Treewidth distance on phylogenetic trees

Theoretical Computer Science 731 (2018) 99–117

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Treewidth distance on phylogenetic trees

Steven Kelk [a], Georgios Stamoulis [a,*], Taoyang Wu [b]

[a] Department of Data Science and Knowledge Engineering (DKE), Maastricht University, Maastricht, the Netherlands
[b] School of Computing Sciences, University of East Anglia, Norwich, United Kingdom

ARTICLE INFO

Article history:
Received 26 June 2017
Received in revised form 5 January 2018
Accepted 5 April 2018
Available online 18 April 2018
Communicated by F.V. Fomin

Keywords:
Graph theory
Phylogenetics
Treewidth
Algorithmic graph theory
Computational biology

ABSTRACT

In this article we study the treewidth of the *display graph*, an auxiliary graph structure obtained from the fusion of phylogenetic (i.e., evolutionary) trees at their leaves. Earlier work has shown that the treewidth of the display graph is bounded if the trees are in some formal sense topologically similar. Here we further expand upon this relationship. We analyze a number of reduction rules, commonly used in the phylogenetics literature to obtain fixed parameter tractable algorithms. In some cases (the *subtree* reduction) the reduction rules behave similarly with respect to treewidth, while others (the *cluster* reduction) behave very differently, and the behavior of the *chain reduction* is particularly intriguing because of its link with graph separators and forbidden minors. We also show that the gap between treewidth and Tree Bisection and Reconnect (TBR) distance can be infinitely large, and that unlike, for example, planar graphs the treewidth of the display graph can be as much as linear in its number of vertices. A number of other auxiliary results are given. We conclude with a discussion and list a number of open problems.

1. Introduction

Phylogenetic trees are used extensively within computational biology to model the history of a set of species (known as taxa) X; the internal nodes represent evolutionary diversification events such as speciation [39]. Within the field of phylogenetics there has long been interest in quantifying the topological dissimilarity of phylogenetic trees and understanding whether this dissimilarity is biologically significant. This has led to the development of many *incongruency measures* such as Subtree Prune and Regraft (SPR) distance and Tree Bisection and Reconnect (TBR) distance [1]. Most of these measures are **NP**-hard to compute and this is indeed true for SPR, TBR distances. More recently such measures have also attracted attention because of their importance in methods which merge dissimilar trees into *phylogenetic networks*; phylogenetic networks are simply the generalization of trees to graphs [31].

Parallel to such developments there has been growing interest in the role of the graph-theoretic parameter *treewidth* within phylogenetics. Treewidth is an intensely studied parameter in algorithmic graph theory and it indicates, at least in an algorithmic sense, how far an undirected graph is from being a tree (see e.g. [7,11,12] for background). The enormous focus on treewidth is closely linked to the fact that a great many **NP**-hard optimization problems become (fixed parameter) tractable on graphs of bounded treewidth [18]. A seminal paper by Bryant and Lagergren [16] linked phylogenetics to treewidth by demonstrating that, if a set of trees (not necessarily all on the same set of taxa X) can simultaneously be topologically embedded within a single "supertree"—a property known as *compatibility*—then an auxiliary graph known as the *display graph* has bounded treewidth. Since this paper a small but growing number of papers at the interface of

* Corresponding author.
E-mail addresses: steven.kelk@maastrichtuniversity.nl (S. Kelk), georgios.stamoulis@maastrichtuniversity.nl (G. Stamoulis), taoyang.wu@uea.ac.uk (T. Wu).

https://doi.org/10.1016/j.tcs.2018.04.004
0304-3975/

graph theory and phylogenetics have explored this relationship further. Much of this literature focuses on the link between compatibility and (restricted) triangulations of the display graph (e.g. [41,29,24,42]), but more recently the algorithmic dimension has also been tentatively explored [5,27,33]. In the spirit of the original Bryant and Lagergren paper, which used heavy meta-theoretic machinery to derive a theoretically efficient algorithm for the compatibility problem, Kelk et al. [34] showed that the treewidth of the display graph of two trees is bounded as a linear function of the TBR distance (equivalently, the size of a Maximum Agreement Forest—MAF [1]) between the two trees, and then used this insight to derive theoretically efficient algorithms for computation of many different incongruency measures. In that article it was empirically observed that in practice the treewidth of the display graph is often much smaller than the TBR distance (and thus also the many incongruency measures for which TBR is a lower bound). This raises two natural questions. First, in how far can this apparently low treewidth be exploited to yield genuinely practical dynamic programming algorithms running over low-width tree decompositions? There has been some progress in this direction in the compatibility literature (notably, [5]) but there is still much work to be done. Second, how *exactly* does the treewidth of the display graph behave, both in the sense of extremal results (e.g. how large can the treewidth of a display graph get?) and in the sense of understanding when and why the treewidth differs significantly from measures such as TBR.

Here we focus primarily on the second question. We begin with a more structural perspective. We show that, given an arbitrary (multi)graph G on n vertices with maximum degree k, one can construct two unrooted binary trees $T_1(G)$ and $T_2(G)$ such that their display graph $D = D(T_1(G), T_2(G))$ has at most $O(nk)$ vertices and edges and G is a minor of D. We combine this with the known fact that cubic expanders (a special family of 3-regular graphs) on n vertices have treewidth $\Omega(n)$ to yield the result that display graphs on n vertices can also (in the worst case) have treewidth linear in n. This contrasts, for example, with planar graphs on n vertices which have treewidth at most $O(\sqrt{n})$ [20]. We also show how a more specialized construction can be used to embed arbitrary grid minors [17] into display graphs with a much smaller inflation in the number of vertices and edges.

We then continue by analyzing how reduction rules often used in the computation of incongruency measures impact upon the treewidth of the display graph. Not entirely surprisingly the *common pendant subtree* reduction rule [1] is shown to preserve treewidth. The *cluster* reduction [4,36,14], however, behaves very differently for treewidth than for many other incongruency measures. Informally speaking, if both trees can be split by deletion of an edge into two subtrees on X' and X'', many incongruency measures combine additively around this *common split*, while treewidth behaves (up to additive terms) like the maximum function. We use this later in the article to explicitly construct a family of tree pairs such that the treewidth of their display graph is 3, but the TBR distance of the trees (and their MP distance—a measure based on the phylogenetic principle of parsimony [25,37,33]) grows to infinity. The third reduction rule we consider is the *chain rule*, which collapses common caterpillar-like regions of the trees into shorter structures. For incongruence measures it is often the case that truncation of such chains to $O(1)$ length preserves the measure [1,15,45], although sometimes the weaker result of truncation to length $f(k)$ [44,43] (for some function that depends only on the incongruency parameter k) is the best known. We show that truncation of common chains to length $f(tw)$, where tw is the treewidth of the display graph, indeed preserves treewidth; this uses asymptotic results on the number of vertices and edges in forbidden minors for treewidth. Proving that truncation to $O(1)$-length preserves treewidth remains elusive; we prove the intermediate result that truncation to length 2 can cause the treewidth to decrease by at most 1. The case when the chain is not a separator of the display graph seems to be a particularly challenging bottleneck in removing the "-1" term from this result. Although intuitively reasonable, it remains unclear whether truncation to length $O(1)$ is treewidth-preserving, for some universal constant.

In the last two mathematical sections of the paper we prove that, if two trees have TBR- or MP-distance 1, then the treewidth of their display graph is 3. However, the converse certainly does not hold: we construct the aforementioned "infinite gap" examples where the display graph has treewidth 3 but both TBR distance and MP-distance spiral off to infinity.

Finally, we reflect on the wider context of these results and discuss a number of open problems.

In conclusion, we observe that for (algorithmic) graph theorists the interface between treewidth and phylogenetics continues to yield many new questions which will likely require a new "phylo-algorithmic" graph theory to be answered. For phylogeneticists the appeal remains structural-algorithmic: can we convert the apparently low treewidth of display graphs into competitive, or even superior, algorithms for computation of incongruency measures?

2. Preliminaries

An *unrooted binary phylogenetic tree* T on a set of leaf labels (known as *taxa*) X is an undirected tree where all internal vertices have degree three and the leaves are bijectively labeled by X. If we (exceptionally) allow some internal vertices of T to have degree two, then we call these vertices *roots* (abusing slightly the usual root meaning). When it is understood from the context we will often drop the prefix "unrooted binary phylogenetic" for brevity.

Let $Y \subseteq X$. Then, for a tree T on X we denote by $T|Y$ the tree which is obtained by forming a minimal subgraph T' of T that spans all leaves labeled by Y, and suppressing any vertices of degree 2.

In this manuscript the *display graph* of two binary phylogenetic trees plays a central role (Fig. 1):

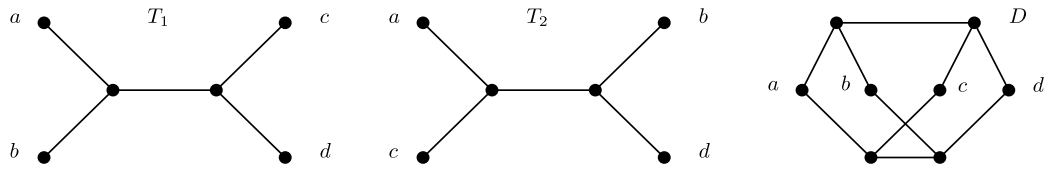

Fig. 1. An example of the display graph D of two binary phylogenetic trees T_1, T_2. As we will see later, we can safely suppress all degree two vertices without altering the treewidth of D. Observe that by doing that, the resulting display graph D is isomorphic to K_4 and thus has treewidth 3.

Definition 2.1. Let $T_1 = (V_1 \cup X, E_1), T_2 = (V_2 \cup X, E_2)$ be two trees, both on the same set of leaf labels X. The *display graph* of T_1, T_2, denoted by $D(T_1, T_2)$, is formed by identifying vertices with the same leaf label and forming the disjoint union of these two trees, i.e., $D(T_1, T_2) = (V_1 \cup V_2 \cup X, E_1 \cup E_2)$.

This definition can be extended in a straightforward way to more than 2 trees. We remark that in [16], the definition of the display graph of two (or more) trees does not necessarily insist that the sets of taxa of the trees are identical. Here we will focus on the case where the two trees are defined on exactly the same set of taxa X.

For two phylogenetic trees T_1, T_2 we say that T_1 *displays* T_2 if the latter can be obtained by contracting edges in an induced subtree of the former. We say that two (or more) trees are *compatible* if there exists another tree on X that displays all the trees. Note that for two unrooted binary phylogenetic trees on the same set of labels X compatibility is simply equivalent to the existence of a label-preserving isomorphism between the two trees.

A *tree decomposition* of an undirected graph $G = (V, E)$ is a pair $(\mathcal{B}, \mathbb{T})$ where $\mathcal{B} = \{B_1, \ldots, B_q\}$, $B_i \subseteq V(G)$, is a multiset of *bags* and $\mathbb{T}$ is a tree whose q nodes are in bijection with $\mathcal{B}$, satisfying the following three properties:

(tw1) $\cup_{i=1}^{q} B_i = V(G)$;
(tw2) $\forall e = \{u, v\} \in E(G), \exists B_i \in \mathcal{B}$ s.t. $\{u, v\} \subseteq B_i$;
(tw3) $\forall v \in V(G)$ all the bags B_i that contain v form a connected subtree of $\mathbb{T}$.

The *width* of $(\mathcal{B}, \mathbb{T})$ is equal to $\max_{i=1}^{q} |B_i| - 1$. The *treewidth* of G is the smallest width among all possible tree decompositions of G. For a graph G, we denote $tw(G)$ the treewidth of G. Without ambiguity we will often simply use $\mathbb{T}$ to refer to tree decompositions, rather than the more formal $(\mathcal{B}, \mathbb{T})$ notation. Given a tree decomposition $\mathbb{T}$ for some graph G, we denote by $V(\mathbb{T})$ the (multi) set of its bags and by $E(\mathbb{T})$ the set of its edges (connecting bags). Property (tw3) is also known as *running intersection property*. We note that the treewidth of any graph G is at most $|V(G)| - 1$: consider a bag with all vertices of G. This is a valid tree decomposition of width $|V(G)| - 1$. Thus the treewidth is always a finite parameter for any finite graph. We call a tree decomposition *optimal* if it has minimum possible width.

Another, equivalent, definition of treewidth is based on chordal graphs. We remind that a graph G is chordal if every induced cycle in G has exactly three vertices. The treewidth of G is the minimum, ranging over *all* chordal completions $c(G)$ of G (we add edges until G becomes a chordal graph), of the size of the maximum clique in $c(G)$ minus one. Under this definition, each bag of a tree decomposition of G naturally corresponds to a maximal clique in a chordal completion of G [6].

For a graph $G = (V, E)$ and an edge $e = \{u, v\} \in E(G)$, the *deletion* of e is the operation which simply deletes e from $E(G)$ and leaves the rest of the graph G the same. The *contraction* of e, denoted G/e, is the operation where edge e is deleted and its incident vertices u, v are identified. We say that a graph H is a *minor* of another graph G if H can be obtained by repeated applications of edge deletions and/or edge contraction, followed possibly by deleting isolated vertices, on G.[1] The order that these operations are performed does not matter and it will always result in H.

2.1. Phylogenetic distances and measures

Several distances have been proposed to measure the incongruence between (i.e., the dissimilarity of) two or more phylogenetic trees on the same set of taxa. The most relevant distances for the purpose of this article are the so-called *Tree Bisection and Reconnect* distance and the *Maximum Parsimony Distance* which are defined in the following.

