perl this compatibility test, and the implementation is freely available for download from the BioPerl collection of Perl modules for computational biology [12]. Given two semi-labeled trees T_1 and T_2 with common labels $\mathcal{A} = \mathcal{A}(T_1) \cap \mathcal{A}(T_2)$, if the trees are incompatible, the actual implementation collects and returns all labels $A \in \mathcal{A}$ such that the smallest member of $\mathcal{C}_{\mathcal{A}}(T_1|\mathcal{A})$ containing A does not coincide with the smallest member of $\mathcal{C}_{\mathcal{A}}(T_2|\mathcal{A})$ containing A, as well as all pairs of clusters $X_1 \in \mathcal{C}_{\mathcal{A}}(T_1|\mathcal{A})$ and $X_2 \in \mathcal{C}_{\mathcal{A}}(T_2|\mathcal{A})$ such that $X_1 \cap X_2 \neq \emptyset$, $X_1 \not\subseteq X_2$, and $X_2 \not\subseteq X_1$. This additional information constitutes a *certificate of incompatibility*, which can be useful for checking the underlying phylogenetic studies that have lead to incompatible clusters.

The following Perl code illustrates the use of the `Bio::Tree::Compatible` module for testing compatibility of two semi-labeled trees and listing all pairs of incompatible clusters in the trees.

compatible(T_1, T_2)
$\mathscr{A} := \mathscr{A}(T_1) \cap \mathscr{A}(T_2)$
$\bar{T}_1 := T_1|\mathscr{A}$
$\bar{T}_2 := T_2|\mathscr{A}$
foreach label $A \in \mathscr{A}$ **do**
  let X_1 be the smallest member of $\mathscr{C}_{\mathscr{A}}(\bar{T}_1)$ containing A
  let X_2 be the smallest member of $\mathscr{C}_{\mathscr{A}}(\bar{T}_2)$ containing A
  if $X_1 \neq X_2$ **then**
    return X_1 and X_2 are incompatible
foreach cluster $X_1 \in \mathscr{C}_{\mathscr{A}}(\bar{T}_1)$ **do**
  foreach cluster $X_2 \in \mathscr{C}_{\mathscr{A}}(\bar{T}_2)$ **do**
    if $X_1 \cap X_2 \neq \emptyset$ and $X_1 \not\subseteq X_2$ and $X_2 \not\subseteq X_1$ **then**
      return X_1 and X_2 are incompatible
return T_1 and T_2 are compatible

Fig. 7 Algorithm for testing ancestral compatibility of two semi-labeled trees T_1 and T_2

```
use Bio::Tree::Compatible;
use Bio::TreeIO;

my $filename = $ARGV[0];
my $input = new Bio::TreeIO('-format' => 'newick',
                            '-file' => $filename);
my $t1 = $input->next_tree;
my $t2 = $input->next_tree;

my ($incompat, $ilabels, $inodes) =
  $t1->Bio::Tree::Compatible::is_compatible($t2);

if ($incompat) {
  print "the␣trees␣are␣incompatible\n";

  my %cluster1 = %{
    $t1->Bio::Tree::Compatible::cluster_representation };
  my %cluster2 = %{
    $t2->Bio::Tree::Compatible::cluster_representation };

  if (scalar(@$ilabels)) {
    foreach my $label (@$ilabels) {
      my $node1 = $t1->find_node(-id => $label);
      my $node2 = $t2->find_node(-id => $label);
      my @c1 = sort @{ $cluster1{$node1} };
      my @c2 = sort @{ $cluster2{$node2} };
      print "label␣$label";
      print "␣cluster"; map { print "␣",$_ } @c1;
      print "␣cluster"; map { print "␣",$_ } @c2;
      print "\n";
    }
  }

  if (scalar(@$inodes)) {
    while (@$inodes) {
      my $node1 = shift @$inodes;
      my $node2 = shift @$inodes;
      my @c1 = sort @{ $cluster1{$node1} };
      my @c2 = sort @{ $cluster2{$node2} };
      print "cluster"; map { print "␣",$_ } @c1;
      print "␣properly␣intersects␣cluster";
      map { print "␣",$_ } @c2; print "\n";
    }
  }
} else {
  print "the␣trees␣are␣compatible\n";
}
```

An application of `Bio::Tree::Compatible` is shown in Fig. 8. The input consists of three pairs of semi-labeled trees, describing the evolution of net-veined *Lilliaflorae* (obtained from study S2x4x96c17c14c22 in the TreeBASE [4] phylogenetic database), *Asteraceae* (obtained from studies S2x4x96c16c23c42 and S2x7x96c15c50c04 in TreeBASE), and *Dipsacales* (studies S2x7x96c15c 21c45 and S866 in TreeBASE).

Using the `Bio::Tree::Compatible` module, we have performed a systematic study of tree compatibility on TreeBASE, which currently contains 2,592

