Available online at www.sciencedirect.com
ScienceDirect
Journal of Discrete Algorithms 6 (2008) 458–471
JOURNAL OF DISCRETE ALGORITHMS
www.elsevier.com/locate/jda

A 3-approximation algorithm for the subtree distance between phylogenies

Magnus Bordewich [a], Catherine McCartin [b], Charles Semple [c,*]

[a] *Department of Computer Science, Durham University, Durham DH1 3LE, United Kingdom*
[b] *Institute of Information Sciences and Technology, Massey University, Palmerston North, New Zealand*
[c] *Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand*

Available online 5 January 2008

Abstract

In this paper, we give a (polynomial-time) 3-approximation algorithm for the rooted subtree prune and regraft distance between two phylogenetic trees. This problem is known to be NP-complete and the best previously known approximation algorithm is a 5-approximation. We also give a faster fixed-parameter algorithm for the rooted subtree prune and regraft distance than was previously known.
© 2007 Elsevier B.V. All rights reserved.

MSC: 05C05; 92D15

Keywords: Rooted subtree prune and regraft; Agreement forest

1. Introduction

Phylogenetic (evolutionary) trees are used in evolutionary biology to represent the tree-like evolution of a collection of present-day species. For many groups of species, including most mammals, this representation is appropriate. However, not all groups of species are suited to this type of representation. Collectively known as reticulation events, non-tree-like evolutionary processes such as hybridization, horizontal gene transfer, and recombination result in species being a composite of genes derived from different ancestors. Such groups of species include certain plant and fish species.

Historically, one of the main mathematical tools that has been used to understand and model reticulate evolution is the graph-theoretic operation called 'subtree prune and regraft'. Informally, this operation prunes a subtree of a rooted tree and then reattaches it to another part of the tree. The use of this tool in evolutionary biology dates back to at least 1990 [9] and has been regularly used since as a way to model reticulate evolution (see, for example, [11,12,15]). The reason for this use is that if two phylogenetic trees on the same set of species are inconsistent, but this inconsistency can be explained by a single reticulation event, then one tree can be obtained from the other by a single subtree prune and regraft operation. Moreover, if the inconsistency of the two trees requires more than one reticulation event, the

* Corresponding author. Tel.: +64 3 364 2600; fax: +64 3 364 2587.
E-mail addresses: m.j.r.bordewich@durham.ac.uk (M. Bordewich), C.M.McCartin@massey.ac.nz (C. McCartin), c.semple@math.canterbury.ac.nz (C. Semple).

1570-8667/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.jda.2007.10.002

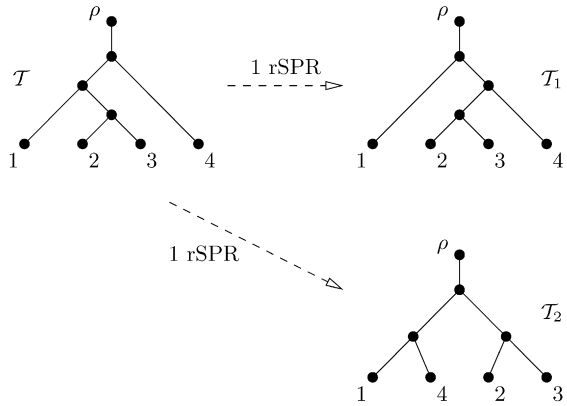

Fig. 1. Each of $\mathcal{T}_1$ and $\mathcal{T}_2$ are obtained from $\mathcal{T}$ by a single rSPR operation.

minimum number of subtree prune and regraft operations that transforms one tree into the other provides a lower bound on the number of such events. This lower bound gives an indication of the extent to which reticulation has influenced the evolutionary history of the present-day species under consideration. Here one thinks of the two initial trees as correctly representing the tree-like evolution of different parts of the genomes of the present-day species.

This paper is concerned with the problem of computing the above minimum number of operations. In the rest of this section, we formalize this problem, provide some additional background, and informally state the main results of the paper. The organization of this paper is given at the end of the section.

A *rooted binary phylogenetic X-tree* is a rooted tree whose root has degree two and all other interior vertices have degree three, and whose leaf set is X. For example, ignoring ρ and its incident edge, $\mathcal{T}$ is such a tree in Fig. 1, where $X = \{1, 2, 3, 4\}$. Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree. For the upcoming definition of a rooted subtree prune and regraft operation, we regard the root of $\mathcal{T}$ as a vertex ρ at the end of a pendant edge adjoined to the original root (see Fig. 1). Now let $e = \{u, v\}$ be an edge of $\mathcal{T}$ not incident with ρ, where u is the vertex that is in the path from ρ to v. Let $\mathcal{T}'$ be the rooted binary phylogenetic tree obtained from $\mathcal{T}$ by deleting e and then adjoining a new edge f between v and the component C_u that contains u by:

(i) creating a new vertex u' which subdivides an edge in C_u, and adjoining f between u' and v, and
(ii) contracting the degree-two vertex u.

We say that $\mathcal{T}'$ has been obtained from $\mathcal{T}$ by a single *rooted subtree prune and regraft* (rSPR) *operation*. Note that, up to isomorphism, $\mathcal{T}'$ may be equal to $\mathcal{T}$ if u' is adjacent to u. We define the rSPR *distance* between two arbitrary rooted binary phylogenetic X-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ to be the minimum number of rooted subtree prune and regraft operations that is required to transform $\mathcal{T}_1$ into $\mathcal{T}_2$. We denote this distance by $d_{\text{rSPR}}(\mathcal{T}_1, \mathcal{T}_2)$. It is well known that, for any such pair of trees, one can always obtain one from the other by a sequence of single rSPR operations. Thus this distance is well defined. Moreover, this distance is a metric on the collection of rooted binary phylogenetic X-trees. To illustrate, consider Fig. 1. Each of $\mathcal{T}_1$ and $\mathcal{T}_2$ are obtained from $\mathcal{T}$ by a single rSPR operation.

The computational problem that is the focus of this paper is the following:

Problem rSPR
Instance: Two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, and an integer k.
Question: Is $d_{\text{rSPR}}(\mathcal{T}, \mathcal{T}') \leqslant k$?

Using a characterization of this problem in terms of 'agreement forests' (see Section 2) and ideas originating from Hein et al. [10], Bordewich and Semple [3] showed that rSPR is NP-complete.

Two positive approaches for dealing with a computationally hard problem are to find polynomial-time approximation algorithms and fixed-parameter algorithms for the problem. In this paper, we give both a polynomial-time 3-approximation algorithm for rSPR, and a fixed-parameter algorithm for rSPR. The approach used in the algorithms is new and builds upon ideas used in the fixed-parameter algorithms for related problems by Hallett and McCartin [7]

and Hallett et al. [8]. A short summary of approximation and fixed-parameter algorithms as well as a comparison of these new algorithms with previous algorithms is given next.

1.1. Approximation algorithm

For a minimization problem, an algorithm is said to be an r-approximation if for all instances it guarantees to output a feasible solution which is at most r times the size of an optimal solution. The existence of polynomial-time approximation algorithms varies greatly amongst NP-hard problems. For example, for any constant r, there is no such algorithm for the general traveling salesman problem unless P = NP, while for the traveling salesman problem in the Euclidean plane there is such an algorithm for every $r > 1$ [1]. In this latter case, we say that the problem exhibits a *polynomial-time approximation scheme* (PTAS).

Using a different definition of agreement forest not corresponding to RSPR, two approximation algorithms have appeared in the literature [10,13]. Both algorithms work in a similar way and are stated as 3-approximation algorithms. However, each contains an oversight in the analysis. Nevertheless, Bonet et al. [2] show that with careful analysis these approaches give a 5-approximation algorithm for RSPR with running time $O(n)$, where $n = |X|$. Our new algorithm, which takes a different approach, improves the approximation ratio to 3, but with running time $O(n^5)$. It is known that, unless P = NP, there is no PTAS for RSPR and, in particular, no (polynomial-time) r-approximation algorithm for $r < \frac{2113}{2112}$ [4].

1.2. Fixed-parameter algorithm

The idea behind fixed-parameter complexity is that while the general case of RSPR is NP-hard, many biologically relevant cases require a relatively small number of rSPR operations and so may be tractable. In particular, if we take k as the parameter, we show that RSPR may be solved in time $O(4^k k^4 + n^3)$, where $n = |X|$. The importance of this result is in the separation of the variables n and k; it shows that, for a reasonable range of k, the problem may be tractable even for a very large n. This last algorithm greatly improves the running time of the $O((56k)^{2k} + n^3)$ fixed-parameter algorithm for RSPR given by Bordewich and Semple [3]. We refer readers unfamiliar with fixed parameterability to [6].

