Available online at www.sciencedirect.com

SCIENCE DIRECT®

Theoretical Computer Science 351 (2006) 296–302

Theoretical Computer Science

www.elsevier.com/locate/tcs

Compatibility of unrooted phylogenetic trees is FPT

David Bryant[a], Jens Lagergren[b,*]

[a] McGill Centre for Bioinformatics, Montreal, Que., Canada

[b] Stockholm Bioinformatics Center and Department of Numerical Analysis and Computer Science, KTH, Stockholm, Sweden

Abstract

A collection of $T_1, T_2, \ldots, T_k$ of unrooted, leaf labelled (phylogenetic) trees, all with different leaf sets, is said to be *compatible* if there exists a tree T such that each tree T_i can be obtained from T by deleting leaves and contracting edges. Determining compatibility is NP-hard, and the fastest algorithm to date has worst case complexity of around $\Omega(n^k)$ time, n being the number of leaves. Here, we present an $O(nf(k))$ algorithm, proving that compatibility of unrooted phylogenetic trees is *fixed parameter tractable* (FPT) with respect to the number k of trees.

1. Introduction

The evolutionary history of a set of species is typically represented in the form of a *phylogenetic tree*, a tree with leaves labelled by the different species and interior vertices representing hypothetical ancestors. While the structure of these phylogenetic trees is extremely simple, the mathematics of phylogenetic trees quickly becomes non-trivial [10]. The combinatorics of phylogenetic trees has been a gold mine for algorithmically inclined computational biologists, yielding many interesting, and generally NP-hard, optimisation problems.

One of the more central combinatorial problems in phylogenetics has been that of *compatibility* (defined formally in Section 2). This problem was first discussed by Gordon [9], who introduced the attractive notion of subtrees as *samples* of the true evolutionary tree. Suppose we have a phylogenetic tree for a large set of species. This tree intuitively implies relationships between any subset of the set of species, thereby inducing a sub-phylogenetic tree for those species. Gordon's problem is: given a collection of phylogenetic trees for different sets of species, can we find a 'super' tree such that all the input trees are restrictions, or samples, of the larger tree.

Gordon's problem can be solved in polynomial time in one special case. We say that a phylogenetic tree is *rooted* if we have identified a node in the tree corresponding to the common ancestor of all of the species at the leaves. Hence rooted trees are directed, in the graph theoretic sense. The compatibility of rooted trees (where the resulting super tree is also rooted) can be determined in polynomial time, by adapting a dynamic programming algorithm of Aho et al. [2]. The same approach can also be applied in the case that all of the input trees are unrooted but share a species in common [11].

[*] Corresponding author.

E-mail addresses: bryant@mcb.mcgill.ca (D. Bryant), jensl@nada.kth.se (J. Lagergren).

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.10.033

The general compatibility problem (for *unrooted trees*) is, however, NP-hard, as first proven by Steel [11]. Indeed the problem is hard even if all of the input trees contain only four leaves. While the problem may be hard for a large number of small trees, Steel showed that the problem can be solved in polynomial time for a bounded number of large trees. His strategy was to consider all possible places that the unrooted trees might be rooted, testing each of the $O(n^k)$ possibilities using Aho et al.'s algorithm. This approach is clearly only practical when n and k are small.

In this note we show that compatibility of k unrooted trees is *fixed parameter tractable* (FPT) with respect to parameter k [8]. Specifically, we describe an algorithm solving the problem that runs in time $O(nf(k))$, where f is a function of k. The proof involves two key steps. First, we define a simple way of amalgamating phylogenetic trees into a graph, called the *display graph*, that will, if the trees are compatible, have bounded treewidth. Second, we show that testing compatibility can be converted into an expression in second order monadic logic on this graph, enabling us to apply the general algorithm of [3].

The structure of this paper is as follows. In Section 2 we give formal definitions of the key concepts and problems. In Section 3 we define the display graph and prove that the graph will have bounded treewidth if the input trees are compatible. In Section 4 we prove the compatibility of unrooted trees is FPT. We conclude in Section 5 with a discussion of extensions and future work.