Given an unrooted binary phylogenetic tree T on X, a *Tree Bisection and Reconnect* (TBR) move is defined as follows [1]: (1) we delete an edge of T to obtain two subtrees T' and T''. (2) Then we select two edges $e_1 \in T', e_2 \in T''$, subdivide them with two new vertices v_1 and v_2 respectively, add an edge from v_1 to v_2, and suppress all vertices of degree 2. In case either T' or T'' is a single leaf, then the new edge connecting T' and T'' is incident to that leaf. Let T_1, T_2 be two unrooted binary phylogenetic trees on the same set of leaf-labels. The TBR-distance from T_1 to T_2, denoted $d_{TBR}(T_1, T_2)$, is the *minimum* number of TBR moves required to transform T_1 into T_2 (or, equivalently, T_2 to T_1).

[1] Equivalently we can say that H is a minor of G if H can be obtained by vertex deletions, edge deletions and edge contractions in G.

Computing the TBR-distance is essentially equivalent to the *Maximum Agreement Forest (MAF)* problem: Given an unrooted binary phylogenetic tree on X and $X' \subset X$ we let $T(X')$ denote the minimal subtree that connects all the elements in X'.[2] An *agreement forest* of two unrooted binary trees T_1, T_2 on X is a partition of X into non-empty blocks $\{X_1, \ldots, X_k\}$ such that (1) for each $i \neq j$, $T_1(X_i)$ and $T_1(X_j)$ are node-disjoint and $T_2(X_i)$ and $T_2(X_j)$ are node-disjoint, (2) for each i, $T_1|X_i = T_2|X_i$. A *maximum agreement forest* is an agreement forest with a minimum number of components (such that it *maximizes* the agreement), and this minimum is denoted $d_{MAF}(T_1, T_2)$. In 2001 it was proven by Allen and Steel [1] that $d_{MAF}(T_1, T_2) = d_{TBR}(T_1, T_2) + 1$.

In order to define the Maximum Parsimony Distance [25,37,33] between two unrooted binary phylogenetic trees T_1, T_2 both on X, we need first to define the concept of *character* on X which is simply a surjection $f : X \to \mathbf{C}$ where $\mathbf{C}$ is a set of *states*. Given a tree T on X, and a character f also on X, an *extension* of f to T is a mapping f' from $V(T)$ to $\mathbf{C}$ such that $f'(\ell) = f(\ell)$, $\forall \ell \in X$. An edge $e = \{u, v\}$ with $f'(u) \neq f'(v)$ is known as a *mutation* induced by f'. The minimum number of mutations ranging over all extensions f' of f is called the *parsimony score* of f on T and is denoted by $l_f(T)$. Given two trees T_1, T_2 their *maximum parsimony distance* $d_{MP}(T_1, T_2)$ is equal to $\max_f |l_f(T_1) - l_f(T_2)|$.

Both the TBR and MP distances are **NP**-hard to compute [1,32] and they are also *metric* distances i.e., they satisfy the four axioms of metric spaces: (a) non-negativity, (b) identity of indiscernibles (c) symmetry and (d) triangle inequality [1,25].

Given an unrooted binary phylogenetic tree T and a distance d (such as TBR or MP), we define the *unit ball* or the *unit neighborhood* of T under d to be $u_d(T) = \{T' : d(T, T') = 1\}$ i.e., the set of all trees T' that are within distance one from T under the distance d. Such neighborhoods are important because usually they are building blocks of "local search" algorithms that try to find trees that optimize some particular criterion. The diameter $\Delta_n(d)$ is defined as the maximum value d taken over all pairs of phylogenetic trees with n taxa (see [40, Section 2.5] for a recent review on various results on the unit ball and the diameter of several tree rearrangement metrics).

3. Treewidth distance

The main purpose of this manuscript is to define and study the properties of the *treewidth distance* between two phylogenetic trees. As mentioned in the introduction, the study of treewidth in the context of phylogenetics was triggered by the pioneering work of Bryant & Lagergren [16] who proved that a necessary condition for a set of trees (not necessarily on the same set of taxa) to be compatible, is that their display graph has bounded treewidth. They used this insight to leverage a (theoretical) positive algorithmic result. Here we are interested in the question: in how far does the treewidth of the display graph *itself* function directly as a measure of phylogenetic incongruence? Hence the following natural definition:

Definition 3.1 *(Treewidth distance).* Given two unrooted binary phylogenetic trees T_1, T_2, both on the same set of leaf labels X, where $|X| \geq 3$, their treewidth distance is defined to be $tw(D(T_1, T_2)) - 2$ and is denoted as $d_{tw}(T_1, T_2)$.

It is easy to see that for two unrooted binary phylogenetic trees T_1, T_2 we have that $d_{tw}(T_1, T_2) \geq 0$, for $|X| \geq 3$. This is a direct consequence of the fact that if $|X| \geq 3$ then the display graph contains at least one cycle and hence $tw(D(T_1, T_2)) \geq 2$. If $|X| < 3$ then T_1, T_2 are trivially isomorphic (they are either a single edge or a single vertex) and it does not make much sense to define a distance between such trees. So we can discard these boundary cases without any loss of generality in our study. (Of course, the treewidth of the display graph is still well-defined in these omitted boundary cases.) On the other hand we will leverage the well-known fact that $tw(D(T_1, T_2)) = 2$ for two unrooted binary phylogenetic trees on X, $|X| \geq 3$, if and only if T_1 and T_2 are compatible (see e.g. [27]). As mentioned earlier, compatibility in this context is the same as label-preserving isomorphism, so it is natural to speak of equality and write $T_1 = T_2$. Note that it was shown in [34] that $tw(D(T_1, T_2)) \leq d_{MAF}(T_1, T_2) + 1 = d_{TBR}(T_1, T_2) + 2$, and hence $d_{tw}(T_1, T_2) \leq d_{TBR}(T_1, T_2)$.

We remark that, because computation of treewidth is fixed parameter tractable [8,22], so too is d_{tw}. As we discuss in the final section of the paper it is not known whether d_{tw} can be computed in polynomial time, but ongoing research efforts by the algorithmic graph theory community to compute treewidth efficiently in practice (see e.g. [10,19]) will naturally strengthen the appeal of d_{tw} as a phylogenetic measure.

A rather easy but important observation, whose proof we include here for completeness, and that we will use extensively in the rest of the manuscript is that treewidth (and hence treewidth distance) remains unchanged by edge subdivision and degree-2 vertex suppression operations—with one trivial exception. We say that a graph is a *unique triangle* graph if it contains exactly one cycle such that this cycle has length 3 and at least one of the cycle vertices has degree 2. A unique triangle graph has treewidth 2.

Given a graph $G = (V, E)$, let $e = \{u_1, u_2\} \in E$ be any edge of G and v be any degree-2 vertex of G (if any) with neighbors v_1, v_2. We define the following two operations:

Subdivision of an edge e: This defines a new graph $G' = (V', E')$ where $V' = V \cup \{w\}, w \notin V$ and $E' = (E \setminus \{e\}) \cup (\{u_1, w\}, \{w, u_2\})$.

[2] Note that in $T(X')$, unlike $T|X'$, we do *not* suppress vertices of degree 2.

Suppression of a degree-2 vertex v**:** This defines a new graph $G'' = (V'', E'')$ where $V'' = V \setminus \{v\}$, $E'' = E \setminus (\{v_1, v\} \cup \{v, v_2\}) \cup \{v_1, v_2\}$.

Observation 3.1. *Let $G = (V, E)$ be a graph, which is not a unique triangle graph and let $e = \{u_1, u_2\} \in E$ be any edge of G and v be any degree-2 vertex of G (if any) with neighbors v_1, v_2. Consider the following two graphs:*

1. *$G' = (V', E')$ where we obtain G' after a single application of the edge subdivision step on edge $e \in E(G)$, and*
2. *$G'' = (V'', E'')$ where G'' is obtained from G after suppressing a degree-2 vertex $v \in V(G)$.*

Then we have that:

$$tw(G) = tw(G') = tw(G'').$$

Proof. For the subdivision of an edge case, let G' be the resulting graph after the subdivision of some edge e. It is immediate that the treewidth of G' is at least $q = tw(G)$ since G is a minor of G' and treewidth is non-increasing under minor operations. To show that the treewidth cannot increase we argue as follows. If G is a tree, then G' is also a tree and $tw(G) = tw(G') = 1$ and we are done. So, we assume that G is not a tree so $q \geq 2$. Take a bag B of an optimal tree decomposition $\mathbb{T}$ of G with largest bag size at least 3, that contains the endpoints u_1, u_2 of e. Create a new bag $B' \notin V(\mathbb{T}) : B' = \{u_1, u_2, w\}$ and attach it to B. This operation cannot increase the treewidth of the tree decomposition and it is immediate that the new tree decomposition is a valid one for G'.

Now we will handle the degree-2 vertex suppression operation. This can be simulated by two edge contraction operations, which are minor operations, so the treewidth cannot increase. In the other direction (i.e. proving that the treewidth cannot decrease), we see that if G is a tree the treewidth is immediately preserved. If G is not a tree, let G'' be the resulting graph after a single degree-2 vertex suppression operation on a vertex v with neighbors, in G, v_1, v_2 such that in G'' we have that $\{v_1, v_2\} \in E''$. Take an optimal tree decomposition of G'', let this be $\mathbb{T}''$. By assumption that G is not a tree and that G is not a unique triangle graph, G'' contains at least one cycle. Hence, $tw(G'') \geq 2$ i.e., the size of the largest bag is at least 3. In $\mathbb{T}''$, locate a bag A that contains the pair of vertices v_1, v_2. Such a bag must exist by definition. Create a new bag $A' = \{v_1, v, v_2\}$ and attach it to A thus creating a new tree decomposition $\mathbb{T}'''$. It is immediate that $\mathbb{T}'''$ is a valid tree decomposition for G with width the same as the width of $\mathbb{T}''$, and the claim follows. □

Recall that if two unrooted binary trees T_1, T_2 are incompatible, then $tw(D(T_1, T_2)) \geq 3$, so the display graph cannot be a unique triangle graph (which has treewidth 2). A single suppression or subdivision operation is therefore (by Observation 3.1) treewidth-preserving, meaning that repeated applications of these operations cannot cause the unique triangle graph to arise, and hence they are also treewidth-preserving. Summarizing,

Observation 3.2. *Let T_1 and T_2 be two unrooted binary phylogenetic trees on the same set of taxa X. If T_1 and T_2 are incompatible, then the following operations can be applied arbitrarily to $D(T_1, T_2)$ without altering its treewidth: suppression of degree-2 vertices, and subdivision of edges.*

In subsequent sections we will often use Observation 3.2 to (in particular) suppress some or all of the taxa in the display graph without altering its treewidth.

3.1. Metric properties of d_{tw}

Given the definition of the treewidth distance, it is tempting to see if indeed such a distance is a metric distance e.g., it satisfies the four axioms of metric distances. We already argued that it satisfies the non-negativity condition and trivially it satisfies the identity of indiscernibles because $T_1 = T_2 \Leftrightarrow d_{tw}(T_1, T_2) = 0$ as demonstrated in the previous discussion. The symmetry condition is also trivially satisfied because $D(T_1, T_2) = D(T_2, T_1)$ i.e., the display graph is identical in both cases and thus has the same treewidth.

The only case left is to see if d_{tw} satisfies the triangle inequality property: given three unrooted binary phylogenetic trees T_1, T_2, T_3 all on X is it the case that $d_{tw}(T_1, T_3) \leq d_{tw}(T_1, T_2) + d_{tw}(T_2, T_3)$? Unfortunately, this is false as shown in Fig. 2. By using appropriate software, for example QuickBB [26], we can see that $d_{tw}(T_1, T_2) = 1, d_{tw}(T_2, T_3) = 2$ and $d_{tw}(T_1, T_3) = 4 > d_{tw}(T_1, T_2) + d_{tw}(T_2, T_3)$. We remark that, although mathematically disappointing, the absence of the triangle inequality is not a great hindrance in practice. Some other well-known phylogenetic measures, such as hybridization number, also do not obey the triangle inequality [38].