The paper is organized as follows. Section 2 details some notation and concepts that will be used throughout the paper. Also included in Section 2 is the above-mentioned characterization of RSPR in terms of agreement forests. This characterization is crucial to obtaining the results in this paper. In Sections 3 and 4, we describe our polynomial-time 3-approximation and $O(4^k k^4 + n^3)$ fixed-parameter algorithms for RSPR, respectively. These sections also contain the two main results of the paper: theorems stating the correctness of the algorithms. The proofs of these theorems rely on two key lemmas. The proofs of these lemmas are given in Section 5. Unless otherwise stated, the notation and terminology in this paper follows [14].

2. Preliminaries

For ease of reading, we will denote the union of two sets P and Q by $P + Q$. If $Q = \{q\}$, that is, Q is a singleton, we denote $P + Q$ by $P + q$ and $P - Q$ by $P - q$.

2.1. Phylogenetic trees, forests, and partial orders

Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree. The set X is referred to as the *label* set of $\mathcal{T}$ and is frequently denoted by $\mathcal{L}(\mathcal{T})$. A collection $\mathcal{F}$ of subtrees of $\mathcal{T}$ is a *forest* of $\mathcal{T}$ if $\mathcal{F}$ can be obtained by deleting a (possibly empty) subset of edges of $\mathcal{T}$. For a subset E of the edge set of $\mathcal{F}$, we denote the forest obtained by deleting each of the edges in E by $\mathcal{F} - E$. If $\mathcal{C}$ is a component of $\mathcal{F}$, then the intersection of X with the vertex set of $\mathcal{C}$ is referred to as the *label set of* $\mathcal{C}$.

For a forest $\mathcal{F}$, we impose a partial order on the set that is the union of the vertex and edge sets of $\mathcal{F}$. In particular, for elements g and h in this union, we write $g < h$ if g and h are in the same component of $\mathcal{F}$, $g \neq h$, and h is on the path from g to the root in this component. The set

$$\{g\colon\ g < h,\ g \text{ is a vertex or edge of } \mathcal{F}\}$$

is referred to as the set *below* h. Furthermore, if x and y are vertices of the same component of $\mathcal{F}$, the *most recent common ancestor* of x and y is the minimal vertex of $\mathcal{F}$ that is an ancestor of both x and y under this partial order. Note that, when restricted to the vertex set of $\mathcal{F}$, this partial order differs from that used in [14], but is more consistent with the other definitions used in this paper.

Lastly, let $\mathcal{C}$ be a component of $\mathcal{F}$ and let X' be a subset of the label set of $\mathcal{C}$. The minimal rooted subtree of $\mathcal{C}$ that connects the vertices of $\mathcal{C}$ labeled by the elements of X' is denoted by $\mathcal{C}(X')$. Furthermore, the *restriction* of $\mathcal{C}$ to X', denoted by $\mathcal{C}|X'$, is the rooted binary phylogenetic tree that is obtained from $\mathcal{C}(X')$ by contracting any non-root vertices of degree two.

2.2. Agreement forests

Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. For the purposes of the definitions in this subsection, we regard the root of both $\mathcal{T}$ and $\mathcal{T}'$ as a vertex ρ at the end of a pendant edge adjoined to the original root. Furthermore, we also regard ρ as part of the label sets of both $\mathcal{T}$ and $\mathcal{T}'$, thus we view their label sets as $X + \rho$.

An *agreement forest* for $\mathcal{T}$ and $\mathcal{T}'$ is a collection $\{\mathcal{T}_\rho, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ of trees, where $\mathcal{T}_\rho$ is a rooted tree with label set $\mathcal{L}_\rho$ and $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$ are rooted binary phylogenetic trees with label sets $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k$ such that the following properties are satisfied:

(i) The label sets $\mathcal{L}_\rho, \mathcal{L}_1, \ldots, \mathcal{L}_k$ partition $X + \rho$ and, in particular, $\rho \in \mathcal{L}_\rho$.
(ii) For all $i \in \{\rho, 1, 2, \ldots, k\}$, $\mathcal{T}_i = \mathcal{T}|\mathcal{L}_i = \mathcal{T}'|\mathcal{L}_i$.
(iii) The trees in $\{\mathcal{T}(\mathcal{L}_i)\colon i \in \{\rho, 1, 2, \ldots, k\}\}$ and $\{\mathcal{T}'(\mathcal{L}_i)\colon i \in \{\rho, 1, 2, \ldots, k\}\}$ are vertex-disjoint rooted subtrees of $\mathcal{T}$ and $\mathcal{T}'$, respectively.

It is easily seen that if $\mathcal{F}$ is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, then, up to contracting non-root vertices of degree two, $\mathcal{F}$ can be obtained from each of $\mathcal{T}$ and $\mathcal{T}'$ by deleting k edges. A *maximum-agreement forest* for $\mathcal{T}$ and $\mathcal{T}'$ is an agreement forest in which k (the number of components minus one) is minimized. The minimum possible value for k is denoted by $m(\mathcal{T}, \mathcal{T}')$. In Fig. 2, $\mathcal{F}_1$ and $\mathcal{F}_2$ are both agreement forests for $\mathcal{T}$ and $\mathcal{T}'$. Indeed, it is easily checked that $\mathcal{F}_1$ is a maximum-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, and so $m(\mathcal{T}, \mathcal{T}') = 2$.

Bordewich and Semple [3] showed that $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$ can be characterized in terms of agreement forests. In particular, they proved the following theorem.

Theorem 2.1. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees. Then $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = m(\mathcal{T}, \mathcal{T}')$.*

The importance of this result for us is that any r-approximation algorithm for approximating the size of a maximum-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$ equates to an r-approximation algorithm for $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$. A similar in-

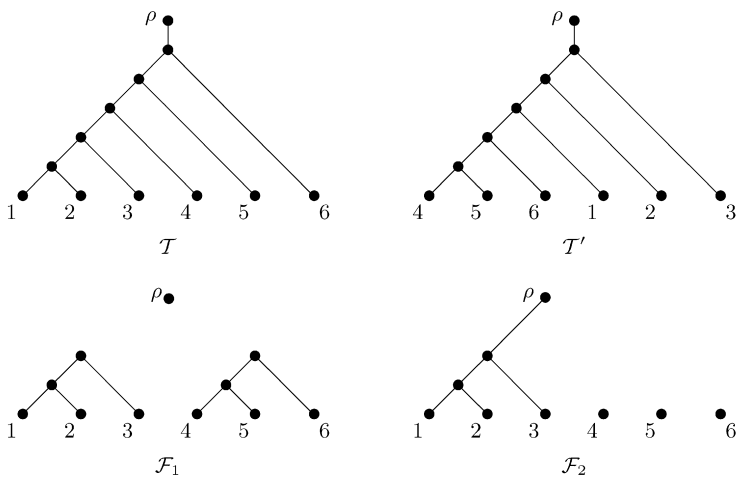

Fig. 2. $\mathcal{F}_1$ and $\mathcal{F}_2$ are agreement forests for $\mathcal{T}$ and $\mathcal{T}'$.

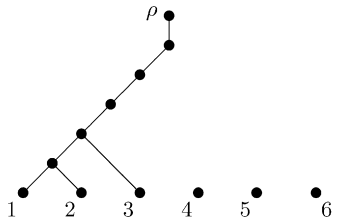

Fig. 3. A forest of $\mathcal{T}$ that yields an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$.

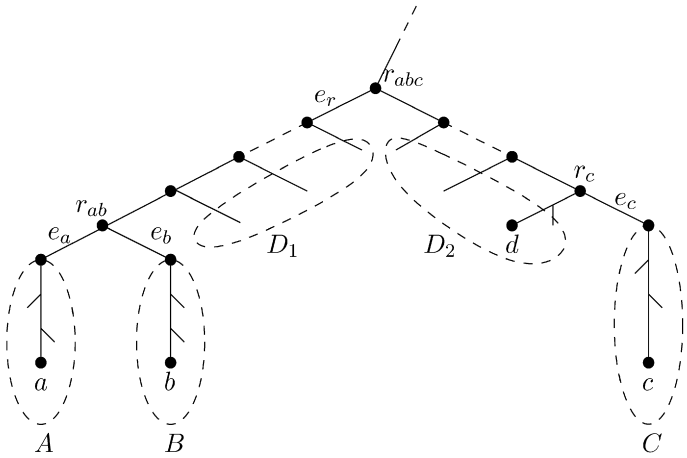

Fig. 4. The layout of a minimal incompatible triple.

terpretation can be made for fixed-parameter algorithms that find the exact size of a maximum-agreement forest for $\mathcal{T}$ and $\mathcal{T}'$.