2. Definitions

An *unrooted phylogenetic tree* T is a connected, undirected, acyclic graph with leaves (degree one vertices) labelled bijectively by the label set $\mathcal{L}(T)$ and no vertices of degree two. If $\mathcal{L}(T) = X$ for some finite set X then T is a *phylogenetic X-tree* in the terminology of [10]. A *rooted phylogenetic tree* is defined in the same way except that an internal vertex ρ, which may have degree two, is distinguished and called the *root*.

Let T be a rooted or unrooted phylogenetic tree and let e be an edge between two internal vertices of T. We use T/e to denote the *contraction* of T obtained by deleting e and identifying its incident vertices. For a set of internal edges E' we let T/E' denote the tree obtained by contracting every edge $e \in E'$. It is easy to check that the order of contractions is irrelevant.

Given a subset $Y \subseteq \mathcal{L}(T)$ for an unrooted phylogenetic tree T we let $T|_Y$ denote the tree obtained by forming the minimal subgraph of T connecting Y and then suppressing vertices of degree two. We say that $T|_Y$ is the *subtree of T induced by Y*. Induced subtrees are defined in the same way for rooted trees, except that the root of $T|_Y$ becomes the vertex in the minimal connecting subgraph that is closest to the root of T, and we supress all degree two vertices except the new root.

Let T_1 and T_2 be rooted or unrooted phylogenetic trees. We say that T_1 *displays* T_2 if T_2 is obtained by contracting edges in an induced subtree of T_1. That is, there is a set of edges E' such that $T_2 = (T_1|_Y)/E'$, where $Y = \mathcal{L}(T_2)$. A collection of phylogenetic trees $T_1, T_2, \ldots, T_k$ is *compatible* if there exists a phylogenetic tree T that displays each tree T_i.

The main problem we consider is:

Compatibility of unrooted phylogenetic trees

Instance: Unrooted phylogenetic trees $T_1, T_2, \ldots, T_k$

Parameter: k, the number of trees

Question: Does there exist an unrooted phylogenetic tree T that simultaneously displays each tree T_i?

3. The display graph and its treewidth

Suppose that T_1 displays T_2 and $Y = \mathcal{L}(T_2)$. From the definition, we can obtain T_2 from $T_1(Y)$ through a series of edge contractions, where each edge contraction involves deleting an edge and identifying its incident vertices. It follows that every vertex of $T_1(Y)$ gets mapped to a vertex of T_2. Several vertices can be mapped to the same vertex, the set of vertices mapped to a single vertex forms a connected subgraph of $T_1(Y)$. More formally we have

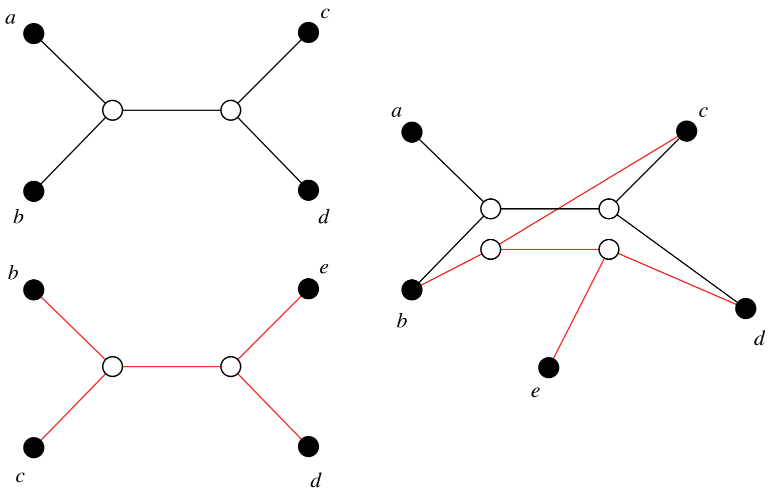

Fig. 1. Two unrooted phylogenetic trees (left) and their display graph (right).