4. Diameters on d_{tw}

In this section we explore the question of how large the treewidth of the display graph of two unrooted binary phylogenetic trees, both on X, can get. More precisely, we consider the diameter $\Delta_n(d_{tw})$ defined as the maximum value d_{tw} taken

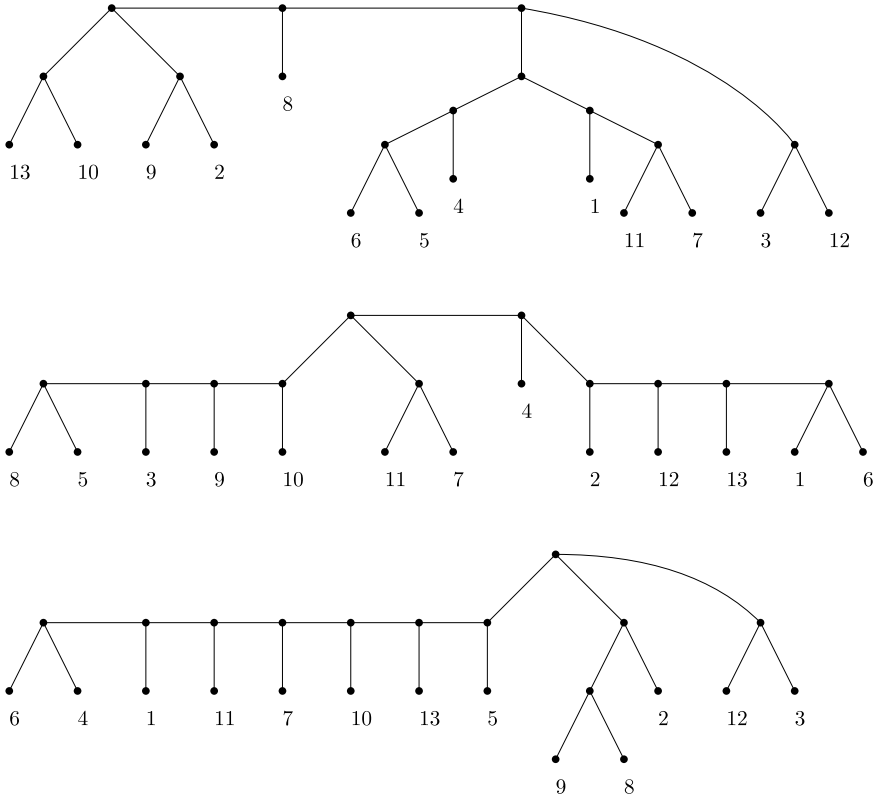

Fig. 2. An example of three trees (from top to bottom: T_1, T_2 and T_3) on a common set of taxa for which the triangle inequality is violated.

over all pairs of phylogenetic trees with n taxa. Somewhat surprisingly, we show that $\Delta_n(d_{tw})$ is bounded below and above by linear functions on n. To prove this, we first present a general result showing how we can embed an arbitrary graph into display graphs (as minor) without adding too many extra edges or vertices.

Theorem 4.1. *Let $G = (V, E)$ be an undirected (multi)graph with n vertices and maximum degree $d \geq 2$. Then we can construct two unrooted binary phylogenetic trees T_1 and T_2 such that both trees have $O(nd)$ taxa, $O(nd)$ nodes and $O(nd)$ edges (and hence their display graph has $O(nd)$ nodes and edges) and G is a minor of $D(T_1, T_2)$.*

Proof. We start by selecting an arbitrary unrooted binary tree T on $n+2$ taxa. Set $T_1 := T$ and $T_2 := T$. The idea is that the n internal nodes of T_1 are in bijection with the n vertices of G. We will add the edges of G one at a time, in the following manner. If an edge $e = \{u, v\}$ of G already exists within T_1, the edge is already encoded so there is nothing to do. If not, we subdivide an arbitrary edge in T_2 and let y be the subdivision node. We then introduce two new taxa x_1^e and x_2^e and a new vertex z in T_2, and add the following edges: $\{u, x_1^e\}$, $\{x_1^e, z\}$, $\{z, y\}$, $\{z, x_2^e\}$ and $\{x_2^e, v\}$. The first and last of these edges is in T_1, the rest are in T_2. In the display graph the path u, x_1^e, z, x_2^e, v will become the image of the edge $\{u, v\}$ (in the embedding of the minor). After encoding all the edges, T_1 and T_2 will each have at most $k = (n+2) + 2|E|$ taxa, so (because T_2 remains binary) each will have at most $k-2$ internal nodes and each at most $2k-3$ edges. Now, observe that the n internal nodes of T_1 might have degree as large as $d+3$. To turn T_1 into a binary tree we replace each vertex u, where $deg(u) > 3$, by a path of $t = deg(u) - 2$ vertices $u_1, ..., u_t$. The first two edges incident to u are now made incident to u_1, the final two edges incident to u are made incident to u_t, and each of the remaining edges is made incident to exactly one of the nodes $u_2, ..., u_{t-1}$. (When obtaining u from the embedding of G, the idea is that the edges of the path will be contracted to retrieve u.) This transformation does not alter the number of taxa, so T_1 and T_2 now have both the same number of internal nodes and edges (i.e. at most $k-2$ and $2k-3$ respectively). Due to the fact that G has maximum degree d, $|E| \leq nd/2$. We conclude that both trees each has at most $(n+2) + nd$ taxa, at most $n(d+1)$ internal nodes and at most $2n + 4 + 4|E| - 3 \leq 2n + 1 + 2nd$ edges. It follows that $D(T_1, T_2)$ has at most $2n(d+1) + ((n+2) + nd)$ nodes in total and at most $4n + 2 + 4nd$ edges. □

We note that the above construction can be easily computed in polynomial time.

Applying the last theorem to complete graphs leads to a lower bound of $\sqrt{n}$ on $\Delta_n(d_{tw})$. However, we can get a better lower bound by using the well known fact that there are cubic expanders on q vertices with treewidth at least ϵq, for some

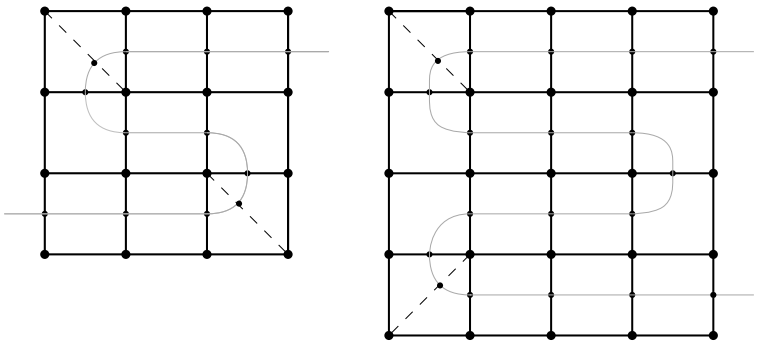

Fig. 3. Embedding grid minors in the display graphs of two unrooted binary trees: grids with even side length (left; $k = 4$) and odd side length (right; $k = 5$). Taxa are shown as small dots inside the grid. Both trees have exactly $(k-1)^2 + 3$ taxa.

constant $\epsilon > 0$ [28,23]. By Theorem 4.1 and its proof, for such a cubic expander graph G with q vertices, there exist two trees T and T' with precisely $4q + 2$ taxa such that G is a minor of $D(T, T')$. If the construction of Theorem 4.1 results in two trees with less than $4q + 2$ taxa, then we can always use, without any loss, the reverse of cherry reduction[3] to "inflate" them to $4q + 2$. For any positive integer q, let G_q be a cubic expander on q vertices. Now, for each n, consider a cubic expander G_q with $q = (n-2)/4$ (or its nearest integer) vertices, and let T, T' be the two trees constructed. Then we have $tw(D(T, T')) \geq tw(G_q) \geq \epsilon(n-2)/4 \geq \epsilon' n$. The upper bound follows from $\Delta_n(d_{tw}) \leq \Delta_n(d_{TBR}) \leq n - 3 - \left\lfloor \frac{\sqrt{n-2}-1}{2} \right\rfloor$, where the second inequality follows from [21, Theorem 1.1].

Corollary 4.1. *We have $\Delta_n(d_{tw}) = \Theta(n)$ as $n \to \infty$. More precisely, there exists a constant $\epsilon > 0$ such that $\epsilon n < \Delta_n(d_{tw}) < n - 3$ for all $n \geq 4$.*

The construction (and bounds) described in Theorem 4.1 can be refined significantly in specific cases. Consider the $k \times k$ grid graph, which has maximum degree 4 and k^2 nodes. When taking $n = k^2$ the theorem yields a bound of $\approx 13n$ nodes. However, consider the construction shown in Fig. 3, which distinguishes the cases k even and k odd. The two sides of the curve indicate the two trees that are needed and the points at which the curve touches the grid become the taxa of the two trees. (Note that, without the dashed edges, we would be forced to model the corresponding corners of the grid with degree-2 nodes in the phylogenetic trees, and phylogenetic trees do not usually contain degree-2 nodes. This minor technicality only affects two of the four corners of the grid.[4])

As in the theorem the degree-4 nodes can be split into two degree-3 nodes. In both the odd and even cases it can be verified that both the resulting unrooted binary trees have $(k-1)^2 + 3$ taxa and thus that the display graph has $3(k-1)^2 + 5$ nodes in total. This is $\approx 3n$, a significant improvement on the generic bound. In fact it is not far from "best possible". A $k \times k$ grid contains $(k-1)^2$ chordless 4-cycles, and because a tree cannot contain a cycle the embedding of each cycle must pass through at least 2 taxa in the display graph. Each taxon can be shared by at most two 4-cycles (because the display graph has maximum degree 3) yielding a lower bound of $(k-1)^2$ on the number of taxa required.

5. The treewidth of the display graph under phylogenetic reduction rules

In this section we investigate the effect of several common phylogenetic reduction rules on the treewidth of the display graph. We will study the following three rules: (i) common pendant subtree, (ii) common chain and (iii) cluster reduction rule. Such rules constitute the building block of many FPT algorithms for computing phylogenetic distances. We will see that the three reduction rules behave somewhat differently with respect to the treewidth of the display graph. In particular, we will show how the subtree reduction operation, where compatible subtrees are collapsed to a single taxon, preserves the treewidth of the display graph. For the second case, the collapsing of a common chain (a maximal "caterpillar-like" region) in both trees down to length 2, could potentially decrease the treewidth of the display graph by *at most* one. On the other hand we show that if we collapse common chains down to length that is a function of the treewidth of the display graph, then we preserve the treewidth. The open question here is if this gap can be understood better i.e., if we can collapse the common chains to a constant length and preserve the treewidth. Finally, we investigate the cluster reduction rule where clusters are formed if in each tree there is an edge (called a *common split*) such that deleting this edge causes both trees to be split into two subtrees on X' and X''. We will see that the treewidth of the display graph is (up to additive terms)

[3] See next section: intuitively, a *cherry* reduction contracts a common *cherry* (two leaves with a common parent) to a single vertex. The inverse operation simply replaces a leaf with a cherry on two leaves with new labels. As we will see, neither of these operations alter the treewidth of the graph.

[4] Note that if we "round off" the 4 corners of the grid its treewidth (which is k) is unaffected and the dashed edges are not required.

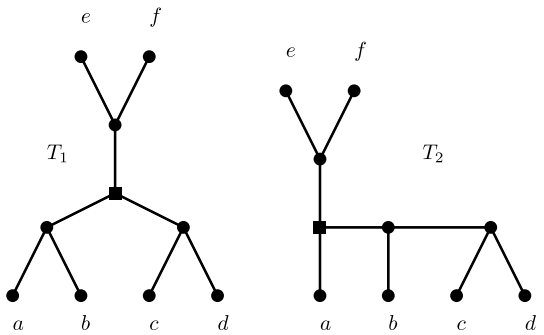

Fig. 4. An illustration of the concept of *common pendant subtree*. If $X' = \{a, b, c, d\}$ we see that $T_1|X' = T_2|X'$ (because of the suppression of the parent of a in T_2) but this is not true if we take into account the "root" location (in bold squares), so $\{a, b, c, d\}$ does not induce a common pendant subtree. Maximal common pendant subtrees are induced by $\{e, f\}$ and $\{c, d\}$.

equal to the *maximum* of the treewidth of the two clusters. We note that this is in contrast to other phylogenetic distance measures which usually behave *additively* with respect to the distances of the two clusters.

It is well known that compatibility is preserved under the described reductions. For this reason we will assume that the two input trees T_1, T_2 on X are *not* compatible. This immediately gives us a lower bound on the cardinality of the taxon set, namely $|X| \geq 3$ since any two trees on 2 taxa are by definition compatible (both trees are single edges). Moreover the treewidth of their display graph is at least 3.

We start with the common pendant subtree rule.

5.1. Subtree reduction rule

Let T_1, T_2 be two unrooted binary phylogenetic trees on the same set of taxa X. A subtree T is called a *pendant* subtree of T_i, $i \in \{1, 2\}$ if there exists an edge e the deletion of which detaches T from T_i. A subtree T, which induces a subset of taxa $X' \subset X$, is called *common pendant subtree* of T_1 and T_2 if $T_1|X' = T_2|X'$ and if the additional following condition holds:

- ▷ Let e_i be the edge of tree $T_i, i \in \{1, 2\}$ the deletion of which detaches T from T_i and let $v_i \in e_i, i \in \{1, 2\}$ be the endpoint of e_i "closest" to the taxon set X'. Let's say that we root each $T_i|X'$ at v_i, thus inducing a *rooted* binary phylogenetic tree $(T_i|X')^\rho$ on X'. We require that $(T_1|X')^\rho = (T_2|X')^\rho$.

The previous condition formalizes the idea that the point of contact of the pendant subtree with the rest of the tree should explicitly be taken into account when determining whether a pendant subtree is common. (This is consistent with the definition of common pendant subtree elsewhere in the literature.)

In the following we will show that the treewidth of the display graph $D(T_1, T_2)$ of the two phylogenetic trees T_1, T_2 is preserved under the common pendant subtree reduction rule:

Common Pendant Subtree (CPS) reduction: For an example of the concept of the common pendant subtree see Fig. 4. Find a maximal common pendant subtree in T_1, T_2. Let T be such a common subtree with at least two taxa and let X_T be its set of taxa. Clip T from T_1 and T_2. Attach a single label $x \notin X$ in place of T on each T_i. Set $X := (X \setminus X_T) \cup \{x\}$ and let T_1', T_2' be the two resulting trees and $D(T_1', T_2') = D'$ be their resulting display graph.