Let $\mathcal{F}$ be a forest of $\mathcal{T}$. We say that $\mathcal{F}$ *yields* an agreement forest $\{\mathcal{T}_\rho, \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k\}$ for $\mathcal{T}$ and $\mathcal{T}'$ if $\mathcal{F}$ has components $\mathcal{C}_\rho, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ such that $\mathcal{C}_i|\mathcal{L}_i = \mathcal{T}_i$ for all $i \in \{\rho, 1, 2, \ldots, k\}$, where $\mathcal{L}_i$ is the label set of $\mathcal{C}_i$. Informally, $\mathcal{F}$ yields an agreement forest if deleting (iteratively) all degree-1 vertices that are not labeled with an element in $X + \rho$, and contracting all non-root degree-2 vertices results in the agreement forest. To illustrate, consider the two rooted phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$ shown in Fig. 2. Fig. 3 shows a forest of $\mathcal{T}$ that yields the agreement forest $\mathcal{F}_2$ for $\mathcal{T}$ and $\mathcal{T}'$ shown in Fig. 2.

We denote by $e(\mathcal{F}, \mathcal{T}')$ the size of a minimum set E of edges of $\mathcal{F}$ such that $\mathcal{F} - E$ yields an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$. This is well defined since taking E to be the set of all pendent edges of $\mathcal{F}$ yields the agreement forest consisting of isolated vertices. In the case that E is such a minimum set of edges, we say that $\mathcal{F} - E$ yields a *maximum-agreement forest for $\mathcal{F}$ and $\mathcal{T}'$*. Observe that $e(\mathcal{T}, \mathcal{T}') = m(\mathcal{T}, \mathcal{T}') = d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$.

2.3. Incompatible triples

A *triple* is a rooted binary phylogenetic tree with exactly three leaves. In the literature, triples are also called rooted triples. We denote the triple with leaf set $\{a, b, c\}$ that has the property that the path from a to b and the path from c to the root are vertex disjoint by $ab|c$ or, equivalently, $ba|c$.

Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $\mathcal{F}$ be a forest of $\mathcal{T}$. Let $\{a, b, c\}$ be a subset of X. We say that $ab|c$ is a *triple of $\mathcal{F}$* if there is a component $\mathcal{C}_i$ of $\mathcal{F}$ whose label set contains a, b, and c and has the property that $\mathcal{C}_i|\{a, b, c\}$ is $ab|c$. Analogously, $ab|c$ is a triple of $\mathcal{T}'$ if $\mathcal{T}'|\{a, b, c\}$ is $ab|c$. For example, $ab|c$ and $cd|a$ are triples of the tree shown in Fig. 4. Furthermore, $ab|c$ is an *incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$* if $ab|c$ is a triple of $\mathcal{F}$, but $ab|c$ is not a triple of $\mathcal{T}'$. For such an incompatible triple, we define r_{abc} to be the most recent common ancestor of a and c in $\mathcal{F}$ (or equivalently b and c in $\mathcal{F}$), and define r_{ab} to be the most recent common ancestor of a and b in $\mathcal{F}$.

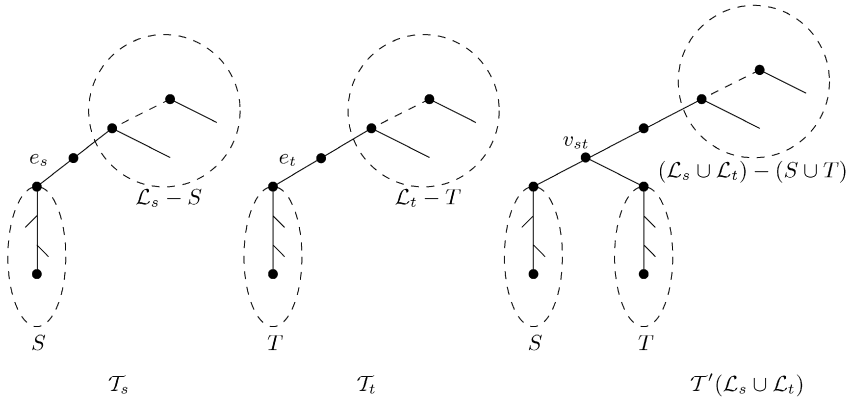

Fig. 5. The layout of a pair of overlapping components $\mathcal{T}_s$ and $\mathcal{T}_t$. The set S (resp. T) is the subset of $\mathcal{L}_s$ (resp. $\mathcal{L}_t$) whose members lie below v_{st} in $\mathcal{T}'$.

Let $ab|c$ be a minimal incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$. We denote the child edge of r_{ab} leading to a by e_a and the child edge of r_{ab} leading to b by e_b. Furthermore, we denote the child edge of r_{abc} leading to r_{ab} by e_r. Finally, let e_c denote the first edge on the path from r_{abc} to c with the property that, for all elements c' of $X - c$ below e_c, the triples $cc'|a$ and $cc'|b$ are triples of both $\mathcal{F}$ and $\mathcal{T}'$. We denote the parent vertex of the edge e_c by r_c. Note that e_c may be the parent edge of c. These definitions are illustrated in Fig. 4, where, for the moment, ignore the dashed ovals and associated labels.

Lastly, we impose a partial order on the triples of $\mathcal{F}$. In particular, we write $ab|c < xy|z$ if (i) r_{abc} is a descendant of r_{xyz} or (ii) $r_{abc} = r_{xyz}$ and r_{ab} is a descendant of r_{xy}. An incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$ is *minimal* if it is minimal with respect to this partial order.

2.4. Overlapping components

Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $\mathcal{F}$ be a forest of $\mathcal{T}$ that contains no incompatible triple with respect to $\mathcal{T}'$. Let $\mathcal{T}_s$ and $\mathcal{T}_t$ be two components of $\mathcal{F}$ with label sets $\mathcal{L}_s$ and $\mathcal{L}_t$. It is important to note that, because of this assumption on triples, $\mathcal{T}|\mathcal{L}_s = \mathcal{T}_s|\mathcal{L}_s = \mathcal{T}'|\mathcal{L}_s$ and $\mathcal{T}|\mathcal{L}_t = \mathcal{T}_t|\mathcal{L}_t = \mathcal{T}'|\mathcal{L}_t$. We say $\mathcal{T}_s$ and $\mathcal{T}_t$ *overlap* in $\mathcal{T}'$ if $\mathcal{T}'(\mathcal{L}_s)$ and $\mathcal{T}'(\mathcal{L}_t)$ share a common vertex. For such a pair of overlapping components, we define a *minimal common vertex*, v_{st} say, in $\mathcal{T}'$ to be a minimal vertex in $\mathcal{T}'(\mathcal{L}_s) \cap \mathcal{T}'(\mathcal{L}_t)$ with respect to the partial order on vertices in $\mathcal{T}'$. Furthermore, with respect to the partial order on edges of $\mathcal{F}$, we let e_s denote the minimal edge in $\mathcal{F}$ whose set of descendants in X is precisely the descendants of v_{st} in $\mathcal{L}_s$. Analogously, we let e_t denote the minimal edge in $\mathcal{F}$ whose set of descendants in X is precisely the descendants of v_{st} in $\mathcal{L}_t$. These definitions are illustrated in Fig. 5.

3. Approximation algorithm

In this section we present our polynomial-time 3-approximation algorithm for RSPR, and state the key lemma and resulting theorem proving the correctness of this algorithm. We will prove the theorem in this section, but the proof of the lemma, Lemma 3.1, is deferred until the last section.

Called SPR-APPROX, the pseudocode for the approximation algorithm is given in Algorithm 3.1, while an intuitive description of the algorithm and why it works is given below. The algorithm SPR-APPROX takes as input two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$. It proceeds by deleting edges from $\mathcal{T}$ to obtain a forest $\mathcal{F}$ of $\mathcal{T}$, until $\mathcal{F}$ yields an agreement forest of $\mathcal{T}$ and $\mathcal{T}'$. To obtain such a forest, it iteratively finds a minimal incompatible triple $ab|c$ of $\mathcal{F}$ with respect to $\mathcal{T}'$, and deletes the associated edges e_a, e_c, and e_r from $\mathcal{F}$. When there are no more incompatible triples of $\mathcal{F}$ with respect to $\mathcal{T}'$, the algorithm iteratively finds components $\mathcal{T}_s$ and $\mathcal{T}_t$ of $\mathcal{F}$ which overlap in $\mathcal{T}'$, and deletes the associated edges e_s and e_t. When there are no more overlapping components, $\mathcal{F}$ yields an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, and the algorithm outputs both the forest $\mathcal{F}$ and the number of edges that have been deleted. We show in Lemma 3.1 that, whenever we delete a set of edges from $\mathcal{F}$ corresponding to either an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$ or a pair of components in $\mathcal{F}$ that overlap in $\mathcal{T}'$, the value $e(\mathcal{F}, \mathcal{T}')$ decreases by at least one. Since we

delete at most three edges at each iteration, it follows that the entire run of the algorithm deletes at most three times more edges than the minimal possible.