Lemma 1. *Suppose that T_1 displays T_2 and $Y = \mathcal{L}(T_2)$. Then there exists a surjective map ϕ from a subgraph of T_1 to T_2 such that*

1. *ϕ maps labelled vertices to vertices with the same label.*
2. *For every vertex v of T_2 the set $\phi^{-1}(v)$ is a connected subgraph of T_1.*
3. *For every edge $\{u, v\}$ of T_2 there is a unique edge $\{u', v'\}$ in T_1 such that $\phi(u') = u$ and $\phi(v') = v$.*

A consequence of the definition is that if we discard vertices of T_1 not in $\phi^{-1}(T_2)$, and contract any remaining edges with endpoints mapped to the same vertex, then we will obtain a phylogenetic tree equal to T_2. We note that the map ϕ is not unique, but this is not really an issue.

Let $T_1, T_2, \ldots, T_k$ be unrooted phylogenetic trees with varying leaf sets. The *display graph* for $T_1, \ldots, T_k$ is formed from the disjoint graph union of $T_1, T_2, \ldots, T_k$ by identifying vertices with the same label. The display graph for two trees is presented in Fig. 1.

A graph G has *treewidth* k if there exists a tree T and a map B from $V(T)$ to subsets of $V(G)$ of size at most $k + 1$ such that

(TW1) For every $a \in V(G)$ there is $v \in V(T)$ such that $a \in B(v)$.
(TW2) For all edges $\{a, b\}$ in $E(G)$ there exists $v \in V(T)$ such that $\{a, b\} \subseteq B(v)$.
(TW3) For every $a \in V(G)$ the set of vertices $\{v : a \in B(v)\}$ forms a connected subgraph of T.

The sets $B(v)$, $v \in V(T)$, are called the *bags* of the decomposition. See [4] for an introduction to treewidth and its applications. Here, we will show that the treewidth of a display graph is bounded when the trees are compatible.

Theorem 1. *Let $T_1, T_2, \ldots, T_k$ be a collection of compatible, unrooted phylogenetic trees with varying leaf sets. The display graph of $T_1, \ldots, T_k$ has treewidth at most k.*

Proof. Our proof is in two parts. First we construct a tree decomposition for G. Then we prove that the tree decomposition is valid and that each bag in the decomposition has size at most $k + 1$.

Suppose that T displays $T_1, T_2, \ldots, T_k$. Let $\phi_1, \ldots, \phi_k$ be maps from subgraphs of T to $T_1, T_2, \ldots, T_k$ given by Lemma 1. We now form the disjoint union of the trees $T_1, \ldots, T_k$ and identify vertices with the same label to obtain the display graph G of $T_1, \ldots, T_k$. The maps ϕ_i can now be considered maps from subgraphs of T to subgraphs of G.

For each vertex v in T define

$$B(v) = \{\phi_i(v) : v \text{ in the domain of } \phi_i;\ 1 \leqslant i \leqslant k\}. \tag{1}$$

Hence B maps vertices v of T to subsets of the vertex set of G.

At this stage, we are most of the way towards having a tree decomposition. We have $|B(v)| \leqslant k$ for all $v \in V(T)$; (TW1) and (TW3) are satisfied (we prove this formally later). However, we need to modify the bags in order to get (TW2) to hold.

We form a new tree T' that is a subdivision of the tree T. Initially, set $T' = T$. We subdivide each edge $\{u, v\} \in E(T)$. Let $\{u_1, v_1\}, \ldots, \{u_m, v_m\}$ be the edges of G such that $u_i \in B(u)$ and $v_i \in B(v)$. There are at most k of these edges. Let $w_1, \ldots, w_m$ be new vertices in T' and replace $\{u, v\}$ by the path $u, w_1, \ldots, w_m, v$. For $i \in \{1, 2, \ldots, m\}$, let

$$B(w_i) = (B(u) \cap B(v)) \cup \{v_1, \ldots, v_i, u_i, \ldots, u_m\}.$$

The idea is that the subsets $B(u), B(w_1), \ldots, B(w_m), B(v)$ constitute a sequence of subsets from $B(u)$ to $B(v)$. The set $B(w_1)$ differs from $B(u)$ only by the addition of the element v_1. To go from $B(w_i)$ to $B(w_{i+1})$ we remove the element u_i and add the element v_{i+1}. In this way, every the endpoints of every edge $\{u_i, v_i\}$ appear in the bag $B(w_i)$, yet each bag has size at most $k + 1$.

