Relaxed Agreement Forests

Relaxed Agreement Forests

Virginia Ardévol Martínez[2], Steven Chaplick[1], Steven Kelk[1(✉)], Ruben Meuwese[1], Matúš Mihalák[1], and Georgios Stamoulis[1]

[1] Department of Advanced Computing Sciences, Maastricht University, Maastricht, The Netherlands
steven.kelk@maastrichtuniversity.nl
[2] Université Paris-Dauphine, PSL University, CNRS, LAMSADE, Paris, France

Abstract. The phylogenetic inference process can produce, for multiple reasons, conflicting hypotheses of the evolutionary history of a set X of biological entities, i.e., phylogenetic trees with the same set of leaf labels X but with distinct topologies. It is natural to wish to quantify the difference between two such trees T_1 and T_2. We introduce the problem of computing a *maximum relaxed agreement forest* (MRAF) and use this as a proxy for the dissimilarity of T_1 and T_2, which in this article we assume to be unrooted and binary. MRAF asks for a partition of the leaf labels X into a minimum number of blocks $S_1, \ldots, S_k$ such that the two subtrees induced in T_1 and T_2 by every S_i are isomorphic up to suppression of degree-2 nodes and taking the labels X into account. Unlike the earlier introduced maximum agreement forest (MAF) model, the subtrees induced by the S_i are allowed to overlap. We prove that it is NP-hard to compute MRAF, by reducing from the problem of partitioning a permutation into a minimum number of monotonic subsequences (PIMS). We further show that MRAF has a $O(\log n)$-approximation algorithm where $n = |X|$ and permits exact algorithms with single-exponential running time. When one of the trees is a caterpillar, we prove that testing whether a MRAF has size at most k can be answered in polynomial time when k is fixed. We also note that on two caterpillars the approximability of MRAF is related to that of PIMS. Finally, we establish a number of bounds on MRAF, compare its behaviour to MAF both theoretically and experimentally and discuss a number of open problems.

1 Introduction

The central challenge of phylogenetics, which is the study of phylogenetic (evolutionary) trees, is to infer a tree whose leaves are bijectively labeled by a set of species X and which accurately represents the evolutionary events that gave rise to X [23]. There are many existing techniques to infer phylogenetic trees from biological data and under a range of different objective functions [19]. The complexity of this problem arises from the fact that we typically only have indirect

R. Meuwese was supported by NWO grant *Deep kernelization for phylogenetic discordance* OCENW.KLEIN.305.

H. Fernau et al. (Eds.): SOFSEM 2024, LNCS 14519, pp. 40–54, 2024.
https://doi.org/10.1007/978-3-031-52113-3_3

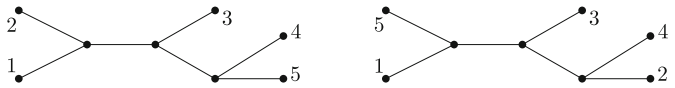

Fig. 1. The two trees, while isomorphic, are not isomorphic when taking the leaf-labeling into account, and thus both MRAF and MAF cannot be of size one. A MRAF has 2 blocks, e.g., $\{1, 2, 3\}$ and $\{4, 5\}$. A MAF has 3 blocks, e.g., $\{1, 2, 3\}$, $\{4\}$, and $\{5\}$.

data available, such as DNA sequences of the species X. Different techniques regularly yield trees with differing topologies, or the same technique constructs different trees depending on which part of a genome the DNA data is extracted from [21]. Hence, it is insightful to formally quantify the dissimilarity between (pairs of) phylogenetic trees, stimulating research into various distance measures.

Here we propose a new dissimilarity measure between unrooted phylogenetic trees T_1, T_2 which is conceptually related to the well-studied *agreement forest* abstraction. An agreement forest (AF) is a partition of X into blocks which induce, in the two input trees, non-overlapping isomorphic subtrees, modulo edge subdivision and taking the labels X into account; computing such a forest of minimum size (a MAF) is NP-hard [14] although it can be computed reasonably well in practice [26]. The AF abstraction originally derives its significance from the fact that, in unrooted (respectively, rooted) phylogenetic trees, an AF of minimum size models *Tree Bisection and Reconnection* (TBR) (respectively, *rooted Subtree Prune and Regraft*, rSPR) distance [1,6]. For background on AFs we refer to recent articles such as [7,8]. Here we propose the *relaxed agreement forest* abstraction (RAF). The only difference is that we no longer require the partition of X to induce non-overlapping subtrees; they only have to be isomorphic (see Fig. 1). We write MRAF to denote a relaxed agreement forest of minimum size. As we will observe, in the worst case MRAF can be constant while MAF grows linearly $|X|$.

The fact that RAFs are allowed to induce overlapping subtrees is potentially interesting from the perspective of biological modelling. Unlike an AF, multiple subtrees of the RAF can pass through a single branch of T_1 (or T_2). This allows us to view T_1 and T_2 as the union of several interleaved, overlapping, common evolutionary histories. It is beyond the scope of this article to expound upon this, but it is compatible with the trend in the literature of phylogenetic trees (or networks) having multiple distinct histories woven within them which sometimes evolve "in parallel" due to phenomena such as incomplete lineage sorting [9,15, 21]. This greater modelling flexibility, rather than computational tractability issues, is our primary reason for studying MRAF.

Our results are as follows. First, we show that it is NP-hard to compute a MRAF. We reduce from the problem of partitioning a permutation into a minimum number of monotone subsequences (PIMS). We show that MRAF has a $O(\log n)$-approximation algorithm where $n = |X|$ and permits exact algorithms with single-exponential running time. When one of the two trees is a caterpillar, we prove that "Is there a RAF with at most k components?" can be answered in polynomial time when k is fixed, i.e., in XP parameterized by k.

We also relate the approximability of MRAF to that of PIMS. Along the way we establish a number of bounds on MRAF, compare its behaviour to MAF and undertake an empirical analysis on two existing datasets. Due to page limits some proofs/details are deferred to an appendix in a preprint of this article [2].