The proof of the following lemma is given in the last section.

Lemma 3.1. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $\mathcal{F}$ be a forest of $\mathcal{T}$.*

(i) *If there exists a minimal incompatible triple $ab|c$ of $\mathcal{F}$ with respect to $\mathcal{T}'$, then*

$$e\big(\mathcal{F} - \{e_a, e_c, e_r\}, \mathcal{T}'\big) \leqslant e(\mathcal{F}, \mathcal{T}') - 1.$$

(ii) *If there is no incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, but there exist two components $\mathcal{T}_s$ and $\mathcal{T}_t$ of $\mathcal{F}$ that overlap in $\mathcal{T}'$, then, for some $j \in \{s, t\}$,*

$$e(\mathcal{F} - e_j, \mathcal{T}') = e(\mathcal{F}, \mathcal{T}') - 1.$$

(iii) *If there is no incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, and no two components of $\mathcal{F}$ that overlap in $\mathcal{T}'$, then*

$$e(\mathcal{F}, \mathcal{T}') = 0.$$

Theorem 3.2. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $n = |X|$. Let $(\mathcal{F}, k)$ be the output of* SPR-APPROX$(\mathcal{T}, \mathcal{T}')$. *Then $\mathcal{F}$ is an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, and k is a 3-approximation for $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$. Moreover, the running time of* SPR-APPROX *is $O(n^5)$.*

Proof. Referring to Algorithm 3.1, suppose that in the running of SPR-APPROX$(\mathcal{T}, \mathcal{T}')$ there were k_1 iterations of the first **while** loop, and k_2 iterations of the second **while** loop. We begin by showing that

$$k_1 + k_2 \leqslant d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') \leqslant 3k_1 + 2k_2 = k. \tag{3.1}$$

To this end, let $\mathcal{F}_0 = \mathcal{T}$ and, for all $i \in \{1, 2, \ldots, (k_1 + k_2)\}$, let $\mathcal{F}_i$ be the forest generated after the first i iterations of the **while** loops in SPR-APPROX$(\mathcal{T}, \mathcal{T}')$. We first prove by induction that, for all i,

$$e(\mathcal{F}_i, \mathcal{T}') + i \leqslant e(\mathcal{T}, \mathcal{T}') \leqslant e(\mathcal{F}_i, \mathcal{T}') + 3i_1 + 2i_2, \tag{3.2}$$

where $i_1 = \min\{i, k_1\}$ and $i_2 = \max\{i - k_1, 0\}$.

For $i = 0$, (3.2) trivially holds. Now suppose that (3.2) holds for all $i' < i$, where $i' \geqslant 0$. If $i \leqslant k_1$, i.e. the ith iteration is in the first **while** loop, then, by the inductive hypothesis,

$$e(\mathcal{F}_{i-1}, \mathcal{T}') + (i - 1) \leqslant e(\mathcal{T}, \mathcal{T}') \leqslant e(\mathcal{F}_{i-1}, \mathcal{T}') + 3(i - 1).$$

By Lemma 3.1(i), $e(\mathcal{F}_i, \mathcal{T}') \leqslant e(\mathcal{F}_{i-1}, \mathcal{T}') - 1$, hence $e(\mathcal{F}_i, \mathcal{T}') + i \leqslant e(\mathcal{T}, \mathcal{T}')$. Furthermore, since $\mathcal{F}_i$ has three fewer edges than $\mathcal{F}_{i-1}$, we have $e(\mathcal{F}_{i-1}, \mathcal{T}') \leqslant e(\mathcal{F}_i, \mathcal{T}') + 3$, so $e(\mathcal{T}, \mathcal{T}') \leqslant e(\mathcal{F}_i, \mathcal{T}') + 3i$ and (3.2) holds.

$\mathcal{F} \leftarrow \mathcal{T}$
$k \leftarrow 0$
while there exists an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$
do $\begin{cases} ab|c \leftarrow \text{minimal incompatible triple of } \mathcal{F} \text{ with respect to } \mathcal{T}' \\ E \leftarrow \{e_a, e_c, e_r\} \text{ with respect to } ab|c \\ \mathcal{F} \leftarrow \mathcal{F} - E \\ k \leftarrow k + 3 \end{cases}$

while there exist a pair of components in $\mathcal{F}$ that overlap in $\mathcal{T}'$
do $\begin{cases} \mathcal{T}_s, \mathcal{T}_t \leftarrow \text{components of } \mathcal{F} \text{ overlapping in } \mathcal{T}' \\ E \leftarrow \{e_s, e_t\} \text{ with respect to } \mathcal{T}_s, \mathcal{T}_t \\ \mathcal{F} \leftarrow \mathcal{F} - E \\ k \leftarrow k + 2 \end{cases}$

return $(\mathcal{F}, k)$

Algorithm 3.1. SPR-Approx$(\mathcal{T}, \mathcal{T}')$.

If $i > k_1$, then the ith iteration is in the second **while** loop. Therefore, by the inductive hypothesis,

$$e(\mathcal{F}_{i-1}, \mathcal{T}') + (i-1) \leqslant e(\mathcal{T}, \mathcal{T}') \leqslant e(\mathcal{F}_{i-1}, \mathcal{T}') + 3k_1 + 2(i - k_1 - 1).$$

By Lemma 3.1(ii), $e(\mathcal{F}_i, \mathcal{T}') \leqslant e(\mathcal{F}_{i-1}, \mathcal{T}') - 1$, and so $e(\mathcal{F}_i, \mathcal{T}') + i \leqslant e(\mathcal{T}, \mathcal{T}')$. Now $\mathcal{F}_i$ has two fewer edges than $\mathcal{F}_{i-1}$, so $e(\mathcal{F}_{i-1}, \mathcal{T}') \leqslant e(\mathcal{F}_i, \mathcal{T}') + 2$. Thus $e(\mathcal{T}, \mathcal{T}') \leqslant e(\mathcal{F}_i, \mathcal{T}') + 3k_1 + 2(i - k_1)$ and (3.2) holds.

It now follows by (3.2) that

$$e(\mathcal{F}, \mathcal{T}') + k_1 + k_2 \leqslant e(\mathcal{T}, \mathcal{T}') \leqslant e(\mathcal{F}, \mathcal{T}') + 3k_1 + 2k_2.$$

Since there are no more **while** loops to complete, Lemma 3.1(iii) implies that $e(\mathcal{F}, \mathcal{T}') = 0$. Recalling that $e(\mathcal{T}, \mathcal{T}') = d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$, we obtain (3.1). Hence k is a 3-approximation for $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$.

In order to bound the running time of SPR-APPROX, note that there are at most $O(n)$ iterations. Each iteration in the first **while** loop involves finding a minimal incompatible triple. There are $O(n^3)$ triples of $\mathcal{F}$ to consider, and a minimal incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$ can be found in time $O(n^4)$, if one exists. Once such a minimal incompatible triple is found, determining and deleting the edges e_a, e_c, and e_r can certainly be done in time $O(n^4)$. Each iteration in the second **while** loop involves finding a pair of components in $\mathcal{F}$ that overlap in $\mathcal{T}'$. There are $O(n^2)$ pairs of components of $\mathcal{F}$ to consider, and such a pair of overlapping components can be found in time $O(n^3)$, if one exists. Again, once the pair is found, determining and deleting the edges e_s and e_t is fast. Hence each iteration takes time at most $O(n^4)$ and the overall running time is $O(n^5)$ as claimed. □

4. Fixed-parameter algorithm

In this section we present our fixed-parameter algorithm, SPR-EXACT, for RSPR. Like SPR-APPROX, the proof of its correctness depends upon a key lemma. The proof of this lemma, Lemma 4.1, is deferred until the last section, while the theorem stating this correctness is established here.