We claim that (T', B) constitute a tree decomposition of G with width k.

1. For each vertex v of T, $B(v)$ defined by (1) has cardinality at most k. This also holds for the corresponding vertices in T'. Furthermore, for all the vertices w_j resulting from a subdivision of an edge $\{u, v\}$ in T we see that each set $B(w_j)$ has cardinality at most $k + 1$.
2. Consider $a \in V(G)$. Then a was in some tree T_i of the k trees which G displays. Thus there is $v \in T$ such that $\phi_i(v) = a$, so $a \in B(v)$. If v is the corresponding vertex in T' then $a \in B(v)$. Hence the decomposition satisfies property (TW1).
3. Choose an edge $\{u_i, v_i\} \in E(G)$ that originally came from some tree in $T_1, \ldots, T_k$. From Lemma 1 there is $\{u, v\} \in E(T)$ such that $u_i \in B(u)$ and $v_i \in B(v)$. By our construction of T' there is w_j such that $\{u_i, v_i\} \subseteq B(w_j)$. Thus the decomposition satisfies property (TW2).
4. We first claim that for each vertex $a \in G$ the set

   $$\{v \in V(T) : a \in B(v)\}$$

   is connected. If a is an unlabelled vertex of G then a originates from exactly one tree T_i. Hence $a \in B(v)$ if and only if $\phi_i(v) = a$. From Lemma 1 this set is connected. On the other hand, if a is a labelled vertex of G then $a \in B(v)$ if and only if $\phi_j(v) = a$ for some j such that T_j contains a vertex with the same label as a. For each such j, the set $\phi_j^{-1}(a)$ is connected and contains the vertex of T with the same label as a. Hence the union $\{v \in V(T) : a \in B(v)\}$ of the sets $\phi_j^{-1}(a)$ is also connected.

   There are two cases that we now need to consider in order to show that the bags of T' containing a are connected. Firstly, if $\{u, v\}$ is an edge of T and $a \in B(u) \cap B(v)$ then $a \in B(w_j)$ for every vertex on the subdivision of $\{u, v\}$ in T'. Secondly, if w_j is some vertex on the subdivision of the edge $\{u, v\}$ of T such that $a \in B(w_j) \cap B(u)$ but $a \notin B(v)$ then, by our construction of $B(w_j)$, we have that $a \in B(u) \cap B(w_1) \cap \cdots \cap B(w_j)$.

   We therefore have that the decomposition satisfies property (TW3). □

4. An FPT algorithm for compatibility

Let n be the total number of leaf labels. Given a collection of trees $T_1, T_2, \ldots, T_k$, it takes $O(nk)$ time to construct the display graph G, and $O(nf(k))$ time to determine whether or not G has treewidth k, for some function f [5]. If G does not have treewidth k then G is incompatible, by Theorem 1. If G has treewidth k then, for this reason, we have a host of powerful algorithmic tools. In particular, we have the general result of Courcelle [7] and Arnborg et al. [3] that problems in second order monadic logic can be solved in linear time (with respect to the number of vertices) on graphs with bounded treewidth. The bulk of this section is spent demonstrating how unrooted tree compatibility can be encoded as such a problem.

As mentioned above, compatibility of k *rooted* phylogenetic trees on a set of n leaves can be determined in $O(n^2k)$ time, using the algorithm of Aho et al. [1]. Any unrooted phylogenetic tree can be *rooted* along an edge e by subdividing e and making the new degree two vertex the root. From Steel [11] we have

Observation 2. *Unrooted phylogenetic trees $T_1, T_2, \ldots, T_k$ are compatible if and only if each tree T_i can be rooted along some edge in such a way that the resulting rooted trees are compatible.*

Since there are only $O(n^k)$ ways that the k trees can be rooted we obtain a polynomial (for fixed k) time algorithm for unrooted phylogenetic tree compatibility [11]. Our approach shares a lot of characteristics with this basic algorithm. We also search over the set of all possible ways to root the unrooted trees. However, we exploit the low treewidth of the display graph G to perform this search in time that is polynomial in n (but not in k).