2 Preliminaries, Basic Properties and Bounds

Let X be a set of labels (*taxa*) representing species. An *unrooted binary phylogenetic tree* T on X is a simple, connected, and undirected tree whose leaves are bijectively labeled with X and whose other vertices all have degree 3. When it is clear from the context we will simply write (*phylogenetic*) *tree* for shorthand. For two trees T and T' both on the same set of taxa X, we write $T = T'$ if there is an isomorphism between T and T' that preserves the labels X. Tree T is a *caterpillar* if deleting the leaves of T yields a path. We say that two distinct taxa $\{a, b\} \subseteq X$ form a *cherry* of a tree T if they have a common parent. The *identity caterpillar* on n leaves is simply the caterpillar with leaves $1, \ldots n$ in ascending order with the exception of the two cherries $\{1, 2\}$ and $\{n-1, n\}$ at its ends; see e.g. the tree on the left in Fig. 1. Note that caterpillars are almost total orders, but not quite: the leaves in the cherries at the ends are incomparable. Managing this subtle difference is a key aspect of our results.

A *quartet* is an unrooted binary phylogenetic tree with exactly four leaves. Let T be a phylogenetic tree on X. If $\{a, b, c, d\} \subseteq X$ are four distinct leaves, we say that quartet $ab|cd$ *is induced by (or simply 'is a quartet of')* T if in T the path from a to b does not intersect the path from c to d. Note that, for any four distinct leaves $a, b, c, d \in X$, exactly one of the three quartets $ab|cd, ac|bd, ad|bc$ will be a quartet of T. It is well-known that $T_1 = T_2$ if and only if both trees induce exactly the same set of quartet topologies [23]. For example, in Fig. 1 $12|45$ is a quartet of the first tree but not a quartet of the second tree. For $X' \subseteq X$, we write $T[X']$ to denote the unique, minimal subtree of T that connects all elements in the subset X'. We use $T|X'$ to denote the phylogenetic tree on X' obtained from $T[X']$ by suppressing degree-2 vertices. If $T_1|X' = T_2|X'$ then we say that the subtrees of T_1, T_2 induced by X' are *homeomorphic*.

Let T_1 and T_2 be two phylogenetic trees on X. Let $\mathcal{F} = \{S_1, \ldots, S_k\}$ be a partition of X, where each block S_i, is referred to as a *component* of $\mathcal{F}$. We say that $\mathcal{F}$ is an *agreement forest* (AF) for T_1 and T_2 if these conditions hold:

1. For each $i \in \{1, 2, ..., k\}$ we have $T_1|S_i = T_2|S_i$.
2. For each pair $i, j \in \{1, 2, ..., k\}$ with $i \neq j$, we have that $T_1[S_i]$ and $T_1[S_j]$ are vertex-disjoint in T, and $T_2[S_i]$ and $T_2[S_j]$ are vertex-disjoint in T_2.

The *size* of $\mathcal{F}$ is simply its number of components, i.e., k. Moreover, an AF with the minimum number of components (over all AFs for T_1 and T_2) is called a *maximum agreement forest* (MAF) for T_1 and T_2. For ease of reading, we will also write MAF to denote the size of a MAF. This is NP-hard to compute [1,14].

A *relaxed agreement forest* (RAF) is defined similarly to an AF, except without condition 2. A RAF with a minimum number of components is a *maximum relaxed agreement forest* (MRAF). We also use MRAF for the size of a MRAF.

Maximum Relaxed Agreement Forest (MRAF)	
Input:	Two unrooted binary phylogenetic trees T_1, T_2 on the same leaf set X, and a number k.
Task:	Partition X into at most k sets $S_1, \ldots, S_k$ where $T_1\|S_i = T_2\|S_i$ for each i.

Observation 1 follows directly from the definitions. Observation 2 shows that MAF and MRAF can behave very differently.

Observation 1. *(a) A RAF with at most* $\lceil \frac{n}{3} \rceil$ *components always exists, where* $n = |X|$*, because if* $|X'| = 3$ *and* $X' \subseteq X$ *we have* $T_1|X' = T_2|X'$ *irrespective of* X' *or the topology of* T_1 *and* T_2*. (b) MRAF is 0 if and only if* $T_1 = T_2$*. (c) A partition* $\{S_1, \ldots, S_k\}$ *of* X *is a RAF of* T_1, T_2 *if and only if, for each* S_i*, the set of quartets induced by* $T_1|S_i$ *is identical to the set of quartets induced by* $T_2|S_i$*.*

Observation 2. *There are instances where MAF is arbitrarily large,* $\Omega(n)$*, while MRAF is constant.*

Proof. Let T be an arbitrary unrooted phylogenetic binary tree on n taxa. We create two trees T_1 and T_2, both on $4n$ taxa. We build T_1 by replacing each leaf x in T with a subtree on $\{a_x, b_x, c_x, d_x\}$ in which a_x, b_x form a cherry and c_x, d_x form a cherry. The construction of T_2 is similar except that a_x, c_x form a cherry and b_x, d_x form a cherry. Note that $T_1|\{a_x, b_x, c_x, d_x\} \neq T_2|\{a_x, b_x, c_x, d_x\}$. MRAF here is 2 because we can take one component containing all the a_x, b_x taxa and one containing all the c_x, d_x taxa. However, MAF is at least n. This is because in any AF at least one of the four taxa in $\{a_x, b_x, c_x, d_x\}$ must be a singleton component, and there are n subsets of the form $\{a_x, b_x, c_x, d_x\}$. □

Given two trees T_1, T_2 on X we say that $X' \subseteq X$ induces a *maximum agreement subtree* (MAST) if $T_1|X' = T_2|X'$ and X' has maximum cardinality ranging over all such subsets. Clearly, $\lceil \frac{n}{MAST} \rceil$ is a lower bound on MRAF, since each component of a RAF is no larger than a MAST. A MAST can be computed in polynomial time [24]. The trivial upper bound on MRAF of $\lceil \frac{n}{3} \rceil$ (see Observation 1), which already contrasts sharply with the fact that the MAF of two trees can be as large as $n(1 - o(1))$ [3], can easily be strengthened via MASTs. For example, it can be verified computationally or analytically that for any two trees on 6 or more taxa, a MAST has size at least 4. We can thus repeatedly choose and remove a homeomorphic size-4 subtree, until there are fewer than 6 taxa left, giving a loose upper bound on MRAF of $n/4 + 2$. In fact, it is known that the size of a MAST on two trees with n leaves is $\Omega(\log n)$ [20] (and that this bound is asymptotically tight). In particular, the lower bound on MAST grows to infinity as n grows to infinity. Hence, the upper bound of $n/4 + 2$ can be strengthened to $n/c + f(c)$ for any arbitrary constant $c > 1$ where f is a function that only depends on c; this is thus $n/c + O(1)$. In fact, by iteratively removing $\Omega(\log n')$ of the *remaining* number of taxa n' we obtain a (slightly) sublinear upper bound on the size of a MRAF. Namely, while $n' \geq \log n + O(1)$, each iteration removes at least $d \log n' \geq d \log \log n$ leaves for some constant d, giving an upper bound of $\frac{n}{d \log \log n} + \log n + O(1)$ which is $O(\frac{n}{\log \log n})$.