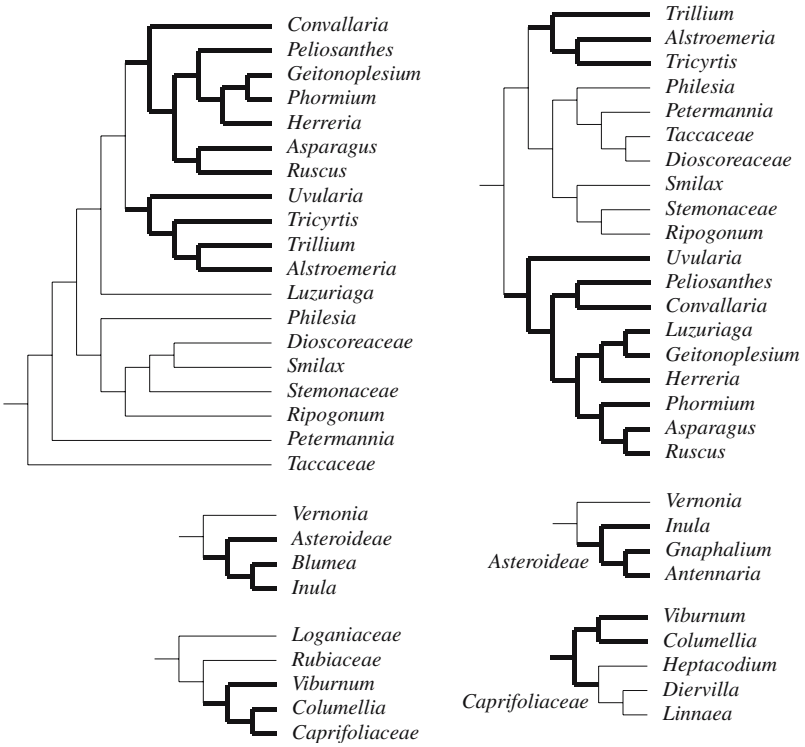

Fig. 8 Two incompatible phylogenetic trees (*top*) obtained from study S2x4x96c17c14c22 in TreeBASE, with incompatible clusters shown with *thick lines*. Two incompatible semi-labeled trees (*middle*) obtained from studies S2x4x96c16c23c42 and S2x7x96c15c50c04 in TreeBASE, one of which has a cluster labeled by a taxon in the other tree. Two incompatible semi-labeled trees (it bottom) obtained from studies S2x7x96c15c21c45 and S866 in TreeBASE, in which an incompatible triple of labels involves three taxa in one tree and two taxa plus one internal label in the other tree

phylogenies with over 36,000 taxa among them. In this study, we have found 2,527 pairs of incompatible semi-labeled trees (like those shown in Fig. 8) from a total of 7,835 pairs of trees that could show incompatibility (by sharing at least three taxa if no internal nodes are labeled, or sharing at least two taxa, one of them internal in one of the trees).

7 Conclusions

Phylogenetic tree compatibility is the most important concept underlying widely-used methods for assessing the agreement of different phylogenetic trees with overlapping taxa and combining them into common supertrees to reveal the tree of life. The study of the compatibility of phylogenetic trees with nested taxa, also known as semi-labeled trees, was asked for in [6], and the notion of ancestral compatibility was introduced in [3,9].

We have analyzed in detail the meaning of the ancestral compatibility of semi-labeled trees from the points of view of the local structure of the trees, of the existence of embeddings into a common supertree, and of the joint properties of their cluster representations. We have established the equivalence between ancestral compatibility and the absence of certain incompatible pairs and triples of labels in the trees under comparison, and have also proved the equivalence between ancestral compatibility and a certain property of the cluster representations of the trees.

Our analysis has lead to a very simple polynomial-time algorithm for testing ancestral compatibility, which we have implemented and is freely available for download from the BioPerl collection of Perl modules for computational biology. Future work includes extending the `Bio::Tree::Compatible` implementation into a `Bio::Tree::Supertree` module for building a common supertree of two compatible semi-labeled trees.

Acknowledgments All four authors have been partially supported by the INTAS project IT 04-77-7178. Moreover, M. Llabrés and F. Rosselló have been partially supported by the Spanish DGES project BFM2003-00771, and G. Valiente has been partially supported by the Spanish CICYT project TIN 2004-07925-C03-01 and the Japan Society for the Promotion of Science through Long-term Invitation Fellowship L05511 for visiting JAIST (Japan Advanced Institute of Science and Technology). G. Valiente acknowledges with thanks R. D. M. Page for many discussions on compatibility of phylogenetic trees. The authors would also like to acknowledge with thanks the editor and anonymous referees, whose comments and criticism have led to a substantial improvement of this paper.

Appendix

Proof of Proposition 1 To simplify the notations, let us denote $T|\mathcal{A}(S)$ by T''. To begin with, we want to prove that the mapping

$$\begin{aligned} f : V(S') &\to V(T'') \\ v &\mapsto v_{T'',\mathcal{A}(v)} \end{aligned}$$

defines a weak topological embedding $f : S' \to T''$.

- *It is injective*. Let v, w be two different nodes of S'. Since every node in S' is the most recent common ancestor of its labeled descendants, that is, $x = v_{S',\mathcal{A}(x)}$ for every $x \in V(S')$, we have that $\mathcal{A}(v) \neq \mathcal{A}(w)$. And then, since $\mathcal{C}_{\mathcal{A}}(S') \subseteq \mathcal{C}_{\mathcal{A}}(T'')$, it turns out that $\mathcal{A}(v), \mathcal{A}(w)$ are two different members of $\mathcal{C}_{\mathcal{A}}(T'')$, and hence $\mathcal{A}(v_{T'',\mathcal{A}(v)}) = \mathcal{A}(v) \neq \mathcal{A}(w) = \mathcal{A}(v_{T'',\mathcal{A}(w)})$, which clearly implies that $v_{T'',\mathcal{A}(v)} \neq v_{T'',\mathcal{A}(w)}$.
- *It preserves labels*. Let $A \in \mathcal{A}(S')$ and $v = v_{S',A}$. Then, $f(v) = v_{T'',\mathcal{A}(v_{S',A})}$ is labeled A because, by the second property of ancestral displaying, the labels of the nodes in S' that are descendants of v are exactly the labels of the nodes in T'' that are descendants of $v_{T'',A}$, and therefore $v_{T'',A}$ is the least common ancestor of the nodes with labels in $\mathcal{A}(v_{S',A})$, that is, $v_{T'',A} = v_{T'',\mathcal{A}(v_{S',A})} = f(v)$, as we claimed.