First we formulate another characterisation of compatibility for rooted phylogenetic trees. Let $T_1', T_2', \ldots, T_k'$ be a set of rooted phylogenetic trees with roots $\rho_1, \ldots, \rho_k$. For each tree T_i' we define the set of *rooted triples* on $\mathcal{L}(T_i')$ by

$$R_i = \{ab|c : \text{the path from } a \text{ to } b \text{ does not intersect the path from } c \text{ to } \rho_i\}.$$

Let R equal $R_1 \cup \cdots \cup R_k$. The following characterisation of compatibility for rooted trees is a reformulation of Theorem 2 in [6].

Lemma 2. *Let $T_1', \ldots, T_k'$ be rooted phylogenetic trees on subsets of a leaf set L. Then $T_1', \ldots, T_k'$ are incompatible if and only if there exists $S \subseteq L$, $|S| \geqslant 3$, such that for all non-empty, proper subsets U of S there exists $uv|w \in R$ with $u \in U$, $v \in S - U$ and $w \in S$.*

The characterisation only makes intuitive sense in the context of the divide and conquer algorithm of Aho et al. [1]. A proof and discussion of this result can be found in [6,10]. For our purposes, we only need a characterisation that we can recode as second order monodic logic.

We translate the choice of roots, together with Lemma 2, into monadic, second order logic on the display graph G. Following [3], the display graph of $T_1, \ldots, T_k$ together with the trees $T_1, \ldots, T_k$, will be represented by a relational structure

$$\mathbf{G} = (V(G), E(G), L, V(T_1), \ldots, V(T_k), E(T_1), \ldots, E(T_k), R^*),$$

where L is $\bigcup_{1 \leqslant i \leqslant k} L(T_i)$ and R^* is the vertex-edge incidence relation in G. We will now describe a formula $\Phi(A)$ such that

$$\mathbf{G} \vDash \Phi(A)$$

if and only if A is a set of edges (one from each tree T_i) in which we can root $T_1, \ldots, T_k$ to make them compatible rooted trees (Observation 2).

First, for each $1 \leqslant i \leqslant k$, define $\Psi_i(u, v, X)$ to express that there is a path with vertex set $Y \subseteq X \subseteq V_i$ between the vertices u and v, i.e.

$$\Psi_i(u, v, X) \equiv C_i(X) \wedge u \in X \wedge v \in X,$$

where $C_i(X)$ expresses that $X \subseteq V_i$ and X induces a connected subgraph of G, i.e. $C_i(X)$ is

$$X \subseteq V_i \wedge (\forall Y, Z \subseteq X\ ((Y \cup Z = X) \rightarrow (\exists y \in Y, z \in Z, e \in E\ (R^*(y, e) \wedge R^*(z, e))))).$$

Similarly, for each $1 \leqslant i \leqslant k$, define $\Psi_i'(u, e, X)$ to express that there is a path with vertex set $Y \subseteq X \subseteq V_i$ between the vertices u and the edge e

$$\Psi_i'(u, e, X) \equiv u \in X \wedge C_i(X) \wedge (\exists v \in X\ (R^*(v, e))).$$

The formula $\Phi(A)$ is the conjunction of

$$\bigwedge_{1 \leqslant i \leqslant k} |A \cap E_i| = 1$$

and

$$\forall S \subseteq L\ \ (|S| \geqslant 3 \rightarrow \exists U \subseteq S\ (U \neq \emptyset \wedge U \neq S \wedge \forall u \in U, v \in (S \setminus U), w \in S\ (\neg R(u, v, w, A))),$$

where R is the relation defined above. In monadic second order logic,

$$R(u, v, w, A)$$

can be expressed as follows:

$$\bigvee_{1 \leqslant i \leqslant k} \exists Y, Z \subset V_i, x \in A \cap E_i \; (\Psi_i(u, v, Y) \wedge \Psi_i'(w, x, Z) \wedge (Y \cap Z = \emptyset)).$$