The pseudocode for SPR-EXACT is given in Algorithm 4.1, while an intuitive description of the algorithm and its correctness is given below. The algorithm SPR-EXACT takes as input two rooted binary phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$, and a parameter k. It proceeds in a similar fashion to SPR-APPROX: deleting edges from $\mathcal{T}$ to obtain a forest $\mathcal{F}$ of $\mathcal{T}$, until $\mathcal{F}$ yields an agreement forest of $\mathcal{T}$ and $\mathcal{T}'$. However, instead of deleting a set E of edges from $\mathcal{F}$ at each iteration, it branches into $|E|$ computation paths with each path corresponding to the deletion of one element of E.

```
F ← T
if k < 0
  do return(no)
  else if there exists an incompatible triple of F with respect to T'
        ab|c ← minimal incompatible triple of F with respect to T'
        E ← {e_a, e_b, e_c, e_r} with respect to ab|c
        Ans_a ← SPR-EXACT(F − e_a, T', k − 1)
        Ans_b ← SPR-EXACT(F − e_b, T', k − 1)
  do    Ans_c ← SPR-EXACT(F − e_c, T', k − 1)
        Ans_r ← SPR-EXACT(F − e_r, T', k − 1)
        if Ans_a = yes or Ans_b = yes or Ans_c = yes or Ans_r = yes
           do return (yes)
           else return (no)

  else if there exists a pair of components of F that overlap in T'
        T_s, T_t ← components of F overlapping in T'
        E ← {e_s, e_t} with respect to T_s, T_t
        Ans_s ← SPR-EXACT(F − e_s, T', k − 1)
  do    Ans_t ← SPR-EXACT(F − e_t, T', k − 1)
        if Ans_s = yes or Ans_t = yes
           do return (yes)
           else return (no)

  else return (yes)
```

Algorithm 4.1. SPR-EXACT($\mathcal{T}, \mathcal{T}', k$).

As with SPR-APPROX, the algorithm SPR-EXACT begins by iteratively finding a minimal incompatible triple $ab|c$ of $\mathcal{F}$ with respect to $\mathcal{T}'$, and deleting each of the associated edges e_a, e_b, e_c and e_r from $\mathcal{F}$ in a separate computation path. When, with respect to $\mathcal{T}'$, there are no more incompatible triples between $\mathcal{F}$ and $\mathcal{T}'$, the algorithm iteratively finds components $\mathcal{T}_s$ and $\mathcal{T}_t$ of $\mathcal{F}$ which overlap in $\mathcal{T}'$, and deletes each of the associated edges e_s and e_t in a separate computation path.

The algorithm runs for at most k iterations before declaring either that along some computation path it has reached a forest $\mathcal{F}$ which yields an agreement forest for $\mathcal{T}$ and $\mathcal{T}'$, or that no such forest can be obtained by deleting k or fewer edges. We show in Lemma 4.1 that in each iteration, one of the computation paths deletes a single edge from $\mathcal{F}$ such that $e(\mathcal{F}, \mathcal{T}')$ decreases by one. This means that the algorithm does find a solution if one exists. Since we branch into at most four paths in each iteration and it turns out that each iteration takes time $O(n^4)$, it follows that the entire run of the algorithm takes time $O(4^k n^4)$, where $n = |X|$. The remark at the end of this section explains how the running time can be improved to $O(4^k k^4 + n^3)$, as claimed in the introduction.

The proof of the following lemma is given in the last section.

Lemma 4.1. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $\mathcal{F}$ be a forest of $\mathcal{T}$.*

(i) *If there exists a minimal incompatible triple $ab|c$ of $\mathcal{F}$ with respect to $\mathcal{T}'$, then, for some $i \in \{a, b, c, r\}$,*

$$e(\mathcal{F} - e_i, \mathcal{T}') = e(\mathcal{F}, \mathcal{T}') - 1.$$

(ii) *If there is no incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, but there exist two components $\mathcal{T}_s$ and $\mathcal{T}_t$ of $\mathcal{F}$ that overlap in $\mathcal{T}'$, then, for some $j \in \{s, t\}$,*

$$e(\mathcal{F} - e_j, \mathcal{T}') = e(\mathcal{F}, \mathcal{T}') - 1.$$

(iii) *If there is no incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, and no two components of $\mathcal{F}$ that overlap in $\mathcal{T}'$, then*

$$e(\mathcal{F}, \mathcal{T}') = 0.$$

Theorem 4.2. *Let $\mathcal{T}$ and $\mathcal{T}'$ be two rooted binary phylogenetic X-trees, and let $n = |X|$. Let k be an integer. Then the output of* SPR-EXACT$(\mathcal{T}, \mathcal{T}', k)$ *is 'yes' if and only if $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') \leqslant k$. Moreover, the running time of* SPR-EXACT *is $O(4^k n^4)$.*

Proof. Using induction on k, we first show that, for any forest $\mathcal{F}$ of $\mathcal{T}$, the output of SPR-EXACT$(\mathcal{F}, \mathcal{T}', k)$ is '*yes*' if and only if $e(\mathcal{F}, \mathcal{T}') \leqslant k$. Since $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') = e(\mathcal{T}, \mathcal{T}')$, it will follow that the output of SPR-EXACT$(\mathcal{T}, \mathcal{T}', k)$ is '*yes*' if and only if $d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}') \leqslant k$.

If $k = 0$, then all calls to SPR-EXACT from within SPR-EXACT$(\mathcal{F}, \mathcal{T}', k)$ will have parameter -1 and therefore return '*no*'. Thus SPR-EXACT$(\mathcal{F}, \mathcal{T}', k)$ outputs '*yes*' precisely if $\mathcal{F}$ is a forest of $\mathcal{T}'$, and so $e(\mathcal{F}, \mathcal{T}') = 0$,

Now suppose that the algorithm returns the correct answer whenever the input parameter is at most k', where $k' \geqslant 0$ and $k' + 1 = k$. First assume that $e(\mathcal{F}, \mathcal{T}') > k' + 1$. Then, for all edges e_i, we have $e(\mathcal{F} - e_i, \mathcal{T}') > k'$. Therefore, within the algorithm SPR-EXACT$(\mathcal{F}, \mathcal{T}', k' + 1)$, $Ans_i =$ *no* for all $i \in \{a, b, c, r, s, t\}$ since each call to SPR-EXACT$(\mathcal{F} - e_i, \mathcal{T}', k')$ returns '*no*'. Furthermore, since $\mathcal{F}$ is not a forest of $\mathcal{T}'$, there is either some incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, or some pair of components of $\mathcal{F}$ overlap in $\mathcal{T}'$. Hence, in this case, SPR-EXACT$(\mathcal{F}, \mathcal{T}', k)$ returns '*no*'.

Now assume that $e(\mathcal{F}, \mathcal{T}') \leqslant k' + 1$. There are three cases to consider:

(i) there exists a minimal incompatible triple $ab|c$ of $\mathcal{F}$ with respect to $\mathcal{T}'$,
(ii) there is no incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, but there exist $\mathcal{T}_s$ and $\mathcal{T}_t$, two components of $\mathcal{F}$ such that $\mathcal{T}_s$ and $\mathcal{T}_t$ overlap in $\mathcal{T}'$, and
(iii) there is no incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, and no two components of $\mathcal{F}$ that overlap in $\mathcal{T}'$.

If (i) holds, then, by Lemma 4.1(i), there is some $i \in \{a, b, c, r\}$ such that $e(\mathcal{F} - e_i, \mathcal{T}') = e(\mathcal{F}, \mathcal{T}') - 1 \leqslant k'$. Hence, by the induction hypothesis, Ans_i in SPR-EXACT$(\mathcal{F}, \mathcal{T}', k)$ returns '*yes*'. If (ii) holds, but not (i), then, by Lemma 4.1(ii), there is some $j \in \{s, t\}$ such that $e(\mathcal{F} - e_j, \mathcal{T}') = e(\mathcal{F}, \mathcal{T}') - 1 \leqslant k'$, and so, by the induction hypothesis, Ans_j in

SPR-EXACT($\mathcal{F}, \mathcal{T}', k$) returns '*yes*'. Lastly, if (iii) holds, then SPR-EXACT($\mathcal{F}, \mathcal{T}', k$) returns '*yes*'. Hence the output of SPR-EXACT($\mathcal{F}, \mathcal{T}', k'+1$) is '*yes*' if and only if $e(\mathcal{F}, \mathcal{T}') \leqslant k'+1 = k$.

We bound the running time of SPR-EXACT by induction on k. If $k = -1$, then the algorithm answers '*no*' in constant time. Now suppose that the running time of SPR-EXACT is $O(4^{k'}n^4)$ for all k', where $-1 \leqslant k' < k$. As for SPR-APPROX, determining if there exists, and if so finding, a minimal incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$ can be done in time $O(n^4)$, while determining the existence of, and finding a pair of components in $\mathcal{F}$ that overlap in $\mathcal{T}'$ can be done in time $O(n^3)$. Since the algorithm makes at most four calls to SPR-EXACT, each with parameter $k-1$, the running time is $O(n^4 + 4 \cdot 4^{k-1}n^4) = O(4^k n^4)$ as claimed. □

Remark. The running time of SPR-EXACT can be easily improved to $O(4^k k^4 + n^3)$ by first applying the kernalization of Bordewich and Semple [3]. This kernalization can be computed in time $O(n^3)$ [5] and involves two types of reductions each of which reduces the size of the label sets of the two initial trees $\mathcal{T}$ and $\mathcal{T}'$ while preserving the rSPR distance between them. At the completion of the kernalization, the resulting two rooted binary phylogenetic trees, $\hat{\mathcal{T}}$ and $\hat{\mathcal{T}}'$ say, have leaf sets of size at most $28d_{\mathrm{rSPR}}(\mathcal{T}, \mathcal{T}')$. Thus, if the size of the leaf set of $\hat{\mathcal{T}}$ is greater than $28k$ we answer '*no*'; otherwise we input $(\hat{\mathcal{T}}, \hat{\mathcal{T}}', k)$ to SPR-EXACT, which now runs in time $O(4^k k^4)$.