Theorem 5.1. *Suppose that T_1 and T_2 are a pair of incompatible unrooted binary phylogenetic trees on X and the pair (T_1', T_2') is obtained from (T_1, T_2) by one application of the Common Pendant Subtree reduction. Then $d_{tw}(T_1, T_2) = d_{tw}(T_1', T_2')$.*

Proof. A *cherry* is simply a size-2 subset of taxa $\{x, y\}$ that have a common parent, and a cherry $\{x, y\}$ is *common* if it is in both trees.

Let us first consider the case that the pair (T_1', T_2') is obtained from a subtree reduction on a common cherry $\{x, y\}$ whose parent is u_i in T_i and the parent of u_i is v_i, $i = 1, 2$. Then the display graph $D' = D(T_1', T_2')$ is obtained from $D = D(T_1, T_2)$ by replacing the vertex subset $\{u_1, x, y, u_2\}$ with a single vertex r which is connected to v_1 and v_2 and these are the only neighbors of r (see Fig. 5). Note that $v_1 \neq v_2$; $v_1 = v_2$ could only happen if $|X| = 3$, but then the trees would be compatible, contradicting the assumption of incompatibility. So $|X| \geq 4$ and $v_1 \neq v_2$ are internal nodes of T_1 and T_2 respectively. Display graphs do not contain edges between internal nodes of different trees, so $\{v_1, v_2\}$ is not an edge in D. D' can be obtained from D by applying Observation 3.2: suppress x, suppress y (and delete the created multi-edge) and then suppress u_2. Hence $tw(D') = tw(D)$. (The surviving vertex u_1 assumes the role of r, since labels are irrelevant to treewidth.)

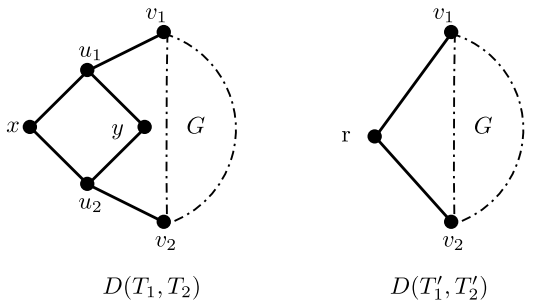

Fig. 5. Reduction of a common cherry $\{x, y\}$ as described in the proof of Theorem 5.1.

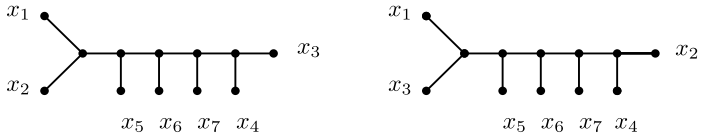

Fig. 6. An example of two trees with a common chain, which is indexed by taxa x_5, x_6, x_7, x_4.

For the more general case: it is easy to see that applying the CPS reduction rule to a subtree that is not a cherry, can be achieved by iteratively applying the CPS reduction to common cherries. This is correct because collapsing a common cherry cannot make two incompatible trees compatible. The result follows. □

Note that in the proof of Theorem 5.1 incompatibility is only used to force $|X| \geq 4$. Common cherries can also be collapsed in compatible trees, without altering the treewidth, as long as $|X| \geq 4$. Collapsing a common cherry in two compatible trees with $|X| = 3$, however, either creates a multigraph or, if multigraphs are not permitted, causes the treewidth of the display graph to decrease (to 1). To avoid such uninteresting boundary technicalities we have focused only on incompatible pairs of trees.

5.2. Chain reduction rule

Let T be an unrooted binary tree on X. For each taxon $x_i \in X$, let p_i be its unique parent in T. Let $C = (x_1, x_2, \ldots, x_t)$ be an ordered sequence of taxa and let $P = (p_1, p_2, \ldots, p_t)$ be the corresponding ordered sequence of their parents, if P is a *path* in T and the p_i are all mutually distinct then C is called a *chain* of length t. A chain C is a *common chain* of two binary phylogenetic trees T_1, T_2 on a common set of taxa, if C is a chain in each one of them. See Fig. 6 for an example. Note that our insistence that the p_i are mutually distinct differs slightly from the definition of chain encountered elsewhere in the literature (see e.g., [33]), in which $p_1 = p_2$ and $p_{t-1} = p_t$ is permitted. However, our more restrictive definition of chain is only a very mild restriction, since a chain of length t under the traditional definition yields a chain of length at least $(t-2)$ under our definition. Our definition ensures that in both trees neither end of the chain is a cherry, which avoids a number of annoying (and uninteresting) technicalities. Let v_i denote the parent of x_i in T_1 and u_i its parent in T_2.

We now define the common chain reduction rule.

Common d-Chain Reduction Rule (d-cc): Let T_1, T_2 be two incompatible unrooted binary phylogenetic trees on a common set of taxa X. Let C be a common chain of T_1, T_2 of length $t \geq 3$. On each $T_i, i \in \{1, 2\}$ clip the chain down to length $d \in \{2, \ldots, t-1\}$ as follows: Keep the first $\lceil d/2 \rceil$ and the last $\lfloor d/2 \rfloor$ taxa and delete all the intermediate ones (i.e., delete all the taxa with indexes in $\{\lceil d/2 \rceil + 1, \ldots, t - \lfloor d/2 \rfloor\}$) and suppress any resulting vertices of degree 2. Let C' be the new clipped common chain on both trees.

Observe that C' has $\lceil d/2 \rceil + \lfloor d/2 \rfloor = d$ taxa and that in each T_1, T_2 the parents of the taxa $x_{\lceil d/2 \rceil}, x_{t-\lfloor d/2 \rfloor+1}$ are connected by an edge. Let $D(T_1, T_2) = D$ be the display graph of T_1, T_2 and $D(T_1', T_2') = D'$ be the display graph of T_1', T_2' after the application of one chain reduction rule. Equivalently, D' can be obtained directly from D by deleting the $(t-d)$ pruned taxa and suppressing unlabeled degree-2 vertices.

In the following we will need to argue that the common chain reduction rule preserves incompatibility. We use the notation $ab|cd$ to denote the *quartet* (unrooted binary phylogenetic tree on four leaves) in which taxa a and b are on one side of the single internal edge but c and d are on the other.

Proposition 5.1. *Let T_1', T_2' be two unrooted binary phylogenetic trees that are obtained after a single application of the operation d-cc(C) $(d \geq 2)$ on two phylogenetic trees T_1 and T_2 that have a common chain C. If T_1, T_2 are incompatible phylogenetic trees, then so are T_1', T_2'.*

Proof. It is sufficient to prove the claim for $d = 2$, since adding taxa to incompatible trees (i.e. clipping fewer taxa from the common chain) cannot make them compatible. Hence, only the chain taxa $\{x_1, x_t\}$ survive in T_1', T_2'. The fact that T_1, T_2 are incompatible means that we can find four taxa $a, b, c, d \in X$ such that T_1 displays $ab|cd$ but T_2 displays $ac|bd$ (i.e., $ab|cd, ac|bd$ are incompatible quartets [39], Corollary 6.3.10). If $\{a, b, c, d\} \cap C = \emptyset$ then the incompatible quartets survive in T_1', T_2' and we are done. Note that it is not possible that all four taxa belong to C because the two quartets would then be identical (i.e. compatible). We distinguish therefore the cases when 1,2 or 3 of the taxa are in C. If exactly 1 of the four taxa is in C, then replacing (if it is not already equal to x_1) this taxon with x_1 produces two incompatible quartets in T_1', T_2' (which are isomorphic to the original two incompatible quartets) and we are done. If exactly 2 are in C, then replacing (where necessary) these with x_1 and x_t again yields isomorphic incompatible quartets in T_1', T_2'. Finally, assume that C contains three of the taxa from $\{a, b, c, d\}$ and let us assume, without any loss, that these are a, b, d and that they occur along the chain in this ascending order. (The case when they are in descending order is symmetrical.) We know, due to incompatibility of the two original quartets, that non-chain taxon c is on the x_t side of the chain in T_1 but on the x_1 side of the chain in T_2. Notice that because of our definition of a common chain, there *must* exist a non-chain taxon z_1 on the x_1 side of the chain in T_1 and a non-chain taxon z_2 (with possibly $z_2 = z_1 \neq c$) on the x_t side of the chain in T_2. If z_1 is also on the x_t side of the chain in T_2 then observe that $z_1x_1|x_tc$ in T_1' and $cx_1|x_tz_1$ in T_2' form incompatible quartets. Otherwise (i.e. z_1 is on the x_1 side of the chain in T_2) then $z_1x_1|x_tc$ (in T_1') and $z_1c|x_1x_t$ (in T_2') form incompatible quartets. □

In fact, due to the fact that T_1 and T_2 are incompatible (and thus so are T_1' and T_2' in view of Proposition 5.1) we can (by Observation 3.2) safely suppress (in D) all the degree-2 nodes labeled by taxa in C, and (in D') all the degree-2 nodes labeled by taxa in C', without altering the treewidth of D or D'. Without loss of generality we assume that this suppression has taken place.

Observe also that the part of D that corresponds to the common chain C now resembles a $2 \times t$ grid and in D' is a $2 \times d$ grid. For a common chain C of length t, let $g(C)$ be the corresponding $2 \times t$ grid in D and similarly define $g(C')$ in D' for the clipped common chain of length d.

Now, assume that we have an optimal tree decomposition $\mathbb{T}$ of D of width k, i.e., the maximum bag size in $\mathbb{T}$ is $k + 1$. First of all, by a standard minor argument, it is immediate that application of the cc-reduction rule cannot increase the treewidth: the resulting display graph D' is a minor of D.

Our strategy will be as follows: Given an optimal tree decomposition $\mathbb{T}'$ for D', we will modify it to construct a tree decomposition for D that in the worst case has width at most $tw(D') + 1$, thus proving $tw(D') \geq tw(D) - 1 = k - 1$. (In some cases we will be able to prove the stronger result that $tw(D) = tw(D')$.)

We distinguish two cases.

Case 1: The common chain $g(C)$ ***is a separator in D.*** In other words, deleting $g(C)$ from D will result in two connected components.[5] In this case we will show that clipping the common chain C down to length 2 by applying a 2-cc step preserves the treewidth of D. We note that an application of a 2-cc step causes $g(C')$ to resemble a C_4 in D', where as usual, C_4 is a cycle of length 4.

Lemma 5.1. *Let T_1', T_2' be two incompatible unrooted binary phylogenetic trees that are obtained after a single application of the operation 2-cc(C) on T_1 and T_2 where $g(C)$ is a separator in $D(T_1, T_2)$. Then $d_{tw}(T_1, T_2) = d_{tw}(T_1', T_2')$.*

Proof. Let D be the display graph of T_1, T_2 and D' the display graph after we clipped the common chain C down to length 2 and let $g(C')$ be the 2×2 grid induced by the common chain in D'. Remember that $g(C')$ has 4 vertices $\{v_1, u_1, v_t, u_t\}$ such that $\{v_1, v_t\} \subset V(T_1)$ and $\{u_1, u_t\} \subset V(T_2)$. Let $\mathbb{T}'$ be an optimal tree decomposition for D'.

Consider the grid $g(C')$ in D' corresponding to the clipped chain C' of length $d = 2$. We will expand $g(C')$ inductively by first inserting the parents v_2, u_2 of the clipped taxon x_2 (and an edge between them): These two vertices will be inserted in the C_4 induced by $\{v_1, v_t, u_1, u_t\}$. After the j-th step, $j \leq t - d$, of this process, we will have retrieved the parents of taxa $x_2, \ldots x_{j+1}$. Step $(j + 1)$ continues by expanding the current $g(C'')$ of length $j + 2$ by inserting the parents v_{j+2}, u_{j+2} in the C_4 induced by $v_{j+1}, v_t, u_{j+1}, u_t$. We will show how, at each step, we can update the tree decomposition $\mathbb{T}'$, without increasing its width, so that the new one will be a valid tree decomposition for the updated display graph.

We will start by proving the base case. For this, we will find helpful the following claim about the structure of $\mathbb{T}'$.

Claim 5.1. *There exists an optimal tree decomposition $\mathbb{T}'$ of D' such that $\mathbb{T}'$ contains two adjacent degree-2 bags A_1 and A_2 where $A_1 = \{v_1, u_1, v_t\}$, $A_2 = \{v_t, u_1, u_t\}$.*

Proof. Observe that since $g(C)$ is a separator in D, then so is $g(C')$ in D'. In D' we delete the edges $\{v_1, v_t\}$ and $\{u_1, u_t\}$ and we obtain, wlog, two connected components D_1' and D_2' such that $\{v_1, u_1\} \subset V(D_1')$ and $\{v_t, u_t\} \subset V(D_2')$.

[5] Note that, if $g(C)$ is a separator in D, then the two trees actually have a *common split* (a term we define formally in Section 5.3). However, the cluster reduction results in that section have a rather different (and implicit) flavor and do not imply the results in this section.

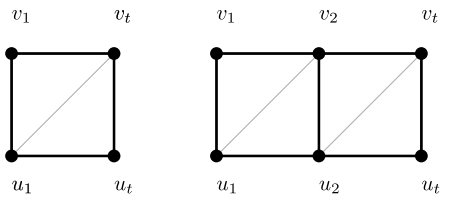

Fig. 7. An example of the inductive construction of Lemma 5.1. We construct a new tree decomposition which facilitates the extra links added by increasing the length of $g(C')$ by one, corresponding to adding the parents of the current missing taxon (in this case x_2). The gray edges are not in the display graph D' but they indicate the maximal cliques induced by the size-3 bags that we add in Lemma 5.1.