Recall that A represents a selection of edges along which each of the trees T_i is rooted. Thus $R(u, v, w, A)$ is true if there is a tree T_i such that, with the rooting implied by A, the path from u to v is vertex disjoint from the path from w to the root. In otherwords, $R(u, v, w, A)$ holds if and only if $uv|w$ is a rooted triple in one of the trees rooted according to A.

Hence the problem of determining which way to root the unrooted trees to give compatible rooted trees can be translated into second order monadic logic, defined on the display graph G. As G has treewidth at most k we obtain

Theorem 3. *Compatibility for unrooted phylogenetic trees can be solved in* $\mathrm{O}(ng(k))$ *time, for some function* $g(k)$.

Although never explicitly stated in [3], the following theorem follows by applying standard backtracing techniques to the automatons constructed in Section 5 of [3].

Theorem 4. *Let* $\Phi(X_1, \ldots, X_l)$ *be a monadic second order property and* K *a class of graphs of bounded treewidth. Given* $\mathbf{G} \in K$, *if there are sets* $A_1, \ldots, A_l$ *such that* $\mathbf{G} \vDash \Phi(A_1, \ldots, A_l)$, *those sets can be found in linear time.*

We note that when $T_1, \ldots, T_k$ are compatible, the theorem above implies that roots for the trees can be determined in linear time, so that we can then apply Aho et al.'s [1] to actually construct a phylogenetic tree that displays $T_1, \ldots, T_k$. The construction problem therefore takes $\mathrm{O}(n^2 g(k))$ time for some function g.

5. Discussion and future work

We have shown that there exists a linear time, FPT algorithm for compatibility of unrooted trees. For this result to have a significant impact in the evolutionary biology community we still need to design an algorithm that is simple, efficient, and easy to implement. This should allow us to also derive bounds for the function of k in the complexity—something that is complicated, and perhaps meaningless, with the current algorithm.

The basic result we present here can be easily extended to provide FPT algorithms for variations on the compatibility problem. In application, the trees $T_1, \ldots, T_k$ are only estimates, so it is desirable to incorporate relaxations of the strict compatibility condition. Efficient algorithms for supertrees that 'almost' display all of the input trees (where 'almost' can have a number of interpretations) would definitely be of interest to phylogeneticists.

Acknowledgements

DB was supported by NSERC Grant number 238975-01 and FQRNT Grant number 2003-NC-81840. Mike Steel posed the problem solved in this paper at the Dagstuhl seminar 03311 on Fixed Parameter Algorithms.

References

[1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[2] A.V. Aho, T.G. Sagiv, T.G. Szymanski, J.D. Ullman, Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions, SIAM J. Comput. 10 (3) (1981) 405–421.

[3] S. Arnborg, J. Lagergren, D. Seese, Easy problems for tree-decomposable graphs, J. Algorithms 12 (2) (1991) 308–340.

[4] H.L. Bodlaender, A tourist guide through treewidth, Acta Cybernet. 11 (1993) 1–21.

[5] H.L. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. 25 (6) (1996) 1305–1317.

[6] D. Bryant, M. Steel, Extension operations on sets of leaf-labelled trees, Adv. Appl. Math. 16 (1995) 425–453.

[7] B. Courcelle, The monadic second-order logic of graphs. I. Recognizable sets of finite graphs, Inform. Comput. 85 (1) (1990) 12–75.

[8] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer, Berlin, 1999.

[9] A.D. Gordon, Consensus supertrees: the synthesis of rooted trees containing overlapping sets of labeled leaves, J. Classification 3 (1986) 335–348.

[10] C. Semple, M. Steel, Phylogenetics, Oxford University Press, Oxford, 2003.

[11] M. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, J. Classification 9 (1992) 91–116.