5. Proofs of Lemmas 3.1 and 4.1

In this section we prove the two key lemmas of the paper, namely, Lemmas 3.1 and 4.1. The proofs of these lemmas will in turn require some additional lemmas.

Let $\mathcal{F}$ be an arbitrary forest of a rooted binary phylogenetic X-tree $\mathcal{T}$, and let u and v be vertices of $\mathcal{F}$. We will write $u \sim v$ if u and v are in the same component of $\mathcal{F}$, or equivalently, if $\mathcal{F}$ contains a (undirected) path from u to v. For the purposes of this section, two forests $\mathcal{F}$ and $\mathcal{F}'$ of $\mathcal{T}$ are *isomorphic* if they consist of components $\mathcal{C}_\rho, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$ and $\mathcal{C}'_\rho, \mathcal{C}'_1, \mathcal{C}'_2, \ldots, \mathcal{C}'_k$, respectively, such that, up to the ordering of these components, the label sets of $\mathcal{C}_i$ and $\mathcal{C}'_i$ agree for all $i \in \{\rho, 1, 2, \ldots, k\}$. Essentially, two forests of $\mathcal{T}$ are isomorphic precisely if their components partition $X + \rho$ in the same way. Observe that if $\mathcal{F}$ and $\mathcal{F}'$ are isomorphic, then $\mathcal{C}_i|\mathcal{L}_i = \mathcal{C}'_i|\mathcal{L}_i$ for all $i \in \{\rho, 1, \ldots, k\}$, where $\mathcal{L}_i$ is the common label set of $\mathcal{C}_i$ and $\mathcal{C}'_i$. The first of the additional lemmas, Lemma 5.1, will be used frequently in this section.

Lemma 5.1. *Let $\mathcal{T}$ be a rooted binary phylogenetic X-tree and let $\mathcal{F}$ be a forest of $\mathcal{T}$. Let e and f be edges in the same component of $\mathcal{F}$, and let E be a subset of edges of $\mathcal{F}$ such that $f \in E$ but $e \notin E$. Let v_f be the end-vertex of f closest to e and let v_e be an end-vertex of e. If*

(i) *$v_f \sim v_e$ in $\mathcal{F} - E$, and*
(ii) *for all $x \in X + \rho$, we have $x \not\sim v_f$ in $\mathcal{F} - (E + e)$,*

then $\mathcal{F} - (E - f + e)$ is isomorphic to $\mathcal{F} - E$.

Proof. It suffices to show that if $x, y \in X + \rho$, then $x \sim y$ in $\mathcal{F} - E$ if and only if $x \sim y$ in $F - (E - f + e)$. First suppose that $x \sim y$ in $\mathcal{F} - E$, but $x \not\sim y$ in $\mathcal{F} - (E - f + e)$. Then the path from x to y in $\mathcal{F} - E$ uses e, but not f. Therefore (i) implies that either $x \sim v_f$ or $y \sim v_f$ in $\mathcal{F} - (E + e)$; a contradiction to (ii). Thus $x \sim y$ in $\mathcal{F} - (E - f + e)$.

Now suppose that $x \not\sim y$ in $\mathcal{F} - E$, but $x \sim y$ in $\mathcal{F} - (E - f + e)$. Then the path from x to y in $\mathcal{F} - (E - f + e)$ uses f, but not e. But then either $x \sim v_f$ or $y \sim v_f$ in $\mathcal{F} - (E + e)$; again a contradiction to (ii). Thus $x \not\sim y$ in $\mathcal{F} - (E - f + e)$, completing the proof of the lemma. □

Throughout the rest of this section, $\mathcal{T}$ and $\mathcal{T}'$ will always denote two rooted binary phylogenetic X-trees, and $\mathcal{F}$ will always denote a forest of $\mathcal{T}$. Also, E will denote a subset of edges of $\mathcal{F}$ such that $\mathcal{F} - E$ yields a maximum-agreement forest for $\mathcal{F}$ and $\mathcal{T}'$. Moreover, extending the notation introduced earlier, let $ab|c$ be a minimal incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$. Relative to $\mathcal{F}$, we will use A, B, and C to denote the subsets of X that are descendants of e_a, e_b, and e_c, respectively. Furthermore, D_1 and D_2 will denote those subsets of $X - (A + B + C)$ such that $ad_1|c$ is a triple of $\mathcal{F}$ for all $d_1 \in D_1$, and $cd_2|a$ is a triple of $\mathcal{F}$ for all $d_2 \in D_2$. Observe that if X' is the set of descendant labels of

r_{abc}, then D_1 and D_2 partition the set $X' - (A + B + C)$. These definitions are illustrated in Fig. 4. The above set-up will simplify the statements of the upcoming lemmas.

Lemma 5.2. *Let $ab|c$ be a minimal incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$. Then*

(i) *For all $a' \in A$, $y \in B + D_1$, and $c' \in C$, the triple $a'y|c'$ is an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$.*
(ii) *If there exist $a' \in A$ and $y \in B + D_1$ such that $a' \sim y$ in $\mathcal{F} - E$, then $c' \not\sim d'$ in $\mathcal{F} - E$ for all $c' \in C$ and $d' \in D_1 + D_2$.*

Proof. For the proof of (i), suppose that there are elements $a' \in A$, $y \in B + D_1$, and $c' \in C$ such the $a'y|c'$ is a triple of $\mathcal{T}'$. First assume that $|A|, |B|, |C| \geqslant 2$. By the minimality of $ab|c$, we have that $aa'|c'$ is a triple of $\mathcal{T}'$, and so $ay|c'$ must be a triple of $\mathcal{T}'$. Also, by the definition of e_c, the triple $cc'|a$ is a triple of $\mathcal{T}'$, so $ay|c$ is a triple of $\mathcal{T}'$. If $y \in B$, then $by|c$ is a triple of $\mathcal{T}'$ and so it follows that $ab|c$ is a triple of $\mathcal{T}'$; a contradiction. If $y \in D_1$, then, by the minimality of $ab|c$, we have that $ab|y$ is a triple of $\mathcal{T}'$. Again it follows that $ab|c$ is a triple of $\mathcal{T}'$; a contradiction. Furthermore, if $A = \{a\}$, $B = \{b\}$, or $C = \{c\}$, then the analogous arguments work, thus completing the proof of (i).

To prove (ii), suppose that there are elements $a' \in A$, $y \in B + D_1$, $c' \in C$, and $d' \in D_1 + D_2$ such that $a' \sim y$ and $c' \sim d'$ in $\mathcal{F} - E$. By (i), the components of $\mathcal{F} - E$ containing a' and y, and containing c' and d' are distinct. Furthermore, as $a'y|c'$ is an incompatible triple of $\mathcal{T}'$, either $yc'|a'$ or $a'c'|y$ is a triple of $\mathcal{T}'$. Since $a' \sim y$ and $c' \sim d'$ in $\mathcal{F} - E$, this implies that both $c'd'|a'$ and $c'd'|y$ are triples of $\mathcal{T}'$. Assume that $c' \neq c$ and $a' \neq a$. Then, as $cc'|a$ and, by minimality, $aa'|c$ are triples of $\mathcal{T}'$, it is routine to check that $cd'|a$ is a triple of $\mathcal{T}'$. If $c' = c$ or $a' = a$, an analogous but easier argument shows that $cd'|a$ is a triple of $\mathcal{T}'$. This fact about $cd'|a$ is used several times in the remainder of the proof.

There are three disjoint cases to consider depending upon the location of d': (I) d' is in D_1; (II) d' is in D_2 but is not descendant of r_c; and (III) d' is a descendent of r_c.

In (I), since the components of $\mathcal{F} - E$ containing a' and y, and c' and d' are disjoint, $a'y|d'$ is a triple of $\mathcal{F}$. Moreover, by the minimality of $ab|c$, we have that $a'y|d'$ is a triple of $\mathcal{T}'$. Therefore, as $c' \sim d'$ in $\mathcal{F} - E$, it follows that $a'y|c'$ is a triple of $\mathcal{T}'$; a contradiction to (i).

If r_c is the same as r_{abc}, then neither (II) nor (III) arises, so we may assume that r_c is not the same as r_{abc}. Then, by the definition of e_c, there is an element $d \in D_2$ that is a descendant of r_c such that either $cd|a$ or $cd|b$ is an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$. Without loss of generality, we may assume that $cd|a$ is an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$.