Consider optimal tree decompositions $\mathbb{T}_1$, $\mathbb{T}_2$ of D'_1, D'_2 respectively. Note that $tw(D'_1) \leq tw(D'), tw(D'_2) \leq tw(D')$ and $tw(D') \geq 3$. Since $\{v_1, u_1\} \in E(D'_1)$, there must be a bag $B_1 \in V(\mathbb{T}_1)$ that contains $\{v_1, u_1\}$. Similarly, there must be a bag $B_2 \in V(\mathbb{T}_2)$ that contains $\{v_t, u_t\}$. Attach to B_1 a new bag $A_1 = \{v_1, u_1, v_t\}$ and attach to B_2 bag $A_2 = \{v_t, u_1, u_t\}$ and join A_1, A_2 by an edge to create a new tree decomposition $\mathbb{T}'$ for D': indeed, it is immediate to see that $\mathbb{T}'$ satisfies all the treewidth conditions. Moreover, the width of this tree decomposition is $\max(tw(D'_1), tw(D'_2), 2)$. Noting that $3 \leq tw(D') \leq \max(tw(D'_1), tw(D'_2), 2) \leq max(tw(D'_1), tw(D'_2)) \leq tw(D')$ it follows that it is an optimal tree decomposition of D'. □

Given $\mathbb{T}'$ as described in the previous claim, delete bags A_1, A_2 and consider the following set of bags: $J_1 = \{v_1, v_2, u_1\}$, $J_2 = \{v_2, u_1, u_2\}$, $J_3 = \{v_2, v_t, u_2\}$ and $J_4 = \{v_t, u_2, u_t\}$. Attach J_1 to B_1 (the bag that was adjacent to A_1) and J_4 to B_2 (the bag that was adjacent to A_2) and create a path of bags from J_1 to J_4. It is easy to argue that this is a valid tree decomposition D'', defined as the display graph after the parents of x_2 have been added; see Fig. 7. First of all, for conditions (tw1) and (tw2) this is immediate by construction. Indeed, v_2 belongs to J_1, J_2, J_3 and u_2 belongs to J_2, J_3, J_4. For (tw2) observe that the edges $\{v_1, v_t\}, \{u_1, u_t\}$ are not present in $g(C'')$ so we do not need to consider them. For the new edges we have that $\{v_1, v_2\} \in J_1$, $\{u_1, u_2\} \in J_2$, $\{v_2, u_2\} \in J_3$, $\{v_2, v_t\} \in J_3$ and $\{u_2, u_t\} \in J_4$. Also, by leveraging the explicit construction of $\mathbb{T}'$ (in particular: $v_t, u_t \notin B_1$ and $u_1, v_1 \notin B_2$) we can easily verify that (tw3) is true for $\mathbb{T}''$. Finally, the width of this new tree decomposition is no greater than the width of $\mathbb{T}'$ because we only add bags of size 3 and, by construction, $\mathbb{T}'$ already contained at least one bag of size 4.

This proves that, for the base case, the treewidth of the new display graph remains unchanged. For the j-th step, we apply the arguments above where as A_1 and A_2 we use the bags $\{v_j, u_j, v_t\}$ and $\{v_t, u_j, u_t\}$ which by induction exist and are adjacent. Delete them and replace them with the following chain of bags, as before: $J_1 = \{v_j, v_{j+1}, u_j\}$, $J_2 = \{v_{j+1}, u_j, u_{j+1}\}$, $J_3 = \{v_{j+1}, v_t, u_{j+1}\}$ and $J_4 = \{v_t, u_{j+1}, u_t\}$. We continue until we add the last missing piece of $g(C)$. □

***Case 2: The common chain** C **is not a separator in** D.* We say that the $2 \times t$ grid $g(C)$ in D that corresponds to the common chain C is not a separator if the deletion of $g(C)$ from D leaves the display graph D connected. See Fig. 6 as an example of such a case and Fig. 8 for an example of their display graph. It is easy to observe that if $g(C)$ is not a separator in D then neither is $g(C')$ in D'. We will show that in this case the treewidth of D after clipping $g(C)$ down cannot decrease by more than a unit term.

Lemma 5.2. *Let T'_1, T'_2 be the two incompatible unrooted binary phylogenetic trees that are obtained after a single application of the 2-cc reduction rule on T_1 and T_2 on a common chain C such that $g(C)$ is* not *a separator in $D(T_1, T_2)$. Then we have $d_{tw}(T'_1, T'_2) \leq d_{tw}(T_1, T_2) \leq d_{tw}(T'_1, T'_2) + 1$.*

Proof. As in the separator case, we will alter the tree decomposition $\mathbb{T}'$ for D' to obtain a new tree decomposition $\mathbb{T}''$ that will be valid for D'' (the display graph with the expanded 2×3 grid $g(C'')$) and which has width at most $tw(D') + 1$. Then, we will argue how we can increase the length of this 2×3 grid $g(C'')$ to any arbitrary length without further increasing the width. So, the $+1$ term might be incurred only when we transfer from the 2×2 to the 2×3 grid but when we retrieve the rest of C we do not have to pay again in terms of increasing the width. The reason for this is that in the transition from length 2 to 3 we guarantee that the tree decomposition for the updated situation has a certain invariant property that we can exploit in order to further increase the length of the grid "for free". The initial tree decomposition might however not possess this property and we have to pay potentially a unit increase in the width of the decomposition to establish it.

Consider the grid $g(C')$ in D' corresponding to the clipped chain C' of length $d = 2$. It contains 4 vertices: $\{v_1, v_t\} \in V(T_1)$ and $\{u_1, u_t\} \in V(T_2)$. As in the separator case we will expand this $g(C')$ inductively by first inserting the parents v_2, u_2 of the clipped taxon x_2 and after the j-th step, $j \leq t - d$ of this process we will have already retrieved the parents of taxa $x_2, \ldots x_{j+1}$. The $(j+1)$th step proceeds by expanding the current $g(C'')$ of length $j+2$ by inserting the parents v_{j+2}, u_{j+2} in the C_4 induced by $v_{j+1}, v_t, u_{j+1}, u_t$.

For the base case, we will distinguish three cases. In all cases we assume without loss of generality that $\mathbb{T}'$ is an optimal *small* tree decomposition of D'. A small tree decomposition is a tree decomposition where no bag in the tree decomposition

is a subset of another (which thus also excludes the possibility of having two copies of the same bag). It is well-known that there exist optimal tree decompositions that are also small.

$|V(D')| > 4$ **and** $\exists$ **bag** $B \in V(\mathbb{T}')$ **such that** B **contains** $\{v_1, v_t, u_1, u_t\}$**.** As a first step, we claim that $|B| \geq 5$. Indeed, assume for the sake of contradiction that B contains only these four vertices and take any bag $A \in V(\mathbb{T}')$ that is adjacent to B in the tree decomposition $\mathbb{T}'$. (Such a bag must exist because $|V(D')| > 4$.) Consider their intersection $A \cap B$. By the smallness assumption on $\mathbb{T}'$ we have that $|A \cap B| \leq 3$. By standard properties of tree decompositions (see e.g., [18]) we know that $A \cap B$ is a *separator* in D' of the following two sets of vertices: $F_A = \cup_{v \in V(T_A)} B_v$, $F_B = \cup_{v \in V(T_B)} B_v$ where T_A is the connected component of $\mathbb{T}'$ that contains bag A and T_B is the connected component of $\mathbb{T}'$ that contains B if we delete the edge $\{A, B\}$ from $E(\mathbb{T}')$. But observe that $A \cap B$ cannot be a separator for separation F_A, F_B because $A \cap B \subset g(C')$ and $g(C')$ is not a separator of D'. A contradiction.

Now we proceed as follows: Create a new bag $H_1 = \{v_1, v_t, u_1, u_t, v_2\}$ and attach it to B with an edge. Create a second bag $H_2 = H_1 \cup \{u_2\} \setminus \{v_1\}$ and attach it to H_1.

We claim this is a valid tree decomposition for D'' (which is D' where $g(C')$ has increased its length by 1). Indeed, property (tw1) is immediate by construction, as is (tw3). For (tw2) observe that bag H_1 takes care of the new edges $\{v_1, v_2\}, \{v_2, v_t\}$ of $g(C'')$ and the bag H_2 of the new edges $\{v_2, u_2\}, \{u_1, u_2\}, \{u_2, u_t\}$. Note that, because $|B| \geq 5$, the new bags H_1 and H_2 do not increase the width of the decomposition.

$|V(D')| = 4$ **and** $\exists$ **bag** $B \in V(\mathbb{T}')$ **such that** B **contains** $\{v_1, v_t, u_1, u_t\}$**.** This situation can only occur if D' is the complete graph on 4 vertices K_4 (since we know $tw(D') \geq 3$). This exceptional case can be dealt with similarly to the previous case, except that the addition of bags H_1 and H_2 increase the width of the decomposition by exactly one. That is, we obtain a decomposition of D'' of width $tw(D') + 1$.

$\nexists B \in V(\mathbb{T}')$ **that contains all of** $\{v_1, v_t, u_1, u_t\}$**.** Note that every chordal completion of D' must introduce the chord $\{v_1, u_t\}$ and/or the chord $\{v_t, u_1\}$. It is well-known that each maximal clique in a chordal completion induces a bag in a corresponding tree decomposition, and each bag in a tree decomposition induces a maximal clique in a corresponding chordal completion. Assume without loss of generality that the chord $\{v_t, u_1\}$ is present[6] Then $\{v_1, u_t\}$ is not present (because otherwise the corresponding bag would contain all of $\{v_1, v_t, u_1, u_t\}$, violating the case assumption). Hence there exist two bags $A \neq B$ of $\mathbb{T}'$ that contain the sets of vertices $\{v_1, u_1, v_t\}$ and $\{u_1, u_t, v_t\}$ respectively (and possibly other vertices). Add the element v_1 to B and, in order to guarantee the running intersection property for v_1, add it also to each of the bags in the unique path from A to B in the tree decomposition T' (all these bags contain $\{u_1, v_t\}$ by the running intersection property). This might increase the width of the decomposition by at most one. We introduce H_1 next to $B \cup \{v_1\}$ and H_2 next to H_1.

- If adding v_1 *does* increase the width, it is because v_1 is added to a bag that already has maximum size. All maximum-size bags in $\mathbb{T}'$ contain at least 4 vertices (because $tw(D') \geq 3$) so after adding v_1 the maximum-size bags in the decomposition contain at least 5 vertices. Specifically, adding H_1 and H_2 cannot further increase the width of the decomposition and we obtain a decomposition of width at most $tw(D') + 1$.
- If adding v_1 does *not* increase the width, then the maximum bag size in our new v_1-augmented decomposition is at least 4 (because $|B \cup \{v_1\}| \geq 4$). Hence, adding H_1 and H_2 cannot increase the width of the decomposition by more than 1. So we again have a decomposition of width at most $tw(D') + 1$.

In all the above three cases we end up with a (not necessarily optimal) tree decomposition in which H_1 and H_2 are two adjacent size 5 bags (of degree 2 and 1 respectively). This process can now be iterated without further raising the width of the decomposition because all added bags will have size at most 5. For example, to add the parents of x_3: add a new bag $\{v_2, u_2, v_t, u_t\}$ next to H_2 ("forget" u_1 from bag H_2) and then add two new bags $\{v_2, v_3, v_t, u_2, u_t\}$ ("introduce" v_3) and $\{v_3, v_t, u_2, u_3, u_t\}$ ("forget" v_2 and "introduce" u_3).

In conclusion, from a clipped chain C' and its corresponding grid $g(C')$ in D' we can retrieve the whole original chain by increasing the treewidth of the resulting display graph by at most 1. Equivalently, clipping a common chain down to length 2 where in the display graph $D(T_1, T_2)$ the common chain is not a separator, cannot decrease the treewidth of the resulting display graph by more than 1. □

Fig. 8 shows that shortening a chain to length 2 might indeed reduce the treewidth of the display graph by 1. A natural question therefore arises: is there a constant $d > 2$ such that, if we clip a chain down to length d, the treewidth of the display graph is guaranteed to not decrease? This seems like a highly non-trivial question with deep connections to forbidden minors. But, at least in the case where the common chain is very large with respect to a function of the treewidth of the display graph D, we can show that shortening chains to a length dependent on the treewidth of D *does* preserve the treewidth.

[6] If $\{v_1, u_t\}$ is present and not $\{v_t, u_1\}$ then by topological symmetry of the chain the argument still goes through: conceptually we are then simply reconstructing the chain in the "opposite" direction.

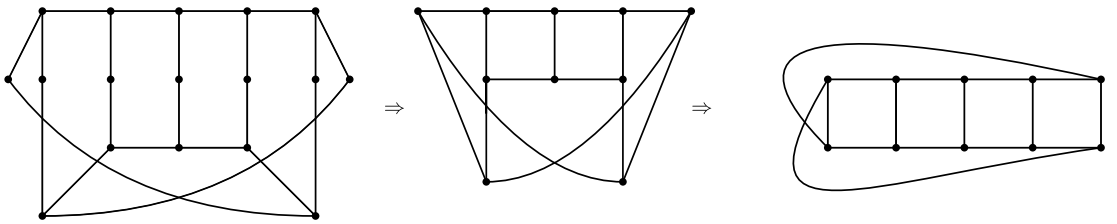

Fig. 8. The display graph (after the suppression of all vertices of degree two) of the two trees T_1, T_2 from Fig. 6. Observe that the final graph contains one of the minimal forbidden minors for treewidth 3, the Moëbius ladder on eight vertices, and $tw(D) = 4$. Observe also that if we clip the common chain down to length 2, then the treewidth of D decreases to 3 because the display graph would be in this case the Moëbius ladder on six vertices.