– *It preserves and reflects paths*. Since $\mathcal{A}(v) = \mathcal{A}(f(v))$ for every $v \in V(S')$, we have the following sequence of equivalences: for every $v, w \in V(S')$,

$$\begin{aligned} &\text{there exists a non-trivial path } v \rightsquigarrow w \\ &\qquad \Longleftrightarrow \mathcal{A}(w) \subsetneq \mathcal{A}(v) \\ &\qquad \Longleftrightarrow \mathcal{A}(f(w)) \subsetneq \mathcal{A}(f(v)) \\ &\qquad \Longleftrightarrow \text{there exists a non-trivial path } f(v) \rightsquigarrow f(w). \end{aligned}$$

The implications $\Leftarrow$ in the first equivalence and $\Rightarrow$ in the last equivalence are given by Corollary 1, while the converse implication in both cases is entailed by the fact that $v, w, f(v), f(w)$ are most recent common ancestors of sets of labeled nodes, and then non-trivial paths between them imply strict inclusions of sets of labels of descendants.

So, we have a weak topological embedding $f : S' \to T''$, and since T'' is a weak $\mathcal{A}$-subtree of T, it induces a weak topological embedding $f : S' \to T$, as we wanted to prove.

Conversely, assume that there exists a weak topological embedding $f : S' \to T$. Then:

– $\mathcal{A}(S) = \mathcal{A}(S') \subseteq \mathcal{A}(T)$ because f preserves labels.
– For every $A, B \in \mathcal{A}(S)$, by construction, $v_{S,A} = v_{S',A}$ and $v_{S,B} = v_{S',B}$, and there exists a path $v_{S,A} \rightsquigarrow v_{S,B}$ in S if and only if there exists a path $v_{S',A} \rightsquigarrow v_{S',B}$ in S'. Moreover, since f preserves labels and preserves and reflects paths, there exists a path $v_{S',A} \rightsquigarrow v_{S',B}$ in S' if and only if there exists a path $v_{T,A} = f(v_{S',A}) \rightsquigarrow f(v_{S',B}) = v_{T,B}$ in T. Combining these equivalences, we obtain that, for every $A, B \in \mathcal{A}(S)$, there exists a path $v_{S,A} \rightsquigarrow v_{S,B}$ in S if and only if there exists a path $v_{T,A} \rightsquigarrow v_{T,B}$ in T.
– Let $X \in \mathcal{C}_{\mathcal{A}}(S)$ and let $v = v_{S,X} = v_{S',X}$. By Lemma 4, $f : S' \to T$ induces a weak topological embedding $f : S' \to T|\mathcal{A}(S') = T|\mathcal{A}(S)$ and then, by Lemma 3, $\mathcal{A}_{T|\mathcal{A}(S)}(f(v)) = \mathcal{A}_{S'}(v) = \mathcal{A}_S(v) = X$. Therefore, $X \in \mathcal{C}(T|\mathcal{A}(S))$, and, being X arbitrary, we conclude that $\mathcal{C}_{\mathcal{A}}(S) \subseteq \mathcal{C}(T|\mathcal{A}(S))$.

This proves that T ancestrally displays S. □

Proof of Proposition 3 It remains to prove the "if" implication. So, let $f_1 : \bar{T}_1 \to \bar{T}$ and $f_2 : \bar{T}_2 \to \bar{T}$ be two weak topological embeddings. By Lemma 4.(ii), f_1 and f_2 induce weak topological embeddings into the restriction of $\bar{T}$ to $\mathcal{A}(\bar{T}_1) = \mathcal{A}(\bar{T}_2)$. We replace then $\bar{T}$ by this restriction, which we shall still denote by $\bar{T}$, and we have that $\mathcal{A}(T) = \mathcal{A}(\bar{T}_1) = \mathcal{A}(\bar{T}_2)$ and, by Lemma 4.(i), that $\mathcal{L}(T) = \mathcal{L}(\bar{T}_1) = \mathcal{L}(\bar{T}_2)$.

Next, for every pair of *different* labels A_1, A_2 such that $v_{T_1,A_1} \in V(\bar{T}_1)$ and $v_{T_2,A_2} \in V(\bar{T}_2)$ and $f_1(v_{T_1,A_1}) = f_2(v_{T_2,A_2})$, we "blow out" this node in $\bar{T}$ into an arc in the following way. To begin with, notice that if A_1, A_2 satisfy this property, then $A_1, A_2 \notin \mathcal{A}(T_1) \cap \mathcal{A}(T_2)$: if, say, $A_2 \in \mathcal{A}(T_1) \cap \mathcal{A}(T_2)$ then, $A_2 \in \mathcal{A}(\bar{T}_1) = \mathcal{A}(\bar{T}_2)$ and hence $v_{T_1,A_2} = v_{\bar{T}_1,A_2}$ and $v_{T_2,A_2} = v_{\bar{T}_2,A_2}$. Then, since $f_1 : \bar{T}_1 \to \bar{T}$ and $f_2 : \bar{T}_2 \to \bar{T}$ preserve labels,

$$f_1(v_{T_1,A_1}) = f_2(v_{T_2,A_2}) = f_2(v_{\bar{T}_2,A_2}) = v_{\bar{T},A_2} = f_1(v_{\bar{T}_1,A_2}) = f_1(v_{T_1,A_2}),$$

and then, f_1 being injective, $v_{T_1,A_2} = v_{T_1,A_1}$, that is, $A_2 = A_1$. Therefore, v_{T_1,A_1} and v_{T_2,A_2} do not keep their labels in $\bar{T}_1$ and $\bar{T}_2$. Now, given the node $w = f_1(v_{T_1,A_1}) = f_2(v_{T_2,A_2})$ (which, by what we have just discussed, will be unlabeled, either), we add a new node w', we split the arc going from w's parent w_0 to w into two arcs (w_0, w'), (w', w) – if w was the root of T, we simply add a new arc (w', w) – and we redefine f_1 by sending v_{T_1,A_1} to w' while we do not change f_2 (alternatively, we could have redefined f_2, by sending v_{T_2,A_2} to w', and left f_1 unchanged). It is straightforward to check that the new mapping f_1 obtained in this way and the 'old' f_2 are still weak topological embeddings from T_1 and T_2 to the new $\mathcal{A}$-tree.