Regarding lower bounds, one can generate pairs of trees on n leaves where a MAST has $O(\log n)$ leaves [18,20]. A MRAF will thus have size $\Omega(\frac{n}{\log n})$.

3 Hardness of MRAF

We discuss a related NP-hard problem regarding partitioning permutations [25].

Partition into Monotone Subsequences (PIMS) **Input:** A permutation π of $\{1,\ldots,n\}$, and a number k. **Task:** Partition $\{1,\ldots,n\}$ into at most k sets such that each set occurs monotonically in π, i.e., either as an increasing or a decreasing sequence.

Due to the classical Erdős Szekeres Theorem [10], for any n-element permutation there is a monotone sequence in π with at least $\sqrt{n}$ elements. This can be used to efficiently partition π into at most $2\sqrt{n}$ monotone sequences [4]. Thus, we may assume that the k in the problem statement is always at most $2\sqrt{n}$.

Theorem 1. *MRAF is NP-hard.*

Proof. Let (π, k) be an input to the PIMS problem, i.e., k is an integer greater than 1 and π is a permutation of $\{1,\ldots,n\}$, where we use π_i to denote the ith element of π. As remarked before, k is at most $2\sqrt{n}$. This will imply that our constructed instance of MRAF will have linear size in terms of the given permutation π, and as such any lower bounds, e.g., arising from the Exponential Time Hypothesis (ETH), will carry over from the PIMS problem to the MRAF problem. For each pair of integers (α, β) where $\alpha + \beta = k$ and $\alpha, \beta \geq 1$[1], we will construct an instance (T_1, T_2) of MRAF such that (T_1, T_2) has a solution consisting of k trees if and only if π can be partitioned into α increasing sequences and β decreasing sequences. The trees T_1 and T_2 are described as follows.

Recall that a *caterpillar* is a tree T where the subtree obtained by removing all leaves of T is a path. The path here is called the *spine* of the caterpillar. Note that, in the caterpillars used to construct T_1 and T_2, some spine vertices will have degree 2. However, to make proper binary trees one should contract any such vertex into one of its neighbors.

We first construct a leaf set $v_1,\ldots,v_n$ corresponding to the permutation. We create an identity caterpillar I whose spine is the n-vertex path $(x_1,\ldots,x_n)$ such that x_i is adjacent to v_i. Next, we create a caterpillar P whose spine is the n-vertex path $(y_1,\ldots,y_n)$ such that y_i is adjacent to v_{π_i}. Observe that already for the MRAF instance (I, P), any (r, s) partition of π leads to a solution to (I, P) consisting of k trees. However, the converse is not yet enforced. In particular, if the input to MRAF is (I, P), then the components in a MRAF (which are caterpillars) have cherries at their ends which, crucially, might be ordered differently in I than in P. This can violate monotonicity. To counter this we extend I and P to obtain T_1, T_2 as shown in Fig. 2. For T_1, we construct $8k$ caterpillars. First, for

[1] $\alpha = 0$ or $\beta = 0$ makes the problem easy.

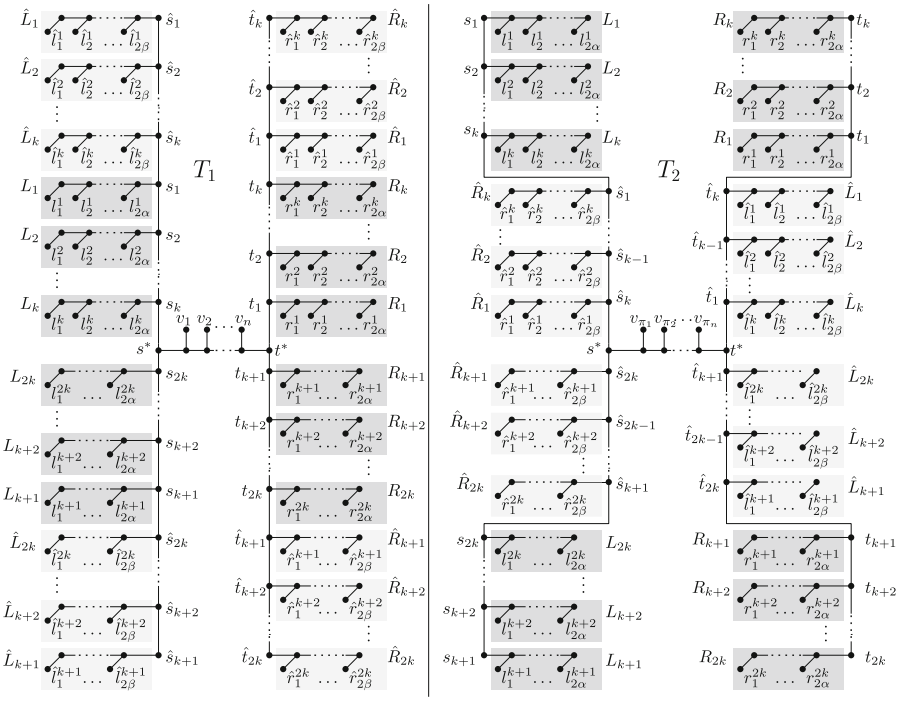

Fig. 2. The two trees T_1, T_2 constructed from an instance of PIMS in the NP-hardness proof. The dark (light) grey leaves are used to induce increasing (decreasing) subsequences in the permutation-encoding taxa in the centre of the trees.

the increasing sequences, we construct $4k$ caterpillars $L_1, \ldots, L_{2k}$, $R_1, \ldots, R_{2k}$ each having 2α leaves and 2α spine vertices. Namely, for each i,

- L_i is the caterpillar with leaf set $\{l_1^i, \ldots, l_{2\alpha}^i\}$ and spine $(w_1^i, \ldots, w_{2\alpha}^i)$ where, for each j, l_j^i is adjacent to w_j^i; and
- R_i is the caterpillar with leaf set $\{r_1^i, \ldots, r_{2\alpha}^i\}$ and spine $(z_1^i, \ldots, z_{2\alpha}^i)$ where, for each j, r_j^i is adjacent to z_j^i.