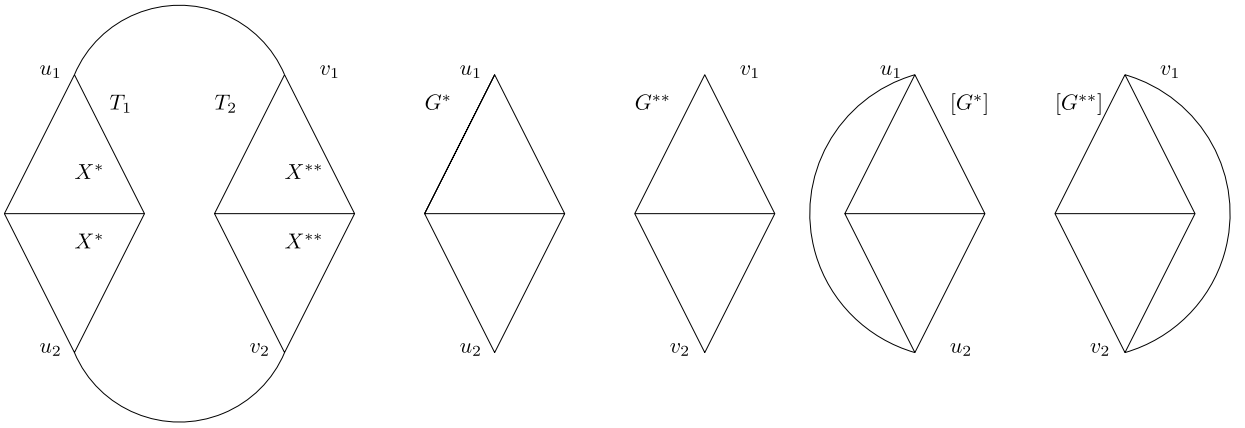

Fig. 9. Left: the display graph when T_1 and T_2 have a common split $X^*|X^{**}$. Center: the graphs G^* and G^{**} obtained by deleting the two edges inducing the common split. Right: the graphs $[G^*]$ and $[G^{**}]$ obtained from G^* and G^{**} by joining the "roots" together.

Theorem 5.2. *Let T_1, T_2 be two incompatible unrooted binary trees and $D(T_1, T_2)$ their display graph such that $tw(D(T_1, T_2)) = k$. Then, there is a function $f(k)$ such that if there exists a common chain C of length $t > f(k)$ then we can clip C down to length $f(k)$ such that $tw(D') = tw(D)$ (where as usual D' is the display graph of the trees with the shortened chains).*

Proof. Given that $tw(D(T_1, T_2)) = k \geq 3$ we can as usual without loss of generality suppress all taxa in the display graph. Now, $D(T_1, T_2)$ must have as a minor one of the forbidden minors for treewidth $k-1$. Forbidden minors for treewidth $k-1$ (where $k-1 \geq 2$) are all connected simple graphs with minimum degree 3. By the work of Lagergren [35] we know that the number of edges (and vertices) in forbidden minors for treewidth k is bounded by a function f' of k which is doubly exponential in $O(k^5)$. Let $d' = f'(k-1)$. Now, fix the image of a forbidden minor for treewidth $k-1$ inside D. Each vertex v of the minor has degree at most d', and (crudely) a degree d' vertex v can be split into at most $\leq d'$ degree-3 vertices on the image inside D (these are the vertices which via edge contractions will merge to form v). Hence a common chain longer than $(d')^2$ must necessarily contain ever more vertices which are not on the image at all, or which are degree-2 vertices on the image. For a sufficiently large function f the point is reached that, if the chain is longer than $f((d')^2)$, reducing the length of the chain by 1 cannot destroy the forbidden minor: either the image survives or a slight modification of it (with fewer degree-2 vertices) can be embedded in the graph. Hence, shortening the chain to length $f((d')^2)$ cannot reduce the treewidth below k. □

5.3. Cluster reduction rule

In this subsection we will study how the treewidth of the display graph relates to the treewidth of its *clusters* which are related to common splits:

Definition 5.1. Let T_1 and T_2 be two unrooted binary phylogenetic trees on the same set of taxa X. We say that T_1 and T_2 have a *common split* $X^*|X^{**}$ if X^* and X^{**} together form a bipartition of X and, for $i \in \{1, 2\}$, T_i has some edge e_i such that deleting e_i separates X^* from X^{**} in that tree.

In the following proofs we will refer extensively to Fig. 9.

Lemma 5.3. *Let T_1 and T_2 be two incompatible unrooted binary phylogenetic trees on the same set of taxa X and let $X^*|X^{**}$ be a common split of T_1 and T_2. Let $p = tw(D(T_1|X^*, T_2|X^*))$ and $q = tw(D(T_1|X^{**}, T_2|X^{**}))$. Then*

$$\max(p,q) \leq tw(D(T_1,T_2)) \leq \max(p,q)+1.$$

Proof. First we observe that the lower bound $\max(p,q) \leq tw(D(T_1,T_2))$ is immediate, since both $D(T_1|X^*, T_2|X^*)$ and $D(T_1|X^{**}, T_2|X^{**})$ are minors of $D(T_1,T_2)$.

For the upper bound, we will first deal with the case when $|X^*|, |X^{**}| \geq 3$. Let $e_1 = \{u_1, v_1\}$ be the edge that induces the $X^*|X^{**}$ split in T_1, and let $e_2 = \{u_2, v_2\}$ be the edge which induces the split in T_2. If we delete both the edges $\{u_1, v_1\}$ and $\{u_2, v_2\}$ from $D(T_1,T_2)$ then we obtain a graph with two connected components. Each one of these two components has two degree-2 vertices, the endpoints of the two deleted edges. One of these components is a "rooted" version of $D(T_1|X^*, T_2|X^*)$, which we call G^*, and the other is a "rooted" version of $D(T_1|X^{**}, T_2|X^{**})$, which we call G^{**} where, in contrast with $D(T_1|X^*, T_2|X^*)$, $D(T_1|X^{**}, T_2|X^{**})$, we do not suppress the degree-2 vertices v_1, v_2, u_1, u_2. Note that, due to the cardinality constraints on X^* and X^{**}, $p = tw(G^*)$ and $q = tw(G^{**})$ because $D(T_1|X^*, T_2|X^*)$ can be obtained from G^* by suppressing the degree-2 vertices which does not alter the treewidth (because the pathological case of Observation 3.1 does not apply). Similarly for the other component. Assume without loss of generality that u_1 and u_2 are in G^*, and v_1 and v_2 are in G^{**}. Let $\mathbb{T}^*$ and $\mathbb{T}^{**}$ be minimum-width tree decompositions of G^* and G^{**} respectively. Locate a bag B^* of $\mathbb{T}^*$ that contains u_1 and a bag B^{**} of $\mathbb{T}^{**}$ that contains v_1. Introduce a bag $\{u_1, v_1\}$ and insert it between B^* and B^{**}. Clearly, the width in this merged tree decomposition is not altered. It remains only to ensure that the decomposition covers the edge $\{u_2, v_2\}$. This can be achieved simply by adding (say) u_2 to every bag in the tree decomposition of G^{**}, which increases the size of all bags by at most one. The result follows.

Now, we deal with the case where $|X^*| \leq 2$ and/or $|X^{**}| \leq 2$. First of all, we observe that since T_1, T_2 are incompatible by assumption, it is not the case that $|X^*|, |X^{**}| \leq 2$ at the same time. So, at least one of $|X^*|, |X^{**}|$ must be at least 3. Suppose $|X^*| = 2$ and $|X^{**}| \geq 3$. Observe that in this case $tw(G^*) = 2 \neq p = 1$ but $tw(G^{**}) = q \geq 2$, so $\max(p,q) \geq 2$. Hence the construction from the previous case—adding bag $\{u_1, v_1\}$ and then adding u_2 to all bags—again cannot increase the width of the decomposition by more than 1. The case $|X^*| = 1$ is somewhat strange because then $D(T_1|X^*, T_2|X^*)$ is just a single vertex. However, the upper bound still goes through because $tw(G^{**}) = q \geq 2$ and $D(T_1,T_2)$ can be obtained from G^{**} by connecting the two roots of G^{**} by an edge and then subdividing this new edge with a single degree-2 vertex. Adding an edge to a graph can increase its treewidth by at most 1, and edge subdivision is treewidth invariant. □

Now, let $[G^*]$ be the graph obtained from G^* by adding the edge $\{u_1, u_2\}$, and $[G^{**}]$ be obtained from G^{**} by adding the edge $\{v_1, v_2\}$. See again Fig. 9.

Observation 5.1. *$tw(G^*) \leq tw([G^*]) \leq tw(D(T_1,T_2))$ and $tw(G^{**}) \leq tw([G^{**}]) \leq tw(D(T_1,T_2))$.*

Proof. The lower bounds are immediate by a standard minor argument. The upper bounds are also obtained via minors. Specifically, observe that $[G^*]$ can be obtained from $D = D(T_1,T_2)$ by completely contracting the part of D that lies between v_1 and v_2 (i.e. the X^{**} part of D). A symmetrical argument holds for $[G^{**}]$ by completely contracting the X^* part of D. □

The following theorem strengthens Lemma 5.3 by adding necessary and sufficient conditions for the lower bound to be attained.

Theorem 5.3. *Let T_1 and T_2 be two incompatible unrooted binary phylogenetic trees on the same set of taxa X and let $X^*|X^{**}$ be a common split of T_1 and T_2. Let $p = tw(D(T_1|X^*, T_2|X^*))$ and $q = tw(D(T_1|X^{**}, T_2|X^{**}))$. Assume, without loss of generality, that $p \leq q$. Then $tw(D(T_1,T_2)) = \max(p,q)$ if and only if the following holds:*

1. *(Case $p < q$): $tw([G^{**}]) = tw(G^{**})$,*
2. *(Case $p = q$): $tw([G^{**}]) = tw(G^{**})$ and $tw([G^*]) = tw(G^*)$.*

Proof. We consider both cases and both directions of implication.

1. (Case $p < q$, $\Rightarrow$) Assume $p < q$ and $tw(D(T_1,T_2)) = \max(p,q) = q$. Now, by Observation 5.1, $tw([G^{**}]) \leq tw(D(T_1,T_2)) = q = tw(G^{**})$. The bound $tw(G^{**}) \leq tw([G^{**}])$ also follows from Observation 5.1, so $tw([G^{**}]) = tw(G^{**})$.
2. (Case $p = q$, $\Rightarrow$) Assume $p = q$ and $tw(D(T_1,T_2)) = \max(p,q) = p = q$. Both $tw([G^{**}]) = tw(G^{**})$ and $tw([G^*]) = tw(G^*)$ follow from Observation 5.1.
3. (Case $p < q$, $\Leftarrow$) Observe that the statement $tw([G^{**}]) = tw(G^{**})$ holds if and only if there exists a minimum-width tree decomposition of G^{**} in which v_1 and v_2 are both in the same bag B^{**}. So, let us assume the existence of such a tree decomposition $\mathbb{T}^{**}$ and bag B^{**}. Construct a minimum-width tree decomposition $\mathbb{T}^*$ of G^*. Suppose $\mathbb{T}^*$ contains a bag B^* that contains both u_1 and u_2. We can merge $\mathbb{T}^*$ and $\mathbb{T}^{**}$ by inserting bags $\{u_1, v_1, u_2\}$ and $\{u_2, v_1, v_2\}$

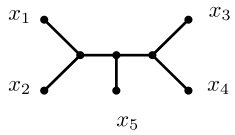

Fig. 10. Without loss of generality we can assume that an unrooted binary phylogenetic tree on 5 taxa has (up to relabeling of taxa) the topology T_1.

between B^* and B^{**}. The size-3 bags do not influence the width of the decomposition, so $tw(D(T_1, T_2)) \leq \max(p, q)$, and $tw(D(T_1, T_2)) = \max(p, q)$ then follows from Lemma 5.3. If no such bag B^* exists then create it by first adding (say) u_2 to every bag of $\mathbb{T}^*$. The addition of u_2 to every bag potentially increases the width of $\mathbb{T}^*$ by 1, but due to the fact that $p < q$ we have $p + 1 \leq q$, so $\max(p + 1, q) \leq \max(p, q)$ and the earlier argument goes through.