Consider (II). In this case, $cd|d'$ is a triple of $\mathcal{F}$. Since $cd'|a$ is a triple of $\mathcal{T}'$, but $cd|a$ is not, $cd|d'$ is not a triple of $\mathcal{T}'$. Thus $cd|d'$ is an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, contradicting the minimality of $ab|c$.

Lastly, consider (III). If $d = d'$, then $cd'|a$ is an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, contradicting the fact that $cd'|a$ is a triple of $\mathcal{T}'$. Therefore assume that $d \neq d'$. Then $dd'|c$ is a rooted triple of $\mathcal{F}$. Since $cd'|a$ is a triple of $\mathcal{T}'$, but $cd|a$ is not, $dd'|c$ is not a triple of $\mathcal{T}'$. Hence $dd'|c$ is an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, again contradicting the minimality of $ab|c$. This completes the proof of the lemma. □

Lemma 5.3. *Let $ab|c$ be an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$. Then there exists an edge $f \in E$ such that, for some $i \in \{a, b, c, r\}$, the forest $\mathcal{F} - (E - f + e_i)$ is isomorphic to $\mathcal{F} - E$.*

Proof. First suppose that, for all $a' \in A$, we have $a' \not\sim r_{ab}$ in $\mathcal{F} - E$. Then take f to be the first edge in E that is on the path from r_{ab} to a in $\mathcal{F}$. It follows by Lemma 5.1 that $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_a)$. Similarly, if $b' \not\sim r_{ab}$ (resp. $c' \not\sim r_c$) in $\mathcal{F} - E$ for all $b' \in B$ (resp. $c' \in C$), then taking f to be the first edge in E on the path from r_{ab} to b (resp. r_c to c) in $\mathcal{F}$, we have that $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_b)$ (resp. $\mathcal{F} - (E - f + e_c)$).

Now suppose that there are elements $a' \in A$, $b' \in B$, and $c' \in C$ such that $a' \sim r_{ab} \sim b'$ and $c' \sim r_c$ in $\mathcal{F} - E$. By Lemma 5.2(i), $a'b'|c'$ is not a triple in $\mathcal{T}'$, so there is an edge in E that is on the path from r_{ab} to r_c in $\mathcal{F}$. Let f be the edge in E on this path that lies closest to r_c.

There are two cases to consider depending upon the location of f. Firstly, assume that f is on the path from r_{abc} to r_c. Since $c' \sim r_c$ in $\mathcal{F} - E$, it follows by Lemma 5.2 that $d' \not\sim r_c$ in $\mathcal{F} - E$ for all $d' \in D_1 + D_2$. Therefore, by Lemma 5.1, $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_c)$. Secondly, assume that f is on the path from r_{ab} to r_{abc}.

Since f was chosen closest to r_c, we have that $c' \sim r_{abc}$ in $\mathcal{F} - E$. Thus, by Lemma 5.2, $d' \not\sim r_{abc}$ in $\mathcal{F} - E$ for all $d' \in D_1 + D_2$; otherwise $c' \sim d'$ in $\mathcal{F} - E$. Hence, by Lemma 5.1, $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_r)$. □

Lemma 5.4. *Let $ab|c$ be an incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$. Then there exists an edge $f \in E$ such that $\mathcal{F} - (E - f + \{e_a, e_c, e_r\})$ is isomorphic to a subforest of $\mathcal{F} - E$.*

Proof. Similar to the proof of Lemma 5.3, first suppose that, for all $a' \in A$ (resp. $c' \in C$), we have $a' \not\sim r_{ab}$ (resp. $c' \not\sim r_c$) in $\mathcal{F} - E$. Take f to be the first edge in E on the path from r_{ab} to a (resp. r_c to c) in $\mathcal{F}$. Then, by Lemma 5.1, $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_a)$ (resp. $\mathcal{F} - (E - f + e_c)$), and so the statement of the lemma holds. Therefore, suppose that there are elements $a' \in A$ and $c' \in C$ such that $a' \sim r_{ab}$ and $c' \sim r_c$ in $\mathcal{F} - E$.

Assume there exists some $y \in B + D_1$ such that $y \sim r_{ab} \sim a'$ in $\mathcal{F} - E$. By Lemma 5.2, $a'y|c'$ is an incompatible triple of $\mathcal{T}'$ and, for all $d' \in D_1 + D_2$, we have $c' \not\sim d'$ in $\mathcal{F} - E$. Hence $c' \not\sim y$ in $\mathcal{F} - E$, so E contains some edge on the path from r_{ab} to r_c. Now let f be the closest such edge to r_c. If f is on the path from r_{abc} to r_c, then, by Lemma 5.1, $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_c)$. If f is on the path r_{ab} to r_{abc}, then $c' \sim r_{abc}$ and so, by Lemma 5.2, $d' \not\sim r_{abc}$ for all $d' \in D_1$. Therefore, by Lemma 5.1, $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_r)$. Thus under this assumption the lemma holds.

On the other hand, now assume that there is no $y \in B + D_1$ such that $y \sim r_{ab} \sim a'$ in $\mathcal{F} - E$. Then, in particular, $b' \not\sim r_{ab}$ for all $b' \in B$. Under this assumption, take f to be the first edge in E on the path from r_{ab} to b in $\mathcal{F}$. To show that $\mathcal{F} - (E - f + \{e_a, e_c, e_r\})$ is isomorphic to a subforest of $\mathcal{F} - E$ it is enough to show that for all $x, y \in X + \rho$ such that $x \sim y$ in $\mathcal{F} - (E - f + \{e_a, e_c, e_r\})$, we have $x \sim y$ in $\mathcal{F} - E$. So, for the purposes of obtaining a contradiction, suppose that there exist $x, y \in X + \rho$ such that $x \sim y$ in $\mathcal{F} - (E - f + \{e_a, e_c, e_r\})$, but $x \not\sim y$ in $\mathcal{F} - E$. Then, in $\mathcal{F} - (E - f + \{e_a, e_c, e_r\})$, the path from x to y contains f but none of the elements in $\{e_a, e_c, e_r\}$. It follows that, without loss of generality, $x \in B$ and, moreover, that $y \notin A$. Furthermore, by Lemma 5.1, $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_b)$, and so $y \notin B$. Since e_r is not in the path from x to y in $\mathcal{F} - (E - f + \{e_a, e_c, e_r\})$, it follows that $y \in D_1$, implying that $y \sim r_{ab}$; a contradiction. This completes the proof of the lemma. □

Lemma 5.5. *Suppose that no triple of $\mathcal{F}$ is incompatible with $\mathcal{T}'$. Let $\mathcal{T}_s$ and $\mathcal{T}_t$ be two components of $\mathcal{F}$ such that $\mathcal{T}_s$ and $\mathcal{T}_t$ overlap in $\mathcal{T}'$. Then there exists an edge $f \in E$ such that, for some $i \in \{s, t\}$, the forest $\mathcal{F} - (E - f + e_i)$ is isomorphic to $\mathcal{F} - E$.*

Proof. With respect to $\mathcal{T}_s$ and $\mathcal{T}_t$, let v_{st} be a minimal common vertex of $\mathcal{T}'$. Furthermore, let S denote the subset of $\mathcal{L}_s$ that are descendants of v_{st} in $\mathcal{T}'$ and let T denote the subset of $\mathcal{L}_t$ that are descendants of v_{st} in $\mathcal{T}'$, where $\mathcal{L}_s$ and $\mathcal{L}_t$ are the label sets of $\mathcal{T}_s$ and $\mathcal{T}_t$, respectively. Recall that e_s is the minimal edge in $\mathcal{F}$ whose set of label descendants is precisely S and e_t is the minimal edge in $\mathcal{F}$ whose set of label descendants is precisely T (see Fig. 5). Since $\mathcal{F} - E$ yields a maximum-agreement forest for $\mathcal{F}$ and $\mathcal{T}'$, either (I) there is no path in $\mathcal{F} - E$ connecting an element in S with an element in $\mathcal{L}_s - S$ or (II) there is no path in $\mathcal{F} - E$ connecting an element in T with an element in $\mathcal{L}_t - T$.

Without loss of generality, we may assume that (I) holds. If $e_s \in E$, then the statement holds trivially with $f = e_s$, so suppose $e_s \notin E$ and let r_s be an end-vertex of e_s. Then either (i), for all $s' \in S$, we have $s' \not\sim r_s$ in $\mathcal{F} - E$ or (ii), for all $s'' \in \mathcal{L}_s - S$, we have $s'' \not\sim r_s$ in $\mathcal{F} - E$. If (i) holds, then fix an element $s_1 \in S$ and take f to be the first edge on the path from r_s to s_1 in $\mathcal{F}$ which is in E. If (ii) holds, then fix an element $s_2 \in \mathcal{L}_s - S$ and take f to be the first edge on the path from r_s to s_2 in $\mathcal{F}$ which is in E. In either case, Lemma 5.1 implies that $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_s)$. This completes the proof of the lemma. □

At last, we prove the two key lemmas of the paper.