For the decreasing sequences, we similarly construct $4k$ caterpillars $\hat{L}_1, \ldots, \hat{L}_{2k}$, $\hat{R}_1, \ldots, \hat{R}_{2k}$ each having 2β leaves and 2β spine vertices. Namely, for each i,

- $\hat{L}_i$ is the caterpillar with leaf set $\{\hat{l}_1^i, \ldots, \hat{l}_{2\beta}^i\}$ and spine $(\hat{w}_1^i, \ldots, \hat{w}_{2\beta}^i)$ where, for each j, $\hat{l}_j^i$ is adjacent to $\hat{w}_j^i$; and
- $\hat{R}_i$ is the caterpillar with leaf set $\{\hat{r}_1^i, \ldots, \hat{r}_{2\beta}^i\}$ and spine $(\hat{z}_1^i, \ldots, \hat{z}_{2\beta}^i)$ where, for each j, $\hat{r}_j^i$ is adjacent to $\hat{z}_j^i$.

To form T_1, we create two $(4k+1)$-paths $Q_{\text{start}} = (\hat{s}_1, \ldots, \hat{s_k}, s_1, \ldots, s_k, s^*, s_{2k}, \ldots, s_{k+1}, \hat{s}_{2k}, \ldots, \hat{s}_{k+1})$ and $Q_{\text{end}} = (\hat{t}_k, \ldots, \hat{t}_1)\ t_k, \ldots, t_1, t^*, t_{k+1}, \ldots, \hat{t}_{2k}$ $\hat{t}_{k+1}, \ldots, \hat{t}_{2k})$ such that s^* is adjacent to x_1 (i.e., to the "start" of I) and t^* is adjacent to x_n (i.e., to the "end" of I), and for each $i \in \{1, \ldots, 2k\}$:

- s_i is adjacent to $w^i_{2\alpha}$, i.e., the "end" of L_i is attached to s_i, and t_i is adjacent to z^i_1, i.e., the "start" of R_i is attached to t_i; and
- $\hat{s}_i$ is adjacent to $\hat{w}^i_{2\alpha}$, i.e., the "end" of $\hat{L}_i$ is attached to $\hat{s}_i$, and $\hat{t}_i$ is adjacent to $\hat{z}^i_1$, i.e., the "start" of $\hat{R}_i$ is attached to $\hat{t}_i$.

To build T_2, we use the same $8k$ caterpillars $L_i, R_i, \hat{L}_i, \hat{R}_i$ but attach them differently to the "central" path P of T_2. First we make an adjustment to Q_{start} and Q_{end}. In T_2, these become: $Q_{\text{start}} = (s_1, \ldots, s_k, \hat{s}_1, \ldots, \hat{s_k}, s^*, \hat{s}_{2k}, \ldots, \hat{s}_{k+1}$ $s_{2k}, \ldots, s_{k+1})$ and $Q_{\text{end}} = (t_k, \ldots, t_1, \hat{t}_k, \ldots, \hat{t}_1, t^*, \hat{t}_{k+1}, \ldots, \hat{t}_{2k}, t_{k+1}, \ldots, \hat{t}_{2k})$ – this swap is done to highlight that in T_2 the $\hat{L}_i, \hat{R}_i$ caterpillars are closer to the central path P than the L_i, R_i caterpillars. Similar to T_1, in T_2, we have s^* adjacent to y_1 (i.e., the "start" of P) and t^* is adjacent to y_n (i.e., the "end" of P). The next part is where we see a difference regarding how we attach the caterpillars (L_i, R_i) of the increasing sequences vs. those $(\hat{L}_i, \hat{R}_i)$ of decreasing sequences.

For each $i \in \{1, \ldots, 2k\}$:

- s_i is adjacent to w^i_1, i.e., the "start" of L_i is attached to s_i and as such L_i occurs "reversed" in T_2 with respect to T_1, and
- t_i is adjacent to $z^i_{2\alpha}$, i.e., the "end" of R_i is attached to t_i.

For each $i \in \{1, \ldots, k\}$:

- $\hat{s}_{k-i+1}$ ($\hat{s}_{2k-i+1}$) is adjacent to $\hat{z}^i_{2\beta}$ ($\hat{z}^{k+i}_{2\beta}$), i.e., the "end" of $\hat{R}_i$ ($\hat{R}_{i+k}$) is attached to $\hat{s}_{k-i+1}$ (and $\hat{s}_{2k-i+1}$) and as such $\hat{R}_i$ ($\hat{R}_{k+i}$) occurs "on the opposite side" in T_2 with respect to its location in T_1, and
- $\hat{t}_{k-i+1}$ ($\hat{t}_{2k-i+1}$) is adjacent to $\hat{w}^i_1$ ($\hat{w}^{k+i}_1$), i.e., the "start" of $\hat{L}_i$ ($\hat{L}_{k+i}$) is attached to $\hat{t}_{k-i+1}$ ($\hat{t}_{2k-i+i}$).

This completes the construction of T_1 and T_2 from π. It is easy to see that this construction can be performed in polynomial time and that our trees contain precisely $16k^2 + 8k + 4 + 4n$ vertices, i.e., since $k \leq 2\sqrt{n}$, our instance of MRAF has $O(n)$ size.

Suppose π can be partitioned into α increasing sequences $\tau_1, \ldots, \tau_\alpha$ and β decreasing sequences $\sigma_1, \ldots, \sigma_\beta$. The leaf set corresponding to τ_i consists of $\{v_p : p \in \tau_i\}$ together with two leaves from each of L_j and R_j ($j \in \{1, \ldots, 2k\}$), i.e., $l^j_{2i-1}, l^j_{2i}, r^j_{2i-1}, r^j_{2i-1}$. Similarly, the leaf set corresponding to σ_i consists of $\{v_p : p \in \sigma_i\}$ together with two leaves from each of $\hat{L}_j$ and $\hat{R}_j$ ($j \in \{1, \ldots, 2k\}$), i.e., $\hat{l}^j_{2i-1}, \hat{l}^j_{2i}, \hat{r}^j_{2i-1}, \hat{r}^j_{2i}$. It can be verified that this is a valid solution to MRAF.

Now suppose that we have a solution $S_1, \ldots, S_k$ to MRAF (T_1, T_2). We need to show that this leads to a solution to the PIMS problem on π consisting of (at most) α increasing sequences and (at most) β decreasing sequences. The proof of the following lemma is in the appendix.