After repeating this process as many times as necessary, and still calling $\bar{T}$ the target $\mathcal{A}$-tree obtained at the end, we obtain weak topological embeddings $f_1 : \bar{T}_1 \to \bar{T}$ and $f_2 : \bar{T}_2 \to \bar{T}$ such that, for every $A_1 \in \mathcal{A}(T_1)$ and $A_2 \in \mathcal{A}(T_2)$, if $A_1 \neq A_2$, then $f_1(v_{T_1,A_1}) \neq f_2(v_{T_2,A_2})$.

Now, we expand this common weak $\mathcal{A}$-supertree $\bar{T}$ of $\bar{T}_1$ and $\bar{T}_2$ to a common weak $\mathcal{A}$-supertree T of T_1 and T_2, by "adding $(T_1 - \bar{T}_1) \sqcup (T_2 - \bar{T}_2)$" to it. More specifically, to obtain T:

- we add to $\bar{T}$ all nodes in $(V(T_1) - V(\bar{T}_1)) \sqcup (V(T_2) - V(\bar{T}_2))$, with their corresponding labels in T_1 or T_2;
- for every $i = 1, 2$, we add all arcs in T_i between nodes in $V(T_i) - V(\bar{T}_i)$;
- for every $i = 1, 2$, and for every arc $(a, b) \in E(T_i)$ with $a \in V(\bar{T}_i)$ and $b \in V(T_i) - V(\bar{T}_i)$, we add an arc between $f_i(a)$ and b;
- for every $i = 1, 2$, and for every $A \in \mathcal{A}(T_i) - \mathcal{A}(\bar{T}_i)$ such that $v_{T_i,A} \in V(\bar{T}_i)$, we label A the node $f_i(v_{T_i,A})$.

The fact that the labels of the nodes in any $V(T_i) - V(\bar{T}_i)$ cannot belong to the set of labels of the other tree and that nodes with different labels in T_1 and T_2 don't have the same image in $\bar{T}$ under f_1 and f_2, entails that the labeling of the new tree T obtained in this way is injective, and hence it is an $\mathcal{A}$-tree. An example of the construction of such a tree T is given in Example 6 below.

It turns out that T is a weak $\mathcal{A}$-supertree of T_1 and T_2: we prove it only for T_1. Consider the mapping $f'_1 : V(T_1) \to V(T)$ that is defined on $V(\bar{T}_1)$ as the original embedding $f_1 : V(\bar{T}_1) \to V(\bar{T})$ and on $V(T_1) - V(\bar{T}_1)$ as the identity. It is clearly injective and preserves labels. Moreover, it preserves paths, because f_1 sends arcs in $\bar{T}_1$ to paths in $\bar{T}$, and arcs outside $\bar{T}_1$ become arcs in T, and it reflects paths, because it reflects paths in $\bar{T}$ and the arcs between nodes in $f'_1(V(T_1))$ that have been added come from arcs in T_1.

Therefore, T_1 and T_2 are ancestrally compatible, as we wanted to prove. □

Example 6 Consider the semi-labeled trees T_1 and T_2 described in Fig. 9: the labels are denoted by capital letters, and numbers have been assigned to non-labeled nodes in order to simplify the description of the construction. The corresponding $\mathcal{A}$-trees $\bar{T}_1$ and $\bar{T}_2$, which are no longer semi-labeled trees, are described in Fig. 10: the nodes c, h and i are the nodes that were formerly labeled C, H and I, and that are no longer labeled in these trees.

The $\mathcal{A}$-trees $\bar{T}_1$ and $\bar{T}_2$ are ancestrally compatible. A weak common $\mathcal{A}$-supertree of them is given by the $\mathcal{A}$-tree $\bar{T}$ described in Fig. 11, together with the weak topological embeddings $f_1 : \bar{T}_1 \to \bar{T}$ and $f_2 : \bar{T}_2 \to \bar{T}$ that are indicated by assigning in the picture to each non-labeled node in T its preimages under f_1 and f_2.

Notice that $f_1(v_{T_1,C}) = f_2(v_{T_2,H})$. To avoid it, we blow up this node into an arc and we separate these two images: the corresponding new weak $\mathcal{A}$-supertree $\bar{T}$ is described in the left-hand side of Fig. 12.

Finally, the weak common $\mathcal{A}$-supertree T of T_1 and T_2 obtained by "adding $(T_1 - \bar{T}_1) \sqcup (T_2 - \bar{T}_2)$ to $\bar{T}$" is described in the right-hand side of Fig. 12 (the embeddings are indicated as before).

Proof of Theorem 1 (i)$\Longrightarrow$(ii) Assume that T_1 and T_2 are ancestrally compatible, and let $f_1 : T_1 \to T$ and $f_2 : T_2 \to T$ be two weak topological embeddings. To prove that they are locally compatible, we shall show that they satisfy conditions (C1) and (C2).

(C1) Assume that T_1 contains a path $v_A \rightsquigarrow v_B$. Since f_1 preserves this path, there exists a path $v_A \rightsquigarrow v_B$ in T, and then this path must be reflected by f_2, yielding a path $v_A \rightsquigarrow v_B$ in T_2.

(C2) Let $A, B, C \in \mathcal{A}(T_1) = \mathcal{A}(T_2)$. Let

$$y = v_{T_1,A,B} \quad \text{and} \quad z = v_{T_1,B,C},$$

and assume that there is a non-trivial path $z \rightsquigarrow y$; see Fig. 13. In particular, y cannot be an ancestor of v_C: otherwise, it would be a common ancestor of v_B and v_C, which would entail a path from y to z that cannot exist.