Proof of Lemma 3.1. First suppose that $ab|c$ is a minimal incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$. Let E be a minimum subset of edges of $\mathcal{F}$ such that $\mathcal{F} - E$ yields a maximum-agreement forest of $\mathcal{F}$ and $\mathcal{T}'$. Note that $|E| = e(\mathcal{F}, \mathcal{T}')$. By Lemma 5.4, there exists an $f \in E$ such that $\mathcal{F} - (E - f + \{e_a, e_c, e_r\})$ is a subforest of $\mathcal{F} - E$. Hence $\mathcal{F} - (E - f + \{e_a, e_c, e_r\})$ yields an agreement forest of $\mathcal{F} - \{e_a, e_c, e_r\}$ and $\mathcal{T}'$. Thus $e(\mathcal{F} - \{e_a, e_c, e_r\}, \mathcal{T}') \leqslant |E - f| = e(\mathcal{F}, \mathcal{T}') - 1$. This inequality gives (i) in the statement of the lemma.

Now suppose $\mathcal{F}$ contains no incompatible rooted triple with respect to $\mathcal{T}'$, but it does contain two components $\mathcal{T}_s$ and $\mathcal{T}_t$ that overlap in $\mathcal{T}'$. Let E be a minimum subset of edges of $\mathcal{F}$ such that $\mathcal{F} - E$ yields a maximum-agreement forest of $\mathcal{F}$ and $\mathcal{T}'$. By Lemma 5.5, there exists an $f \in E$ and $j \in \{s, t\}$ such that $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_j)$. Thus $\mathcal{F} - (E - f + e_j)$ yields a maximum-agreement forest for $\mathcal{F}$ and $\mathcal{T}'$, and so $\mathcal{F} - (E - f + e_j)$ yields an agreement forest for $\mathcal{F} - e_j$ and $\mathcal{T}'$. Therefore

$$e(\mathcal{F} - e_j, \mathcal{T}') \leqslant |E - f| = e(\mathcal{F}, \mathcal{T}') - 1.$$

On the other hand,

$$e(\mathcal{F} - e_j, \mathcal{T}') \geqslant e(\mathcal{F}, \mathcal{T}') - \left|\{e_j\}\right| = e(\mathcal{F}, \mathcal{T}') - 1.$$

Combining the last two inequalities gives (ii) in the statement of the lemma.

Lastly, suppose that $\mathcal{F}$ contains no incompatible triple with respect to $\mathcal{T}'$, and no two components that overlap in $\mathcal{T}'$. Assume that $\mathcal{F}$ consists of components $\mathcal{C}_\rho, \mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$, with label sets $\mathcal{L}_\rho, \mathcal{L}_1, \ldots, \mathcal{L}_k$, respectively. Then, as $\mathcal{F}$ is a forest of $\mathcal{T}$, we have $\mathcal{T}|\mathcal{L}_i = \mathcal{C}_i|\mathcal{L}_i$ for all $i \in \{\rho, 1, 2, \ldots, k\}$, and the trees in $\{\mathcal{T}(\mathcal{L}_i)\colon i \in \{\rho, 1, 2, \ldots, k\}\}$ are vertex disjoint subtrees of $\mathcal{T}$. On the other hand, as $\mathcal{F}$ contains no incompatible triples with respect to $\mathcal{T}'$, every triple of $\mathcal{F}$ is a triple of $\mathcal{T}'$ and so, by [14, Theorem 6.4.1], $\mathcal{T}'|\mathcal{L}_i = \mathcal{C}_i|\mathcal{L}_i$ for all $i \in \{\rho, 1, 2, \ldots, k\}$. Furthermore, as no two components of $\mathcal{F}$ overlap in $\mathcal{T}'$, the trees in $\{\mathcal{T}'(\mathcal{L}_i)\colon i \in \{\rho, 1, 2, \ldots, k\}\}$ are vertex disjoint subtrees of $\mathcal{T}'$. Hence $\mathcal{F}$ yields the agreement forest

$$\left\{\mathcal{C}_i|\mathcal{L}_i\colon i \in \{\rho, 1, 2, \ldots, k\}\right\}$$

for $\mathcal{T}$ and $\mathcal{T}'$. Part (iii) now follows from the definition of $e(\mathcal{F}, \mathcal{T}')$. □

Proof of Lemma 4.1. Let $ab|c$ be a minimal incompatible triple of $\mathcal{F}$ with respect to $\mathcal{T}'$, and let E be a minimum subset of edges of $\mathcal{F}$ such that $\mathcal{F} - E$ yields a maximum-agreement forest of $\mathcal{F}$ and $\mathcal{T}'$. Note that $|E| = e(\mathcal{F}, \mathcal{T}')$. By Lemma 5.3, there exists an $f \in E$ and $i \in \{a, b, c, r\}$ such that $\mathcal{F} - E$ is isomorphic to $\mathcal{F} - (E - f + e_i)$. Hence $\mathcal{F} - (E - f + e_i)$ yields a maximum-agreement forest of $\mathcal{F}$ and $\mathcal{T}'$, and therefore $\mathcal{F} - (E - f + e_i)$ yields an agreement forest of $\mathcal{F} - e_i$ and $\mathcal{T}'$. Thus

$$e(\mathcal{F} - e_i, \mathcal{T}') \leqslant |E - f| = e(\mathcal{F}, \mathcal{T}') - 1.$$

Moreover,

$$e(\mathcal{F} - e_i, \mathcal{T}') \geqslant e(\mathcal{F}, \mathcal{T}') - \left|\{e_i\}\right| = e(\mathcal{F}, \mathcal{T}') - 1.$$

Combining the last two inequalities gives (i).

Parts (ii) and (iii) in the statement coincide with Lemma 3.1(ii) and (iii), and so this completes the proof of the lemma. □

Acknowledgements

The first author was supported by an EPSRC postdoctoral fellowship (EP/D063574/1), while the second and third authors were supported by the New Zealand Marsden Fund.

References

[1] S. Arora, Polynomial time approximation scheme for Euclidean TSP and other geometric problems, in: Proc. IEEE FOCS'96, 1996, pp. 2–11.
[2] M.L. Bonet, K.St. John, R. Mahindru, N. Amenta, Approximating subtree distances between phylogenies, Journal of Computational Biology 13 (2006) 1419–1434.
[3] M. Bordewich, C. Semple, On the computational complexity of the rooted subtree prune and regraft distance, Annals of Combinatorics 8 (2004) 409–423.
[4] M. Bordewich, C. Semple, Computing the minimum number of hybridisation events for a consistent evolutionary history, Discrete Applied Mathematics 155 (2007) 914–928.
[5] M. Bordewich, C. Semple, Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable, IEEE/ACM Transactions on Computational Biology and Bioinformatics 4 (2007) 458–466.
[6] R. Downey, M. Fellows, Parameterized Complexity, Springer, New York, 1998.

[7] M. Hallett, C. McCartin, A faster FPT algorithm for the maximum agreement forest problem, Theory of Computing Systems 41 (2007) 539–550.
[8] M. Hallett, C. McCartin, F. Stephens, A faster FPT algorithm to compute the hybrid number between evolutionary trees, submitted for publication.
[9] J. Hein, Reconstructing evolution of sequences subject to recombination using parsimony, Mathematical Biosciences 98 (1990) 185–200.
[10] J. Hein, T. Jing, L. Wang, K. Zhang, On the complexity of comparing evolutionary trees, Discrete Applied Mathematics 71 (1996) 153–169.
[11] W. Maddison, Gene trees in species trees, Systematic Biology 46 (1997) 523–536.
[12] L. Nakhleh, T. Warnow, C.R. Linder, K.St. John, Reconstructing reticulate evolution in species—theory and practice, Journal of Computational Biology 12 (2005) 796–811.
[13] E.M. Rodrigues, M.-F. Sagot, Y. Wakabayashi, Some approximation results for the maximum agreement forest problem, in: Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques (APPROX and RANDOM), in: Lecture Notes in Computer Science, vol. 2129, Springer, Berlin, 2001, pp. 159–169.
[14] C. Semple, M. Steel, Phylogenetics, Oxford University Press, 2003.
[15] Y.S. Song, J. Hein, Parsimonious reconstruction of sequence evolution and haplotyde blocks: finding the minimum number of recombination events, in: Algorithms in Bioinformatics (WABI 2003), in: Lecture Notes in Bioinformatics, vol. 2812, Springer, Berlin, 2003, pp. 287–302.