Lemma 1. *If some S_j uses three leaves of any caterpillar $C \in \{L_i, R_i, \hat{L}_i, \hat{R}_i : i \in \{1, \ldots, 2k\}\}$ then all elements of S_j are leaves of C.*

A consequence of this lemma is that if some S_j uses more than two leaves from any single one of our left/right caterpillars, then S_j can contain at most $\max\{2\alpha, 2\beta\} < 2k$ elements. In the next part we will see that every S_j must contain precisely $8k$ leaves from our left/right caterpillars in order to cover them all. In particular, this means that no S_j contains more than two leaves from any single left/right caterpillar. Note that, the total number of leaves is $n + 4k \cdot 2\alpha + 4k \cdot 2\beta = n + 8k^2$ where the set of n leaves is $\{v_1, \ldots, v_n\}$ (i.e., corresponding to the permutation) and the $8k^2$ leaves are the leaves of the left/right caterpillars. We now define the following eight leaf sets related to our caterpillars $L_i, R_i, \hat{L}_i, \hat{R}_i$.

- $\mathcal{L}_1 = \{l : l \text{ is a leaf of some } L_i,\ i \in \{1, \ldots, k\}\}$,
- $\mathcal{L}_2 = \{l : l \text{ is a leaf of some } L_i,\ i \in \{k+1, \ldots, 2k\}\}$,
- $\mathcal{R}_1 = \{r : r \text{ is a leaf of some } R_i,\ i \in \{1, \ldots, k\}\}$,
- $\mathcal{R}_2 = \{r : r \text{ is a leaf of some } R_i,\ i \in \{k+1, \ldots, 2k\}\}$.

The definition of $\hat{\mathcal{L}}_1, \hat{\mathcal{L}}_2, \hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2$ is analogous. The proof of the following is also deferred to the appendix.

Lemma 2. *No S_j can contain five elements where each one belongs to a different set among: $\mathcal{L}_1, \mathcal{L}_2, \mathcal{R}_1, \mathcal{R}_2, \hat{\mathcal{L}}_1, \hat{\mathcal{L}}_2, \hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2$.*

Now, observe that a component S_j can contain at most $2k$ taxa from each of the 8 sets listed above. That is because each of the 8 sets is formed from k caterpillars (e.g., $\mathcal{L}_1$ is formed from the caterpillars $L_1, ..., L_k$) and each of these k caterpillars contributes at most 2 taxa to a RAF component. (If one of the k caterpillars contributed more than 2 taxa, we would automatically be limited to at most $2k$ taxa, by Lemma 1.) It follows from this that a component S_j can in total intersect with at most $4 \times 2k = 8k$ taxa ranging over all the 8 sets: intersecting with more would require intersecting with at least 5 of the 8 sets, which as we have shown in Lemma 2 is not possible.

Given that there are k components in the RAF, and T_1, T_2 have $n + 8k^2$ taxa, each of the k components must therefore contain *exactly* $8k$ taxa from the 8 sets, and each component intersects with *exactly* 4 of the 8 sets (as this is the only way to achieve $8k$). In the appendix we prove that the only way for S_j to intersect with four sets *and* a permutation-encoding taxon v_i, is if the four sets are $\{\mathcal{L}_1, \mathcal{L}_2, \mathcal{R}_1, \mathcal{R}_2\}$ or $\{\hat{\mathcal{L}}_1, \hat{\mathcal{L}}_2, \hat{\mathcal{R}}_1, \hat{\mathcal{R}}_2\}$. The permutation-encoding taxa v_i contained in components of the first type, necessarily induce increasing subsequences, and those contained in the second type are descending. There can be at most α components of the first type, and at most β of the second, which means that the permutation π can be partitioned into at most α increasing and β decreasing sequences. This concludes the proof. □

4 Exact Algorithms

We now observe a single-exponential exact algorithm for MRAF and then show that when one input tree is a caterpillar, MRAF is in XP parameterized by k.

Recall that the NP-hard Set Cover problem (U, F), where F consists of subsets of U, is to compute a minimum-size subset of F whose union is U.

Observation 3. *Let T_1, T_2 be two unrooted binary phylogenetic trees on X. Let $U = X$ and let F be the set of all subsets of X that induce homeomorphic trees in T_1, T_2. Each RAF of T_1, T_2 is a set cover of (U, F), and each set cover of (U, F) can be transformed in polynomial time into a RAF of T_1, T_2 with the same or smaller size, by allocating each element of X to exactly one of the selected subsets. In particular, any optimum solution to the set cover instance (U, F) can be transformed in polynomial time to yield a MRAF of T_1, T_2 of the same size.*

Lemma 3. *MRAF can be solved in time $O(c^n)$, $n = |X|$, for some constant c.*

Proof. The construction in Observation 3 yields $|U| = n$ and $|F| \leq 2^n$. Minimum set cover can be solved in time $O(2^{|U|} \cdot (|U| + |F|)^{O(1)})$ thanks to [5]. □

Lemma 3 concerns general instances. When one of the given trees is a caterpillar, we can place MRAF into XP (parameterized by the solution size k). We use dynamic programming for this. We will assume that $n > 3k$, as otherwise an arbitrary partition $S_1, \ldots, S_k$ where each S_i has at most three taxa is a MRAF. For $n > 3k$ it follows that if there is a MRAF for T_1 and T_2, then there always is a MRAF $S_1, \ldots, S_k$ where no S_i is a singleton. To see this, observe that for any MRAF with a singleton S_i, it must contain a component S_j with $|S_j| \geq 3$ (since $n > 3k$), and moving any element from S_j to S_i gives another MRAF where S_i is not a singleton.

We let T_1 be the caterpillar, and T_2 an arbitrary tree. Similarly to our hardness result, we consider, without loss of generality, T_1 to consist of a spine (a path) $(y_1, \ldots, y_n)$ and leaves $v_1, \ldots, v_n$, where leaf v_i, $i = 1, \ldots, n$ is adjacent to vertex y_i. See Fig. 3 for an illustration. The spine naturally orders the leaves (up to arbitrarily breaking ties on the end cherry taxa) and this will guide our dynamic-programming approach. We write $u \prec v$ for two leaves u and v, if u appears before v in the considered ordering along the spine of T_1. We decide whether a MRAF $S_1, \ldots, S_k$ of T_1 and T_2 exists as follows: we enumerate over all possible pairs of vertices l_i, r_i, $i = 1, \ldots, k$, and check (compute) whether there exists a MRAF where the first leaf of S_i, $i = 1, \ldots, k$, is l_i and the last leaf of S_i is r_i. We call such MRAF a *MRAF constrained by* l_i, r_i, $i = 1, \ldots, k$, or simply a *constrained MRAF* if l_i and r_i are clear from the context. If for one of the guesses (enumerations) we find a constrained MRAF, we output YES, and otherwise (if for all guesses we do not find a MRAF) we output NO.