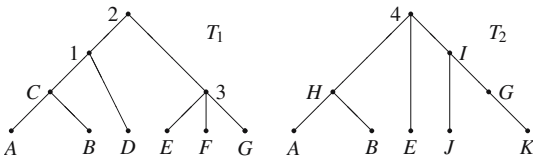

Fig. 9 The semi-labeled trees T_1, T_2 in Example 6

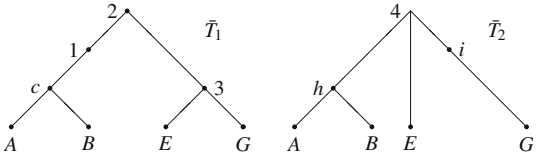

Fig. 10 The $\mathcal{A}$-trees $\bar{T}_1, \bar{T}_2$ corresponding to the semi-labeled trees T_1, T_2 in Fig. 9

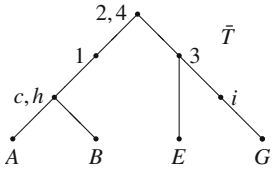

Fig. 11 A weak common $\mathcal{A}$-supertree of $\bar{T}_1$ and $\bar{T}_2$

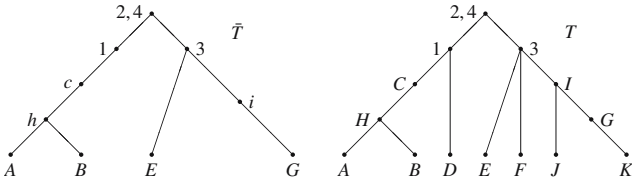

Fig. 12 The new $\mathcal{A}$-tree $\bar{T}$ obtained after blowing out the node c, h in the $\mathcal{A}$-tree T in Fig. 11 (*left*), and the weak common $\mathcal{A}$-supertree T of T_1 and T_2 obtained by adding $(T_1 - \bar{T}_1) \sqcup (T_2 - \bar{T}_2)$ to this $\bar{T}$ (*right*)

Moreover,

$$z = v_{T_1,A,C}.$$

Indeed, there are paths $z \rightsquigarrow v_A$, through y, and $z \rightsquigarrow v_C$, and therefore z is a common ancestor of v_A and v_C. Then, $v_{T_1,A,C}$ must be a node in the path $z \rightsquigarrow v_A$. Assume that it is an intermediate node of this path. If it is an intermediate node of the path $z \rightsquigarrow y$, then it will be a common ancestor of v_B, through y, and v_C, and therefore z cannot be the most recent common ancestor of these two nodes. And if $v_{T_1,A,C}$ is a node of the path $y \rightsquigarrow v_A$, then y will be an ancestor of v_C, something that, as we have seen above, cannot happen.

Let us move now to T. Since f_1 preserves paths, $f_1(y)$ is a common ancestor of v_A and v_B and $f_1(z)$ is a common ancestor of v_B and v_C, and there is a non-trivial path from $f_1(z)$ to $f_1(y)$. Let

$$y' = v_{T,A,B} \quad \text{and} \quad z' = v_{T,B,C}.$$

Then, T contains paths $f_1(y) \rightsquigarrow y'$ and $f_1(z) \rightsquigarrow z'$, and it turns out that there is a non-trivial path $z' \rightsquigarrow f_1(y)$. Indeed, there are paths from z' and from $f_1(y)$ to v_B, and therefore there must exist either a non-trivial path $z' \rightsquigarrow f_1(y)$ or a path $f_1(y) \rightsquigarrow z'$; but the latter cannot exist, because if it existed, then composing it with $z' \rightsquigarrow v_C$ we would obtain a path $f_1(y) \rightsquigarrow v_C$ that, when reflected by f_1, would entail a path $y \rightsquigarrow v_C$ in T_1 that does not exist.

In particular, there is a non-trivial path $z' \rightsquigarrow y'$ in T. Arguing as in T_1, this implies that z' is also the most recent common ancestor of v_A and v_C in T. See Fig. 14 for a representation of the structure of T between $f_1(z)$ and v_A, v_B, v_C.

Consider finally the $\mathcal{A}$-tree T_2, and set $x = v_{T_2,B,C}$. Then, $f_2(x)$ will be a common ancestor of v_B and v_C in T and therefore there will be a path $f_2(x) \rightsquigarrow z'$.

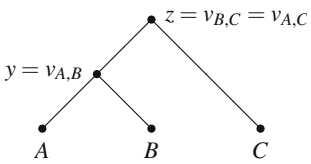

Fig. 13 The structure of T_1 above v_A, v_B, v_C. The *edges* represent paths; any one of them can be trivial, except the path $z \rightsquigarrow y$, which is non-trivial by assumption

Composing this path with $z' \rightsquigarrow v_A$ we obtain a path $f_2(x) \rightsquigarrow v_A$ which entails, since f_2 reflects paths, the existence of a path $x \rightsquigarrow v_A$. Therefore, x is also an ancestor of v_A, and thus there exists a path $x \rightsquigarrow v_{T_2,A,B}$. But then, there cannot exist a non-trivial path $v_{T_2,A,B} \rightsquigarrow x$.

This finishes the proof that T_1 and T_2 satisfy condition (C2).

(ii)$\Longrightarrow$(iii) Assume that T_1 and T_2 satisfy conditions (C1) and (C2).

Let $A \in \mathcal{A}(T_1) = \mathcal{A}(T_2)$. The smallest members of $\mathcal{C}_{\mathcal{A}}(T_1)$ and $\mathcal{C}_{\mathcal{A}}(T_2)$ containing A are, of course, $\mathcal{A}(v_{T_1,A})$ and $\mathcal{A}(v_{T_2,A})$, respectively. Now, the inequality $\mathcal{A}(v_{T_1,A}) \neq \mathcal{A}(v_{T_2,A})$ violates property (C1): if, say, there exists a label $B \in \mathcal{A}(v_{T_1,A}) - \mathcal{A}(v_{T_2,A})$, then T_1 contains a path $v_A \rightsquigarrow v_B$ but T_2 does not contain the corresponding path $v_A \rightsquigarrow v_B$. This proves the first condition in point (iii).