4. (Case $p = q$, $\Leftarrow$) This is very similar to the (Case $p < q$, $\Leftarrow$) argument. The main difference is that, due to the strengthened starting assumption, both bags B^{**} and B^* are guaranteed to exist. Hence the "If no such bag B^*..." part of the argument will never be required. □

The above results show that the treewidth of the display graph behaves rather differently around common splits than other phylogenetic incongruence measures. Many such measures are (essentially) additive (i.e. the distance is the sum of the X^* and X^{**} parts) [4,36,14], contrasting with the maximum function used in treewidth. As we demonstrate later in Section 7 this is one of the reasons why treewidth distance can be substantially lower than, for example, d_{MAF}. A second point worth noting is that, while Theorem 5.3 describes necessary and sufficient conditions for the treewidth of the display graph to achieve the lower bound, it is not yet clear what (phylogenetic) properties of T_1 and T_2 actually create these conditions. Expressed differently, and for simplicity focusing on the case $p < q$: what properties do T_1 and T_2 need to have to ensure $tw([G^{**}]) = tw(G^{**})$? It is perhaps relevant to observe that the graphs $[G^*], [G^{**}]$ can themselves be viewed, modulo a treewidth-invariant suppression of a single degree-2 vertex, as display graphs of appropriately rooted phylogenetic trees. Taking $[G^*]$ as an example: take the two trees $T_1|X^*$ and $T_2|X^*$ and attach a new placeholder taxon ρ at points u_1 and u_2, respectively.

6. The unit ball of d_{tw} compared to that of d_{TBR} and d_{MP}

In this section we will compare the unit ball neighborhood of d_{tw} with those of d_{TBR} and d_{MP}. Recall that given a distance d and a phylogenetic tree T on X the unit neighborhood of T under d is the set of all phylogenetic trees T' on X with the property that $d(T, T') = 1$ (see, e.g. [30,37], for results that characterize the unit ball neighborhoods of d_{TBR} and d_{MP}).

Theorem 6.1. *Suppose that T and T' are a pair of unrooted binary phylogenetic trees on X with $d_{MP}(T, T') = 1$ or $d_{TBR}(T, T') = 1$. Then we also have $d_{tw}(T, T') = 1$.*

Proof. First of all, we note that, because both TBR and MP distance are metrics (and thus satisfy the identity of indiscernibles property) we can assume that T_1 and T_2 are incompatible. We will first show that the claim is true for the TBR distance. Take two (necessarily incompatible) binary phylogenetic trees T, T' such that $d_{TBR}(T, T') = 1$. By combining the results of [1] where it was shown that $d_{MAF}(T_1, T_2) = d_{TBR}(T_1, T_2) + 1$ and the result of [34] where it was shown that $tw(D(T_1, T_2)) \leq d_{MAF}(T_1, T_2) + 1$ we have that

$$tw(D(T_1, T_2)) \leq d_{TBR}(T_1, T_2) + 2,$$

for any two phylogenetic trees T_1, T_2.

Now if T, T' are such that $d_{TBR}(T, T') = 1$ we conclude by the above that $tw(D(T, T')) \leq 3$ and by the assumption that T, T' are incompatible we have that $d_{tw}(T, T') = 1$.

Now we will deal with the Maximum Parsimony distance. Let T, T' be two (necessarily incompatible) unrooted binary phylogenetic trees such that $d_{MP}(T, T') = 1$. Using Theorem 5.1, we assume without any loss of generality that T and T' share no common pendant subtrees. Therefore, we can apply [37, Theorem 6.4] on T, T' which characterizes the unit ball neighborhood of the maximum parsimony distance. There it was shown that $d_{MP}(T, T') = 1$ if and only if either (1) $d_{TBR}(T, T') = 1$, in which case we are done since we are in the TBR case or (2) $d_{TBR}(T, T') = 2$ and using common pendant subtree (CPS) reductions we can transform T and T' into a pair of trees with *precisely* five taxa. (All unrooted binary phylogenetic trees on 5 taxa are caterpillars and modulo relabeling of taxa there is only one caterpillar topology on 5 taxa.) Since d_{tw} is preserved by CPS reduction in view of Theorem 5.1, we can assume without loss of generality that T and T' both have 5 taxa, and T is the tree T_1 depicted in Fig. 10. Let $D = D(T, T')$ be the display graph formed from T and T' in which we subsequently suppress all vertices of degree-2. (Suppression does not alter the treewidth, by Observation 3.2.) It is easy to observe that D has at most (in fact, exactly) 6 vertices.

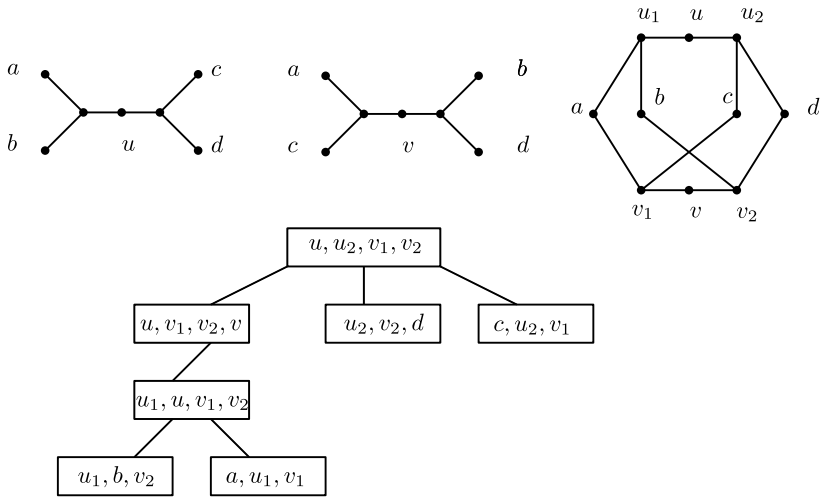

Fig. 11. Top: The two quartets $ab|cd$ and $ac|bd$ and their corresponding display graph (denoted $D^0 = D$ in the proof of Claim 7.1). Bottom: a width-3 tree decomposition of D in which u, v are in the same bag.

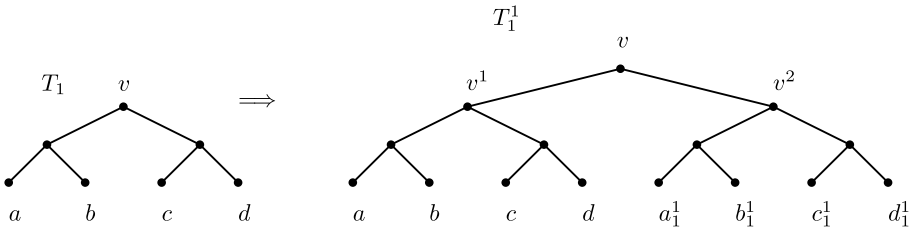

Fig. 12. An example of doubling the tree $T_1 = ab|cd$. When we create the second copy we label the taxa of the new copy appropriately to reflect the stage of doubling they appear at (superscript) and the tree to which they belong (subscript).

Now, assume that $tw(D) > 3$ so that $d_{tw}(T, T') > 1$. Then, D must have as a minor one of the forbidden minors for treewidth 3. In other words, one of the forbidden minors for treewidth 3 can be obtained by a series of edge deletions/contractions on D. There are precisely 4 forbidden minors for treewidth 3 [3], 2 of which are on 6 vertices or less: the K_5 and the Octahedron graph. Both of them have uniform degree 4. On the other hand, recall that the degree of each vertex of D is 3 (because T, T' are unrooted *binary* phylogenetic trees), so each degree-4 vertex of the minor maps to at least 2 vertices of D. This is clearly impossible. So D cannot contain as a minor any of the forbidden minors for treewidth 3 which shows that $tw(D) \leq 3$. By assumption, T, T' are incompatible so $tw(D(T, T')) = 3 \Rightarrow d_{tw}(T, T') = 1$. □

In the following section we will show that the converse of the above claim, namely that $d_{tw}(T_1, T_2) = 1 \Rightarrow d_{MP}(T_1, T_2) = 1$ is certainly not true (and that the same holds for the TBR distance).

7. On the gap between d_{tw} and d_{TBR}, d_{MP}

The purpose of this section is to explore how far treewidth distance d_{tw} can be from the other two distances considered in this manuscript, namely maximum parsimony distance d_{MP} and TBR distance d_{TBR}. In particular we will provide an example of a sequence of pairs of trees whose treewidth distance is as low as 1 (i.e., the treewidth of their display graph is at most 3) but such that the corresponding TBR and MP distances can be arbitrarily large.

The construction starts with the 2 incompatible quartets (unrooted binary trees on 4 taxa) $T_1 = ab|cd$ and $T_2 = ac|bd$. Without any loss of generality, we assume that both of the quartets contain a degree-2 vertex in the "middle" namely, vertices u, v respectively (see Fig. 11). Note that with or without these degree-2 vertices the display graph has treewidth exactly 3 (by Observation 3.2).

Given a tree T with a single degree-2 vertex we define the following *doubling* operation as follows:

Doubling tree operation: Given a tree T, with a unique degree-2 vertex v, the doubling of T, denoted by (T, T), is constructed as follows: we take 2 copies of T and we join with an edge their unique degree-2 vertices. We subdivide this new edge such that (T, T) has a unique degree-2 vertex (see Fig. 12).

This operation will be the base of our construction. We will construct trees T_1^i and T_2^i, for any step i, inductively as follows: $T_1^1 = (T_1, T_1)$ and $T_1^{i+1} = (T_1^i, T_1^i)$. Similarly for T_2^2 and subsequently for $T_2^{i+1} = (T_2^i, T_2^i)$. Let D^i be the display graph of T_1^i and T_2^i. Observe that since we start from T_1, T_2 on a common set of 4 taxa $\{a, b, c, d\}$, all the new doubled trees are on the same taxon set by labeling the new leaves appropriately, and so their display graph is well defined and unique. Initially, let $D = D^0$ be the display graph of $T_1^0 = T_1$ and $T_2^0 = T_2$. We will show that $tw(D^i) = 3, \forall i$.

Claim 7.1. *For every step i we have that $d_{tw}(T_1^i, T_2^i) = 1$. Equivalently, we have that $tw(D^i) = 3$.*

Proof. The proof is by an inductive argument. For the base of the induction, we first construct a tree decomposition of width 3 with specific properties: see Fig. 11.

As is apparent from the base case, we can assume without any loss of generality that the two degree-2 vertices u, v in T_1, T_2 respectively, are in the same bag of the tree decomposition of their display graph D. We will exploit this fact in the following. For the induction step we assume that the display graph D^i formed by T_1^i and T_2^i has treewidth 3. We will show a tree decomposition for D^{i+1} of width equal to the width of the tree decomposition of D^i. We can construct D^{i+1} from D^i as follows: take two copies of D_i, let's call them D_1^i and D_2^i. Each copy D_j^i, $j \in \{1, 2\}$ has two degree-2 vertices: one, let's call it u_j^i is the degree-2 vertex resulting after repeated doubling of the T_1 tree and the other, let's call it v_j^i from doubling the T_2 tree. For each display graph D_j^i let $\mathbb{T}_j^i$ be its tree decomposition which by the inductive hypothesis has width 3. Moreover, as explained, we can assume without any loss of generality that the two degree-2 vertices u_j^i and v_j^i are in the same bag B_j. Observe that D^{i+1} has two new degree two vertices, u^*, v^*: u^* will be connected with each u_j^i and v^* with each v_j^i, $j \in \{1, 2\}$. Construct $\mathbb{T}^{i+1}$ as follows: locate the bags B_j that contain $\{u_j^i, v_j^i\}$, $j \in \{1, 2\}$. Such bags exist by the inductive hypothesis. Create the following chain of bags: $B_1 - \{u^*, u_1^i, v_1^i\} - \{u^*, v^*, v_1^i\} - \{u^*, v^*, v_2^i\} - \{u^*, u_2^i, v_2^i\} - B_2$. It is immediate that $\mathbb{T}^{i+1}$ is a valid tree decomposition for D^{i+1} of width no higher than the width of $\mathbb{T}^i$ (and u^*, v^* are in the same bag) so the claim follows. □

So the treewidth distance d_{tw} of T_1^i and T_2^i remains 1 for any i. We will now give lower bounds on $d_{TBR}(T_1^i, T_2^i)$. We claim that $d_{TBR}(T_1^i, T_2^i) > d_{TBR}(T_1^j, T_2^j)$ for $i > j$. In particular $d_{TBR}(T_1^{i+1}, T_2^{i+1}) > d_{TBR}(T_1^i, T_2^i)$, for all $i \geq 0$. We will prove the claim using the *maximum agreement forest* distance which, by the result of Allen and Steel [1], is equivalent to TBR: $d_{MAF}(T_1, T_2) = d_{TBR}(T_1, T_2) + 1$. First of all, it is not too difficult to verify that (after suppression of the two degree-2 vertices[7]) $d_{MAF}(T_1, T_2) = 2$.

Let T_j^{i+1} be the two trees obtained after we double T_j^i, for $j \in \{1, 2\}$ and let $d_{MAF}(T_1^i, T_2^i) = p \in \mathbb{N}^+$. We assume without loss of generality that neither of T_1^{i+1}, T_2^{i+1} has a degree-2 vertex. We distinguish between two cases: Let $e_1(e_2)$ be the edge used to connect the two copies of $T_1^i(T_2^i)$ to construct $T_1^{i+1}(T_2^{i+1})$. We say that an edge is *deleted* by an agreement forest if it is an edge that is deleted in order to obtain the agreement forest. It is easy to observe that if e_1 is deleted in an agreement forest, then so is e_2 because of the symmetric properties of the constructed graphs T_1^{i+1}, T_2^{i+1}. Now, fix m to be an arbitrary maximum agreement forest.

Edges $e_1(e_2)$ are deleted by m: Note that by deleting $e_1(e_2)$ we obtain two disjoint copies of the trees $T_1^i(T_2^i)$. In this case we observe that $d_{MAF}(T_1^{i+1}, T_2^{i+1}) = 2d_{MAF}(T_1^i, T_2^i) = 2p$ since any maximum agreement forest that does not use $e_1(e_2)$ can and should select a maximum agreement forest for the pair of trees T_1^i, T_2^i, and do this twice (since there are two disjoint copies of these trees).