We now present our algorithm to decide, for input T_1, T_2, and pairs l_i, r_i, $i = 1, \ldots, k$, whether a constrained MRAF exists. We define $L := \{l_1, \ldots, l_n\}$ and $R := \{r_1, \ldots, r_n\}$. We view the process of computing constrained MRAF $S_1, \ldots, S_k$ as an iteration over v_i, $i = 1, \ldots, n$, and assigning $v_i \notin (L \cup R)$ to one of the components $S_1, \ldots, S_k$ (every taxon $v_i \in (L \cup R)$ is already assigned). Figure 3 illustrates this by the gray arrows from each taxon to one of the sets S_i. In the constrained MRAF, taxon v_i can only be assigned to component S_j if and only if $l_j \prec v_i \prec r_j$.

Tree T_2 further limits how taxon v_i can be assigned to components S_j (because we want that $T_1|S_j = T_2|S_j$). Clearly, for any $S_j \subset X$, $T_1|S_j$ is a caterpillar of maximum degree at most three. Thus, since l_j and r_j are the first

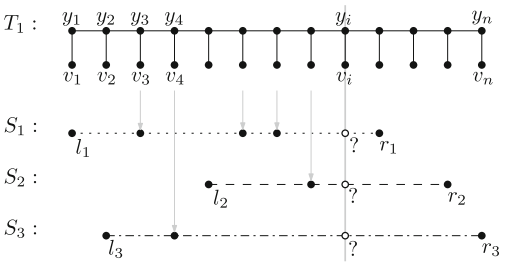

Fig. 3. Caterpillar T_1 induces a natural ordering on the taxa (leaves). The gray vertical arrows assign each taxa to one of the sets S_1, S_2, S_3. At iteration i, the question marks denote possible assignment of v_i.

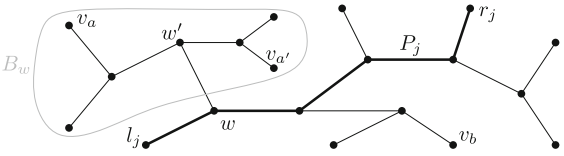

Fig. 4. A bag B_w on the (l_j, r_j)-path P_j. At most one of $v_a, v_{a'}$ can occur in S_j.

and last leaf in $T_1|S_j$, they also need to be first and last in $T_2|S_j$. Hence, the inner vertices of the unique path P_j from l_j to r_j in T_2 is the subdivision of the spine of $T_2|S_j$. For a vertex $w \in P_j$ that has a neighbor $w' \notin P_j$ we define a *bag* B_w *of* P_2 to be the maximal subtree of T_2 rooted at w' that does not include w. See Fig. 4 for an illustration. Observe that for any bag B_w of P_j, at most one taxon from B_w can be assigned to S_j. (Because if two taxa $v_a, v_{a'} \in B_w$, $a < a'$, are assigned to S_j then $l_j v_a | v_{a'} r_j$ will not be a quartet of $T_2|S_j$, while it is a quartet of $T_1|S_j$, and thus $T_1|S_j \neq T_2|S_j$.) The path P_j of $T_2|S_j$ naturally orders all bags of P_j. It follows that for two bags B_w and $B_{w'}$ where B_w appears before $B_{w'}$ in the ordering along P_j, we can select taxa $v_a \in B_w$ and $v_b \in B_{w'}$ into S_j if and only if $v_a \prec v_b$, i.e., if v_a appears before v_b in the caterpillar T_1. We write $v \prec_{P_j} v'$ for taxa v, v' such that v is from a bag B_w and v' is from a bag $B_{w'}$, and B_w appears before bag $B_{w'}$ along path P_j. Relation $\prec_{P_j}$ is thus a partial ordering of X, where any two taxa from the same bag are uncomparable. Observe now that any assignment of taxa to S_j that satisfies the above conditions, i.e., (i) for every $v_i \in S_j$, $l_j \prec v_i r_i$, (ii) for every bag B_w of P_j there is at most one vertex $v_i \in B_w \cap S_j$, and (iii) for any two taxa $v_p, v_q \in S_j$, $p < q$, $v_p \prec_{P_j} v_q$, we have $T_1|S_j = T_2|S_j$.

We can thus assign taxon v_i to component S_j whenever the previously assigned taxon $v_{i'}$ to S_j satisfies $v_{i'} \prec_{P_j} v_i$. We thus do not need to know all previously assigned taxa to S_j, only the last assigned. We compute a (partial) restricted MRAF for taxa $X_i := \{1, 2, \ldots, i\} \cup E$ iteratively for $i = 1, 2, \ldots, k$. We set $X_0 := L \cup R$. For $\boldsymbol{z} = (z_1, \ldots, z_k) \in (X \setminus (L \cup R))^k$ and $i = 0, 1, \ldots, k$ we define a boolean function $\mathrm{craf}^{(i)}(\boldsymbol{z})$ as follows: $\mathrm{craf}^{(i)}(\boldsymbol{z}) :=$ TRUE if and only if

there exists a constrained MRAF $S_1^i, S_2^i, \ldots, S_k^i$ of X_i such that the last taxon from $X_i \setminus R$ in S_ℓ^i, $\ell = 1, \ldots, k$, is z_ℓ.