Let now $X = \mathcal{A}_{T_1}(x) \in \mathcal{C}_{\mathcal{A}}(T_1)$ and $Y = \mathcal{A}_{T_2}(y) \in \mathcal{C}_{\mathcal{A}}(T_2)$ be such that $X \cap Y \neq \emptyset$, say $B \in X \cap Y$. If none of them is included into the other one, then there exist labels $A \in X - Y$ and $C \in Y - X$. Then, $C \notin \mathcal{A}(v_{T_1,A,B})$, because, since x is a common ancestor of v_A and v_B, there is a path $x \rightsquigarrow v_{T_1,A,B}$ that entails the inclusion $\mathcal{A}(v_{T_1,A,B}) \subseteq \mathcal{A}(x)$, and by assumption $C \notin \mathcal{A}(x)$. Therefore, $v_{T_1,B,C}$ is "above" $v_{T_1,A,B}$, that is, there exists a non-trivial path from $v_{B,C}$ to $v_{T_1,A,B}$: since $B \in \mathcal{A}(v_{T_1,A,B}) \cap \mathcal{A}(v_{T_1,B,C})$, if this path does not exist, then there must exist a path $v_{T_1,A,B} \rightsquigarrow v_{T_1,B,C}$ that will entail that $C \in \mathcal{A}(v_{T_1,A,B})$.

In a similar way, we have that $A \notin \mathcal{A}(v_{T_2,B,C})$ and this entails a path $v_{T_2,A,B} \rightsquigarrow v_{T_2,B,C}$ in T_2.

In all, if there exist $X \in \mathcal{C}_{\mathcal{A}}(T_1)$ and $Y \in \mathcal{C}_{\mathcal{A}}(T_2)$ such that $X \cap Y \neq \emptyset$, but $X \not\subseteq Y$ and $Y \not\subseteq X$, then there exist three labels $A, B, C \in \mathcal{A}(T_1) \cap \mathcal{A}(T_2)$ and non-trivial paths $v_{T_1,B,C} \rightsquigarrow v_{T_1,A,B}$ in T_1 and $v_{T_2,A,B} \rightsquigarrow v_{T_2,B,C}$ in T_2, which would contradict the assumption that T_1 and T_2 satisfy condition (C2).

(iii)$\Longrightarrow$(iv) Assume that T_1 and T_2 satisfy the conditions stated in point (iii). Notice that the first condition in (iii) entails that $\mathcal{L}(T_1) = \mathcal{L}(T_2)$, because labels of leaves in an $\mathcal{A}$-tree are characterized by the fact that the smallest member of the cluster representation containing the label is a singleton.

To simplify the notations, we shall denote the join of T_1 and T_2 by simply T. In this case, since $\mathcal{A}(T_1) = \mathcal{A}(T_2)$, this join T is obtained using the construction given in Definition 10.(a). Let us check that it is an $\mathcal{A}$-tree:

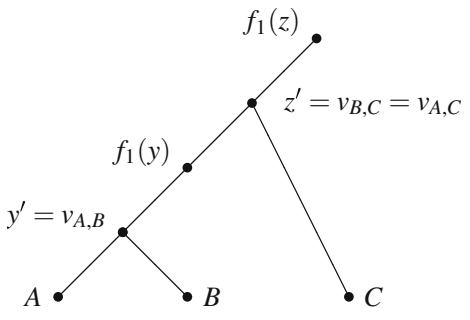

Fig. 14 The structure of T above v_A, v_B, v_C. The *edges* represent paths; any one of them can be trivial, except the path $z' \rightsquigarrow f_1(y)$, which is non-trivial

- It is clear that its leaves are the nodes of the form $w_{A,1}$, and they are labeled. The nodes of T are injectively labeled: it is impossible the existence of two different sets of labels $Y_1, Y_2 \in \mathcal{C}$ such that

$$Y_1 = \left(\bigcup \{Z \in \mathcal{C} \mid Z \subsetneq Y_1\}\right) \sqcup \{A\}, \qquad Y_2 = \left(\bigcup \{Z \in \mathcal{C} \mid Z \subsetneq Y_2\}\right) \sqcup \{A\},$$

because in this case $Y_1 \cap Y_2 \neq \emptyset$ and therefore $Y_1 \subsetneq Y_2$ or $Y_2 \subsetneq Y_1$, which would entail that one of them contains a member of $\mathcal{C}$ that already contains A.
As we shall see below, $\mathcal{A}(T) = \mathcal{A}(T_1) = \mathcal{A}(T_2)$.
- It is a tree. To prove it, assume first that a node $w_{Z,j}$ has two parents. Then, by construction, it must happen that $j = n_Z$ and then the parents are nodes $w_{Y_1,1}$ and $w_{Y_2,1}$ with $Y_1, Y_2 \in \mathcal{C}$, $Y_1 \neq Y_2$, such that $Z \subsetneq Y_1$, $Z \subsetneq Y_2$ and in both cases such that no other member of $\mathcal{C}$ lies strictly between Z and the corresponding Y_i. But then $Y_1 \cap Y_2 \neq \emptyset$ and therefore $Y_1 \subseteq Y_2$ or $Y_2 \subseteq Y_1$: if $Y_1, Y_2 \in \mathcal{C}_{\mathcal{A}}(T_1)$ or $Y_1, Y_2 \in \mathcal{C}_{\mathcal{A}}(T_2)$, by Lemma 1, and if each one of them belongs to a different cluster representation, by assumption. This forbids that both Y_1 and Y_2 are minimal over Z. Therefore, each $w_{Z,j}$ can have only one parent.
Now, if $X, Y \in \mathcal{C}$ and $Y \subseteq X$, there is a unique path $w_{X,i} \rightsquigarrow w_{Y,j}$ for every $i = 1, \ldots, n_X$ and $j = 1, \ldots, n_Y$ (if $X = Y$, then this happens for every $1 \leqslant j \leqslant i \leqslant n_X$). If $X = Y$, it is obvious by construction, and when $Y \subsetneq X$, if