Neither of these edges is deleted by m: Then these edges are used by the image of some component C of the agreement forest m. If we split C into two pieces (at the edges e_1 and e_2) we increase the size of the agreement forest by 1 and obtain an agreement forest that does not use either edge e_1 or e_2. From the previous case we know that any agreement forest that does not use these edges has at least $2p$ components. Hence, $d_{MAF}(T_1^{i+1}, T_2^{i+1}) \geq 2p - 1$.

Lemma 7.1. *The MAF distance between T_1^{i+1} and T_2^{i+1} is at least $2 \times d_{MAF}(T_1^i, T_2^i) - 1 > d_{MAF}(T_1^i, T_2^i)$.*

Theorem 7.1. *There exists an infinite subfamily of trees T_1, T_2 such that $d_{tw}(T_1, T_2) = 1$ whereas $d_{TBR}(T_1, T_2)$ is unbounded.*

Finally, we turn to d_{MP}:

Theorem 7.2. *There exists an infinite subfamily of trees T_1, T_2 such that $d_{tw}(T_1, T_2) = 1$ whereas $d_{MP}(T_1, T_2)$ is unbounded.*

[7] Agreement forests are unaffected by suppression of degree-2 vertices.

Proof. In fact, this is a strengthening of the previous theorem because d_{MP} is always a lower bound on d_{TBR}. Observe that the tree T_1^i contains 2^i copies of each taxon. We assign all the copies of taxa a and b the state 0, and all copies of taxa c and d the state 1. It can be easily verified that the parsimony score of T_1^i on such a character is at most 2^i (e.g. assign state 0 to each node that is the common parent of an $\{a, b\}$ copy, and state 1 to all other internal nodes). However, on the same character the parsimony score of T_2^i will be at least $2 \cdot 2^i$. To see this, observe that there will unavoidably always be one mutation on the two edges between each a and c copy, and one mutation on the two edges between each b and d copy. Hence, $d_{MP}(T_1^i, T_2^i) \geq 2 \cdot 2^i - 2^i$ and this grows to infinity. □

8. Discussion and open problems

In this paper we presented several algorithmic and combinatorial results on the treewidth distance d_{tw}, including its behavior under three commonly used tree reduction rules and its diameter and unit ball neighborhood. There are a number of interesting problems that remain open, and we discuss some of them below.

A major open question is whether it is **NP**-hard to compute the treewidth distance d_{tw} between two trees. This is equivalent to computing the treewidth of the display graph of these two trees, which is a cubic graph after suppressing all degree-2 vertices. Although computing the treewidth of general graphs is **NP**-hard, even for graphs whose maximum degree is at most 9 [2,13], it is still unknown whether the treewidth of cubic graphs can be computed in polynomial time. Hence it is also interesting to understand the complexity of computing the treewidth of cubic graphs, and whether it has the same complexity of computing that of display graphs. One can also investigate whether, compared to general graphs, improved running times and/or approximation ratios can be obtained for approximating the treewidth of display graphs. (See [9] for a recent overview of approximation algorithms for treewidth.) Irrespective of whether it is an **NP**-hard problem, it is of interest to explore whether the structure of display graphs can be leveraged to compute their treewidth quickly *in practice*. Aside from treewidth, the structure of display graphs is itself worthy of attention: is it **NP**-hard to recognize a display graph (after suppression of degree-2 nodes)?

Another question concerns the common chain reduction, that is, whether there exists a universal constant d such that reducing common chains to length d, preserves the treewidth of the display graph? This is likely to require deep insights into forbidden minors—in particular the way they interact with chain-like regions of graphs (that are not separators). In [33] a question with a similar flavor has been raised concerning minors and display graphs. In particular, under which circumstances does the presence of ever larger grid minors in display graphs, act as a certificate for increasing incongruence (i.e. dissimilarity) between two phylogenetic trees? Additionally, one can ask whether concepts such as forbidden minors and forbidden subgraphs require some modification to be useful for the phylogenetics community, for whom display graphs are not a goal in themselves, but a lens through which to better understand the trees that form them. In [24,42], for example, the authors have initiated the study of *(forbidden) phylogenetic minors* as a tool to understand the compatibility of sets of trees. All these minor-related questions appear to be extremely rich and non-trivial.

At the empirical level, initial numerical experiments suggest that treewidth distance can be "low" compared to traditional phylogenetic distances, such as the well-known TBR distance. Is this phenomenon more widespread? In how far is this an artefact of the way treewidth distance decomposes around common splits? Are there traditional phylogenetic distances and measures which are verifiably (and/or empirically) close to treewidth distance—and, if so, why? Finally, and crucially: can we leverage low treewidth distance to develop efficient algorithms (based on dynamic programming over tree decompositions) for other phylogenetic distances and measures?

Acknowledgements

The authors would like to sincerely thank an anonymous referee for the many helpful comments that improved the content and the presentation of the manuscript. Steven Kelk and Taoyang Wu acknowledge the support of London Mathematical Society grant SC7-1516-05 and Georgios Stamoulis the support of a NWO TOP 2 grant.

References

[1] B. Allen, M. Steel, Subtree transfer operations and their induced metrics on evolutionary trees, Ann. Comb. 5 (1) (2001) 1–15.
[2] S. Arnborg, D. Corneil, A. Proskurowski, Complexity of finding embeddings in a k-tree, SIAM J. Algebr. Discrete Methods 8 (2) (1987) 277–284.
[3] S. Arnborg, A. Proskurowski, D.G. Corneil, Forbidden minors characterization of partial 3-trees, Discrete Math. 80 (1) (1990) 1–19.
[4] M. Baroni, C. Semple, M. Steel, Hybrids in real time, Syst. Biol. 55 (1) (2006) 46–56.
[5] J. Baste, C. Paul, I. Sau, C. Scornavacca, Efficient FPT algorithms for (strict) compatibility of unrooted phylogenetic trees, Bull. Math. Biol. 79 (4) (2017) 920–938.
[6] J. Blair, B. Peyton, An introduction to chordal graphs and clique trees, in: Graph Theory and Sparse Matrix Computation, Springer New York, New York, NY, 1993, pp. 1–29.
[7] H. Bodlaender, A tourist guide through treewidth, Acta Cybernet. 11 (1–2) (1994) 1.
[8] H. Bodlaender, A linear-time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. 25 (6) (1996) 1305–1317.
[9] H. Bodlaender, P. Drange, M. Dregi, F. Fomin, D. Lokshtanov, M. Pilipczuk, An $O(c^k n)$ 5-approximation algorithm for treewidth, SIAM J. Comput. 45 (2) (2016) 317–378.
[10] H. Bodlaender, F. Fomin, A. Koster, D. Kratsch, D. Thilikos, On exact algorithms for treewidth, ACM Trans. Algorithms 9 (1) (December 2012) 12:1–12:23.
[11] H. Bodlaender, A. Koster, Treewidth computations I. Upper bounds, Inform. and Comput. 208 (3) (2010) 259–275.

[12] H. Bodlaender, A. Koster, Treewidth computations II. Lower bounds, Inform. and Comput. 209 (7) (2011) 1103–1119.
[13] H. Bodlaender, D. Thilikos, Treewidth for graphs with small chordality, Discrete Appl. Math. 79 (1–3) (1997) 45–61.
[14] M. Bordewich, C. Scornavacca, N. Tokac, M. Weller, On the fixed parameter tractability of agreement-based phylogenetic distances, J. Math. Biol. 74 (1–2) (2017) 239–257.
[15] M. Bordewich, C. Semple, Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable, IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (3) (2007) 458–466.
[16] D. Bryant, J. Lagergren, Compatibility of unrooted phylogenetic trees is FPT, Theoret. Comput. Sci. 351 (3) (2006) 296–302.
[17] J. Chuzhoy, Excluded grid theorem: improved and simplified, in: Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, ACM, 2015, pp. 645–654.
[18] M. Cygan, F. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, 1st edition, Springer Publishing Company, 2015, Incorporated.
[19] H. Dell, T. Husfeldt, B. Jansen, P. Kaski, C. Komusiewicz, F. Rosamond, The first parameterized algorithms and computational experiments challenge, in: Jiong Guo, Danny Hermelin (Eds.), 11th International Symposium on Parameterized and Exact Computation, IPEC 2016, August 24–26, 2016, Aarhus, Denmark, in: LIPIcs, vol. 63, Schloss Dagstuhl – Leibniz-Zentrum Fuer Informatik, 2016, pp. 30:1–30:9.
[20] R. Diestel, Graph Theory, Springer-Verlag Berlin and Heidelberg GmbH & Company KG, 2010.
[21] Y. Ding, S. Grünewald, P. Humphries, On agreement forests, J. Combin. Theory Ser. A 118 (7) (2011) 2059–2065.
[22] R. Downey, M. Fellows, Fundamentals of Parameterized Complexity, vol. 4, Springer, 2013.
[23] V. Dujmovic, D. Eppstein, D. Wood, Genus, treewidth, and local crossing number, in: Emilio Di Giacomo, Anna Lubiw (Eds.), Graph Drawing and Network Visualization – 23rd International Symposium, Revised Selected Papers, GD 2015, Los Angeles, CA, USA, September 24–26, 2015, in: Lecture Notes in Computer Science, vol. 9411, Springer, Los Angeles, CA, USA, 2015, pp. 87–98.
[24] D. Fernández-Baca, S. Vakati, On compatibility and incompatibility of collections of unrooted phylogenetic trees, Discrete Appl. Math. (2017), https://doi.org/10.1016/j.dam.2017.05.002, in press.
[25] M. Fischer, S. Kelk, On the maximum parsimony distance between phylogenetic trees, Ann. Comb. 20 (1) (2016) 87–113.
[26] V. Gogate, R. Dechter, A complete anytime algorithm for treewidth, in: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, UAI '04, AUAI Press, 2004, pp. 201–208.
[27] A. Grigoriev, S. Kelk, N. Lekić, On low treewidth graphs and supertrees, J. Graph Algorithms Appl. 19 (1) (2016) 325–343.
[28] M. Grohe, D. Marx, On tree width, bramble size, and expansion, J. Combin. Theory Ser. B 99 (1) (2009) 218–228.
[29] R. Gysel, K. Stevens, D. Gusfield, Reducing problems in unrooted tree compatibility to restricted triangulations of intersection graphs, in: Raphael Ben, Jijun Tang (Eds.), Algorithms in Bioinformatics (Proceedings of WABI2012), in: Lecture Notes in Computer Science, vol. 7534, Springer, Berlin, Heidelberg, 2012, pp. 93–105.
[30] P. Humphries, T. Wu, On the neighborhoods of trees, IEEE/ACM Trans. Comput. Biol. Bioinform. 10 (3) (2013) 721–728.
[31] D. Huson, R. Rupp, C. Scornavacca, Phylogenetic Networks: Concepts, Algorithms and Applications, Cambridge University Press, 2011.
[32] S. Kelk, M. Fischer, On the complexity of computing MP distance between binary phylogenetic trees, Ann. Comb. 21 (4) (2017) 573–604.
[33] S. Kelk, M. Fischer, V. Moulton, T. Wu, Reduction rules for the maximum parsimony distance on phylogenetic trees, Theoret. Comput. Sci. 646 (20) (2016) 1–15.
[34] S. Kelk, L. van Iersel, C. Scornavacca, M. Weller, Phylogenetic incongruence through the lens of monadic second order logic, J. Graph Algorithms Appl. 20 (2) (2016) 189–215.
[35] J. Lagergren, Upper bounds on the size of obstructions and intertwines, J. Combin. Theory Ser. B 73 (1) (1998) 7–40.
[36] S. Linz, C. Semple, A cluster reduction for computing the subtree distance between phylogenies, Ann. Comb. 15 (3) (2011) 465–484.
[37] V. Moulton, T. Wu, A parsimony-based metric for phylogenetic trees, Adv. in Appl. Math. 66 (2015) 22–45.
[38] C. Semple, Reconstructing Evolution – New Mathematical and Computational Advances, Chapter Hybridization Networks, Oxford University Press, 2007.
[39] C. Semple, M. Steel, Phylogenetics, Oxford University Press, 2003.
[40] M. Steel, Phylogeny: Discrete and Random Processes in Evolution, SIAM, 2016.
[41] S. Vakati, D. Fernández-Baca, Graph triangulations and the compatibility of unrooted phylogenetic trees, Appl. Math. Lett. 24 (5) (2011) 719–723.
[42] S. Vakati, D. Fernández-Baca, Compatibility, incompatibility, tree-width, and forbidden phylogenetic minors, in: LAGOS'15 – {VIII} Latin-American Algorithms, Graphs and Optimization Symposium, Electron. Notes Discrete Math. 50 (2015) 337–342.
[43] L. van Iersel, S. Kelk, C. Scornavacca, Kernelizations for the hybridization number problem on multiple nonbinary trees, J. Comput. System Sci. 82 (6) (2016) 1075–1089.
[44] L. van Iersel, S. Linz, A quadratic kernel for computing the hybridization number of multiple trees, Inform. Process. Lett. 113 (9) (2013) 318–323.
[45] C. Whidden, F. Matsen, Calculating the unrooted subtree prune-and-regraft distance, preprint, arXiv:1511.07529, 2015.