Clearly, $\text{craf}^{(0)}(\boldsymbol{z}) = \text{TRUE}$ if and only if $\boldsymbol{z} = (l_1, l_2, \ldots, l_k)$. Also observe that if no z_j is equal to taxon v_i, then $\text{craf}^{(i)}(z_1, \ldots, z_k)$ is FALSE, because in every partition of X_i, the last element v_i of $X_i \setminus R$ needs to be last in one of the sets S_j. Now, whenever one of z_j is equal to v_i, the function craf^i can be computed recursively as follows:

$$\text{craf}^{(i)}(z_1, \ldots, z_{j-1}, z_j = v_i, z_{j+1} \ldots, z_k) = \bigvee_{\substack{z \in X_{i-1} \setminus R \\ z \prec_{P_j} = v_i}} \text{craf}^{(i-1)}(z_1, \ldots, z_{j-1}, z, z_{j+1}, \ldots, z_k) \quad (1)$$

This recurrence follows simply because removing v_i from every constrained MRAF of X_i gives a constrained MRAF of X_{i-1}. Now we can compute $\text{craf}^{(i)}$ bottom-up using the dynamic programming. For every value $i = 1, \ldots, k$ we enumerate $O(n^k)$ vectors $\boldsymbol{z}$, and compute the value $\text{craf}^{(i)}(\boldsymbol{z})$ using the recursive relation from Eq. (1), thus looking at at most $O(n)$ different entries of $\text{craf}^{(i-1)}$. This thus leads to the overall runtime of $O(k \cdot n^k \cdot n)$. Accounting for the enumeration of the $O(n^{2k})$ pairs l_i, r_i, $i = 1, \ldots, k$ results in the following theorem.

Theorem 2. *MRAF can be computed in time $O(k \cdot n^{3k+1})$ whenever one of the trees is a caterpillar.*

5 Approximation Algorithms

We now provide a polytime approximation algorithm for MRAF (Lemma 4) and relate the approximability of PIMS to that of MRAF on caterpillars (Lemma 5).

Lemma 4. *There is a $O(\log n)$-approximation algorithm for computing MRAF, where $n = |X|$. This algorithm cannot be better than a $(4/3)$ approximation.*

Proof. Given an instance (U, F) of Set Cover, the natural greedy algorithm yields a $O(\log |U|)$ approximation. Recall the encoding of MRAF as a Set Cover instance in Observation 3. We cannot construct this directly, since $|F|$ is potentially exponential in n, but this is not necessary to simulate the greedy algorithm. Let X' be the set of currently uncovered elements of X, initially $X = X'$. We compute a MAST of $T_1|X'$ and $T_2|X'$ in polynomial time [24]. Let S be the leaf-set of this MAST; we add this to our RAF. We then delete S from X' and iterate this process until X' is empty. Figure 6 (in the appendix) shows that this algorithm cannot be better than a (4/3) approximation. □

Lemma 5. *Let π be a permutation of $\{1, \ldots, n\}$ and let T_1 and T_2 be two caterpillars on leaves $\{1, \ldots n\}$ where T_1 is the identity caterpillar and the ith leaf of T_2 is $\pi(i)$. For any solution to the MRAF problem of size k, there is a corresponding solution to the PIMS problem of size at most $k + 2\sqrt{2k}$.*

Proof. We start with an agreement forest for the two caterpillars; each component is itself a caterpillar. We "cut off" one leaf from each end of the components in this forest. (This is because the "interior" of each component induces a monotonic subsequence, but the cherries at the end of each component potentially violate this). This leaves behind a subpermutation of π of length $2k$, which can always be partitioned into at most $2\sqrt{2k}$ monotone subsequences. □

We can create an instance of PIMS from a caterpillar instance of MRAF by treating one caterpillar as the identity and the other as the permutation. Any solution for this PIMS instance yields a feasible MRAF solution. Hence:

$$\text{MRAF} \leq \text{PIMS} \leq \text{MRAF} + 2\sqrt{2 \cdot \text{MRAF}}.$$

Recall that MRAF on caterpillars is in XP by Theorem 3. PIMS is also in XP. Specifically, the PIMS problem is equivalent to the *co-chromatic number* problem on permutation graphs, i.e., partitioning the vertices of a permutation graph into cliques and independent sets. When a graph can be partitioned into r cliques and s independent sets it is sometimes called an (r, s)*-split graph*. It is known that the perfect (r, s)-split graphs can be characterized by a finite set of forbidden induced subgraphs [17]. This implies that their recognition is in XP parameterized by r and s, i.e., when r and s are fixed, (r, s)-split graphs can be recognized in polynomial time—this was later improved to FPT [13]. These XP results are relevant here because they mean that if one of the problems has a polynomial time c-approximation, c constant, then for each fixed constant $\epsilon > 0$ the other has a polynomial-time $(c + \epsilon)$-approximation. For example, given a polynomial-time c-approximation for MRAF, and $\epsilon > 0$, we first use the XP algorithm for PIMS to check in polynomial time whether PIMS $\leq \frac{8c}{\epsilon^2}$. If so, we are done. Otherwise, the described transformation of MRAF solutions to PIMS solutions yields a $(c + \epsilon)$-approximation. The direction from PIMS to MRAF is similar. PIMS has a polynomial-time 1.71-approximation [11]. Hence, for every constant $\epsilon > 0$ MRAF on caterpillars has a polynomial-time $(1.71 + \epsilon)$-approximation.

6 Implementation and Experimental Observations

MRAF can be modelled as the *weak chromatic number* of hypergraph: the minimum number of colours assigned to vertices, such that no hyperedge is monochromatic. The set of vertices is X and there is a hyperedge $\{a, b, c, d\}$ whenever the two trees have a different quartet topology on $\{a, b, c, d\}$, leveraging Observation 1. We implemented this as a constraint program (CP) using MiniZinc [22]. For trees with ≤ 30 leaves the CP solves quickly. Code is available at https://github.com/skelk2001/relaxed_agreement_forests. We used this to extend the analysis of [16] on the grass dataset of [12], consisting of fifteen pairs of trees. See Table 1; as expected MRAF grows more slowly than MAF. An FPT algorithm parameterized by MRAF, if it exists, might therefore scale well in practice. FPT algorithms for MAF struggle for MAF ≥ 25 [26]. In fact, MRAF seems more comparable to the *treewidth* of the *display graph* of the input tree pair (obtained by identifying

Table 1. Comparison of MAF and MRAF for the fifteen tree pairs in the data set [12] analysed in [16]. We also include MAST, the lower bound on MRAF given by $\lceil \frac{n}{MAST} \rceil$, and $tw(D)$ which is the treewidth of the display graph obtained from the tree pair.