$$Y \subsetneq Z_1 \subsetneq Z_2 \subsetneq \cdots \subsetneq Z_k \subsetneq X$$

is a maximal chain of sets of labels between Y and X with $Z_1, \ldots, Z_k \in \mathcal{C}$, then this path is obtained as the composition of paths

$$w_{X,i} \rightsquigarrow w_{X,1} \rightsquigarrow w_{Z_k,n_{Z_k}} \rightsquigarrow w_{Z_k,1} \rightsquigarrow w_{Z_{k-1},n_{Z_{k-1}}} \rightsquigarrow \cdots \rightsquigarrow w_{Z_1,1} \rightsquigarrow w_{Y,n_Y} \rightsquigarrow w_{Y,j}.$$

And this path is unique because every node has at most one parent.
Then, since $\mathcal{A}(T_1) = \mathcal{A}(T_2) \in \mathcal{C}$, because it is the cluster of the roots of both trees, every node $w_{Y,j}$ is a descendant of $w_{\mathcal{A}(T_1),1}$, that is, $w_{\mathcal{A}(T_1),1}$ is the root of T.

This $\mathcal{A}$-tree T satisfies the following properties that we shall use below:

- $\mathcal{A}(w_{Y,j}) = Y$, for every node $w_{Y,j}$.
This is easily proved by algebraic induction over the structure of T. If $Y = \{A\}$ and $j = 1$, then $w_{Y,1}$ is a leaf of T labeled A, while if $Y = \{A\}$ and $j > 1$, then the only labeled descendant of $w_{Y,j}$ in T is the leaf $w_{Y,1}$. Thus, $\mathcal{A}(w_{A,j}) = \{A\}$ for every $A \in \mathcal{L}(A_1) = \mathcal{L}(A_2)$ and $j = 1, \ldots, n_A$.
Now assume that $\mathcal{A}(w_{Z,j}) = Z$ for every $Z \subsetneq Y$ and $j = 1, \ldots, n_Z$, and let us prove it for Y and every $j = 1, \ldots, n_Y$. If $j = 1$, then the children of $w_{Y,1}$ are

the nodes w_{Z,n_Z} with $Z \subsetneq Y$ and maximal with this property. And then, if $w_{Y,1}$ is not labeled,

$$\begin{aligned}\mathcal{A}(w_{Y,1}) &= \bigcup\{\mathcal{A}(w_{Z,n_Z}) \mid Z \subsetneq Y \text{ and maximal with this property}\}\\ &= \bigcup\{\mathcal{A}(w_{Z,n_Z}) \mid Z \subsetneq Y\} = \bigcup\{Z \mid Z \subsetneq Y\} = Y\end{aligned}$$

(in the second equality we use that if $Z \subsetneq Y$, then there exists some maximal $Z_0 \subsetneq Y$ such that $Z \subseteq Z_0$, and then there exists a path $w_{Z_0,1} \rightsquigarrow w_{Z,1}$ that entails that $\mathcal{A}(w_{Z,1}) \subseteq \mathcal{A}(w_{Z_0,1})$), while, if $w_{Y,1}$ is labeled, say with label A, then

$$\begin{aligned}\mathcal{A}(w_{Y,1}) &= \left(\bigcup\{\mathcal{A}(w_{Z,n_Z}) \mid Z \subsetneq Y \text{ and maximal with this property}\}\right) \sqcup \{A\}\\ &= (\{\mathcal{A}(w_{Z,n_Z}) \mid Z \subsetneq Y\}) \sqcup \{A\} = \left(\bigcup\{Z \mid Z \subsetneq Y\}\right) \sqcup \{A\} = Y.\end{aligned}$$

Finally, if $j > 1$, then there is a path $w_{Y,j} \rightsquigarrow w_{Y,1}$ with the origin and all its intermediate nodes elementary and unlabeled, and therefore $\mathcal{A}(w_{Y,j}) = \mathcal{A}(w_{Y,1}) = Y$.

- In particular, $w_{Y,1} = v_{T,Y}$, for every $Y \in \mathcal{C}$, because, as we have just proved, $\mathcal{A}(w_{Y,1}) = Y$, and all children w_{Z,n_Z} of $w_{Y,1}$ are such that $\mathcal{A}(w_{Z,n_Z}) = Z \subsetneq Y$.

Let us prove now that $f_1 : V(T_1) \to V(T)$ is a weak topological embedding $f_1 : T_1 \to T$; by symmetry, it will be true also for T_2.

Let us check that f_1 preserves labels. Let $A \in \mathcal{A}(T_1)$ and $Y = \mathcal{A}(v_{T_1,A})$. Then, in particular, and using the notations of Definition 10, $v_{T_1,A} = v_{T_1,Y} = x^{(1)}_{Y,1}$, and hence $f_1(v_{T_1,A}) = w_{Y,1}$. We must check that this node has label A, that is, that

$$Y = \left(\bigcup\{Z \in \mathcal{C} \mid Z \subsetneq Y\}\right) \sqcup \{A\},$$

because in this case, and only in this case, $w_{Y,1}$ is labeled A.