tree pair	$\|X\| = n$	MAF	MRAF	tw(D)	MAST	$\lceil n/MAST \rceil$
00_rpoC_waxy.txt	10	2	**2**	3	8	2
01_phyB_waxy.txt	14	3	**2**	3	11	2
02_phyB_rbcL.txt	21	5	**3**	3	14	2
03_rbcL_waxy.txt	12	4	**2**	3	9	2
04_phyB_rpoC.txt	21	5	**2**	3	15	2
05_waxy_ITS.txt	15	6	**3**	4	10	2
06_phyB_ITS.txt	30	8	**4**	4	17	2
07_ndhF_waxy.txt	19	5	**3**	4	11	2
08_ndhF_rpoC.txt	34	9	**3**	5	20	2
09_rbcL_rpoC.txt	26	7	**4**	5	14	2
10_ndhF_rbcL.txt	36	7	**4**	3	20	2
11_rbcL_ITS.txt	29	11	**4**	5	17	2
12_ndhF_phyB.txt	40	7	**3**	3	30	2
13_rpoC_ITS.txt	31	11	**4**	6	16	2
14_ndhF_ITS.txt	46	16	**5**	6	20	3

vertices with the same leaf label: the treewidth of this graph is bounded by a function of MAF [16]). We obtained similar results on a more challenging dataset comprising the 163 tree pairs from the dataset in [26] that had at ≤ 50 leaves after pre-processing. See Table 2 in the appendix.

7 Discussion and Open Problems

It remains unclear whether it is NP-hard to compute MRAF on caterpillars, although it seems likely. Can the finite forbidden obstructions that characterize solutions to PIMS be mapped to MRAF on caterpillars and then generalized to general trees? Indeed, how far can MRAF be viewed as a generalization of the PIMS problem to partial orders? Is MRAF on caterpillars FPT? Does it (or PIMS) have a polynomial kernel? What should reduction rules look like, given that rules for MAF seem of limited use (see Appendix A.2)? Strikingly, we do not know whether it is NP-hard to determine whether MRAF ≤ 2 for two general trees, so the FPT landscape is also unclear. How far can the logarithmic approximation for MRAF on general trees, and the 1.71 approximation for MRAF on caterpillars (equivalently, PIMS) be improved? Finally, it will be instructive to elucidate the biological interpretation of this model and to explore MRAF on multiple and/or non-binary trees; such generalisations exist for MAF [7].

References

1. Allen, B., Steel, M.: Subtree transfer operations and their induced metrics on evolutionary trees. Ann. Comb. **5**, 1–15 (2001)
2. Ardevol Martinez, V., Chaplick, S., Kelk, S., Meuwese, R., Mihalák, M., Stamoulis, G.: Relaxed agreement forests. arXiv:2309.01110 [cs.DS] (2023)
3. Atkins, R., McDiarmid, C.: Extremal distances for subtree transfer operations in binary trees. Ann. Comb. **23**, 1–26 (2019)
4. Bar-Yehuda, R., Fogel, S.: Partitioning a sequence into few monotone subsequences. Acta Inform. **35**(5), 421–440 (1998)
5. Björklund, A., Husfeldt, T., Koivisto, M.: Set partitioning via inclusion-exclusion. SIAM J. Comput. **39**(2), 546–563 (2009)
6. Bordewich, M., Semple, C.: On the computational complexity of the rooted subtree prune and regraft distance. Ann. Comb. **8**(4), 409–423 (2005)
7. Bulteau, L., Weller, M.: Parameterized algorithms in bioinformatics: an overview. Algorithms **12**(12), 256 (2019)
8. Chen, J., Shi, F., Wang, J.: Approximating maximum agreement forest on multiple binary trees. Algorithmica **76**, 867–889 (2016)
9. Degnan, J., Rosenberg, N.: Gene tree discordance, phylogenetic inference and the multispecies coalescent. Trends Ecol. Evol. **24**(6), 332–340 (2009)
10. Erdős, P., Szekeres, G.: A combinatorial problem in geometry. Compos. Math. **2**, 463–470 (1935)
11. Fomin, F., Kratsch, D., Novelli, J.C.: Approximating minimum cocolorings. Inf. Process. Lett. **84**(5), 285–290 (2002)
12. Grass Phylogeny Working Group, et al.: Phylogeny and subfamilial classification of the grasses (Poaceae). Ann. Missouri Botanical Garden **88**(3), 373–457 (2001)
13. Heggernes, P., Kratsch, D., Lokshtanov, D., Raman, V., Saurabh, S.: Fixed-parameter algorithms for cochromatic number and disjoint rectangle stabbing via iterative localization. Inf. Comput. **231**, 109–116 (2013)
14. Hein, J., Jiang, T., Wang, L., Zhang, K.: On the complexity of comparing evolutionary trees. Discret. Appl. Math. **71**(1–3), 153–169 (1996)
15. Iersel, L.V., Jones, M., Scornavacca, C.: Improved maximum parsimony models for phylogenetic networks. Syst. Biol. **67**(3), 518–542 (2018)
16. Kelk, S., van Iersel, L., Scornavacca, C., Weller, M.: Phylogenetic incongruence through the lens of monadic second order logic. J. Graph Algorithms Appl. **2**, 189–215 (2016)
17. Kézdy, A., Snevily, H., Wang, C.: Partitioning permutations into increasing and decreasing subsequences. J. Comb. Theory Ser. A **73**(2), 353–359 (1996)
18. Kubicka, E., Kubicki, G., McMorris, F.: On agreement subtrees of two binary trees. Congressus Numerantium 217 (1992)
19. Lemey, P., Salemi, M., Vandamme, A.M.: The Phylogenetic Handbook: A Practical Approach to Phylogenetic Analysis and Hypothesis Testing. Cambridge University Press (2009)
20. Markin, A.: On the extremal maximum agreement subtree problem. Discret. Appl. Math. **285**, 612–620 (2020)
21. Nakhleh, L.: Computational approaches to species phylogeny inference and gene tree reconciliation. Trends Ecol. Evol. **28**(12), 719–728 (2013)
22. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: MiniZinc: towards a standard CP modelling language. In: Bessière, C. (ed.) CP 2007. LNCS, vol. 4741, pp. 529–543. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74970-7_38

23. Steel, M.: Phylogeny: Discrete and Random Processes in Evolution. SIAM (2016)
24. Steel, M., Warnow, T.: Kaikoura tree theorems: computing the maximum agreement subtree. Inf. Process. Lett. **48**(2), 77–82 (1993)
25. Wagner, K.: Monotonic coverings of finite sets. J. Inf. Process. Cybern. **20**(12), 633–639 (1984)
26. van Wersch, R., Kelk, S., Linz, S., Stamoulis, G.: Reflections on kernelizing and computing unrooted agreement forests. Ann. Oper. Res. **309**(1), 425–451 (2022)