So, assume that there exists some $Z \in \mathcal{C}$ such that $Z \subsetneq Y$ and $A \in Z$. Such a Z cannot belong to $\mathcal{C}_{\mathcal{A}}(T_1)$, and therefore there exists some $z \in V(T_2)$ such that $\mathcal{A}(z) = Z$. Since $A \in \mathcal{A}(z)$, there exists a path $z \rightsquigarrow v_{T_2,A}$ in T_2 and therefore $\mathcal{A}(v_{T_2,A}) \subseteq \mathcal{A}(z)$. But, by the first condition in (iii), $\mathcal{A}(v_A) = Y$ and therefore this inequality says $Y \subseteq Z$, which is impossible. Therefore, $A \notin Z$ for every $Z \subsetneq Y$, as we wanted to have.

Finally, let us prove that f_1 preserves and reflects paths. Let $u \rightsquigarrow v$ be a non-trivial path in T_1, so that $\mathcal{A}(v) \subseteq \mathcal{A}(u)$. If $\mathcal{A}(v) = \mathcal{A}(u)$, then $u = x^{(1)}_{\mathcal{A}(v),i}$ and $v = x^{(1)}_{\mathcal{A}(v),j}$ with $i > j$, and then by construction T contains a path from $f_1(u) = w_{\mathcal{A}(v),i}$ to $f_1(v) = w_{\mathcal{A}(v),j}$. If, on the contrary, $\mathcal{A}(v) \subsetneq \mathcal{A}(u)$, then $f_1(u) = w_{\mathcal{A}(u),i}$ and $f_1(v) = w_{\mathcal{A}(v),j}$ for some i, j, and, as we saw when we proved that T is an $\mathcal{A}$-tree, T contains a path $w_{\mathcal{A}(u),i} \rightsquigarrow w_{\mathcal{A}(v),j}$.

Conversely, let $f_1(u) \rightsquigarrow f_1(v)$ be a path in T, and assume that $f_1(u) = w_{\mathcal{A}(u),i}$ and $f_1(v) = w_{\mathcal{A}(v),j}$. Then, the existence of this path entails that

$$\mathcal{A}(v) = \mathcal{A}(w_{\mathcal{A}(v),j}) \subseteq \mathcal{A}(w_{\mathcal{A}(u),i}) = \mathcal{A}(u).$$

If this inclusion is strict, then Corollary 1 implies the existence of a path $u \rightsquigarrow v$ in T_1. On the other hand, if $\mathcal{A}(v) = \mathcal{A}(u)$, then $u = x^{(1)}_{\mathcal{A}(u),i}$ and $v = x^{(1)}_{\mathcal{A}(u),j}$ for some $1 \leqslant i, j \leqslant m_{1,\mathcal{A}(u)}$, and then the definition of f_1 implies that if T contains a path $f_1(u) \rightsquigarrow f_1(v)$, then $i > j$ and therefore there is a path $u \rightsquigarrow v$ in T_1.

This finishes the proof that $f_1 : T_1 \to T$ is a weak topological embedding.

(iv)$\Longrightarrow$(i) This implication is obvious. □

References

1. Baum, B.R.: Combining trees as a way of combining datasets for phylogenetic inference, and the desirability of combining gene trees. Taxon **41**(1), 3–10 (1992)
2. Bininda-Emonds, O.R.P. (ed.): Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life, Computational Biology, vol 4. Kluwer, Dordrecht (2004)
3. Daniel, P., Semple, C.: Supertree algorithms for nested taxa. In: Bininda-Emonds, O.R.P. (ed.) Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life. Computational Biology, vol 4, chap 7, pp. 151–171. Kluwer, Dordrecht (2004)
4. Morell, V.: TreeBASE: The roots of phylogeny. Science **273**(5275), 569–570 (1996). http://www.treebase.org
5. Page, R.D.M.: Modified mincut supertrees. In: Proceedings of the 2nd International Workshop Algorithms in Bioinformatics. Lecture Notes in Computer Science, vol. 2452, pp. 537–552. Springer, Berlin Heidelberg New York (2002)
6. Page, R.D.M.: Taxonomy, supertrees, and the tree of life. In: Bininda-Emonds, O.R.P. (ed.) Phylogenetic Supertrees: Combining information to reveal the tree of life. Computational Biology, vol. 4, pp. 247–265. Springer, Berlin Heidelberg New York (2004)
7. Ragan, M.A.: Phylogenetic inference based on matrix representation of trees. Mol. Phylogenet. Evol. **1**(1), 53–58 (1992)
8. Rosselló, F., Valiente, G.: An algebraic view of the relation between largest common subtrees and smallest common supertrees. Tech. rep., Technical University of Catalonia (2004)
9. Semple, C., Daniel, P., Hordijk, W., Page, R.D.M., Steel, M.: Supertree algorithms for ancestral divergence dates and nested taxa. Bioinformatics **20**(15), 2355–2360 (2004)
10. Semple, C., Steel, M.: Phylogenetics. Oxford University Press, Oxford (2003)
11. Semple, C., Steel, M.A.: A supertree method for rooted trees. Discrete Appl. Math. **105**(1–3), 147–158 (2000)
12. Stajich, J.E., Block, D., Boulez, K., Brenner, S.E., Chervitz, S.A., Dagdigian, C., Fuellen, G., Gilbert, J.G., Korf, I., Lapp, H., Lehvaslaiho, H., Matsalla, C., Mungall, C.J., Osborne, B.I., Pocock, M.R., Schattner, P., Senger, M., Stein, L.D., Stupka, E., Wilkinson, M.D., Birney, E.: The BioPerl toolkit: Perl modules for the life sciences. Genome Res. **12**(10), 1611–1618 (2002). http://www.bioperl.org
13. Steel, M.A., Warnow, T.: Kaikoura tree theorems: Computing the maximum agreement subtree. Inf. Process. Lett. **48**(2), 77–82 (1993)
14. Warnow, T.: Tree compatibility and inferring evolutionary history. J. Algorithms **16**(3), 388–407 (1994)