Marchand *et al.*
Algorithms for Molecular Biology (2025) 20:18
https://doi.org/10.1186/s13015-025-00283-9

Algorithms for Molecular Biology

RESEARCH

Open Access

Finding maximum common contractions between phylogenetic networks

Bertrand Marchand[1*], Nadia Tahiri[1], Shohreh Golpaigani Fard[2], Olivier Tremblay-Savard[2] and Manuel Lafond[1]

Abstract

In this paper, we lay the groundwork on the comparison of phylogenetic networks based on edge contractions and expansions as edit operations, as originally proposed by Robinson and Foulds to compare trees. We prove that these operations connect the space of all phylogenetic networks on the same set of leaves, even if we forbid contractions that create cycles. This allows to define an operational distance on this space, as the minimum number of contractions and expansions required to transform one network into another. We highlight the difference between this distance and the computation of the *maximum common contraction* between two networks. Given its ability to outline a common structure between them, which can provide valuable biological insights, we study the algorithmic aspects of the latter. We first prove that computing a maximum common contraction between two networks is NP-hard, even when the maximum degree, the size of the common contraction, or the number of leaves is bounded. We also provide lower bounds to the problem based on the Exponential-Time Hypothesis. Nonetheless, we do provide a polynomial-time algorithm for weakly galled trees, a generalization of galled trees.

Keywords Phylogenetic networks, Contractions, Algorithms, Weakly galled trees

Introduction

The reconstruction of evolutionary histories, and ultimately of a universal "Tree of Life" based on biological data is one of the core tasks of comparative genomics, and bioinformatics as a whole. However, due to events such as horizontal gene transfer [1, 2] and hybridization [3], evolutionary histories may not always be represented as trees. As a result, the concept of phylogenetic *networks* has emerged to represent evolution in its full generality, and has become a central topic in bioinformatics research [4]. Unlike trees, it allows for the presence of nodes with more than one parent, usually called *reticulations*. Reconstructing phylogenetic networks from data is a notoriously difficult problem, and a wide variety of methods have emerged for tackling it [5–9]. However, given the same data-set, different methods may not always yield the same result, which can be the source of heated debate in the community (see e.g., the exchanges on the reconstructions of COVID phylogenetic networks [10–13]).

This raises the question of the development of *metrics* on phylogenetic networks, in order to be able to compare different predictions, evaluate their accuracy against simulated or gold standard data-sets, and identify outliers or similarities among them. In the case of trees, one of the most established metrics is the Robinson-Foulds distance [14]. It is usually defined as the size of the symmetric difference of the sets of *clades* (i.e. sets of leaves descending from a single node) of the two input trees. Note however that its original definition in [14] presented it as the minimum number of edge contractions and expansions required to edit one tree into the other. The two definitions are equivalent, as each edge contraction (resp. expansion) suppresses (resp. introduces) exactly one

*Correspondence:
Bertrand Marchand
bertrand.marchand@usherbrooke.ca
[1] Departement d'Informatique, University of Sherbrooke, 2500 Boulevard de l'Universite, J1K2R1 Sherbrooke, QC, Canada
[2] Department of Computer Science, University of Manitoba, 75 Chancellors Cir, R3T5V6 Winnipeg, MB, Canada

clade in a tree. In addition, it is also shown in [14] that one can do all contractions, followed by all expansions, without loss of optimality. Therefore, its computation also yields a *maximum common contraction* of the two trees, as evidence of their common structure.

The situation is not as simple in the case of networks. To start with, for networks of unbounded degree (i.e. allowing for an unbounded number of ancestors and descendants per node), even deciding whether two networks are identical is Graph Isomorphism-complete [15]. Nonetheless, several different metrics have been developed for sub-classes of networks. Some of them generalize the clade-based definition of the Robinson-Foulds metric, such as *hardwired cluster* [4, 16], *softwired cluster* [4] and *tripartition* [16, 17] distances. There exist however even binary networks that are distinct while exhibiting the same sets of clusters [4, Figure 6.28]. Likewise, there are pairs of distinct binary networks for which the tripartition distance is 0 [18]. In addition, in the case of the softwired distance, there can be an exponential number of clusters to compare, making a polynomial-time algorithm unlikely. The same problem arises when comparing networks in terms of the trees they display [4, Section 6.14.3]. Another group of proposals for metrics on phylogenetic networks includes the μ-distance [19, 20], as well as the *nodal distance* and *triplet distance* [21]. One could loosely describe them as "connectivity distances", in the sense that they consist in looking at the *paths* connecting leaves with other leaves or internal nodes in the network. While they are polynomial to compute, they are only valid on sub-classes of networks, namely *orchard networks* [20] and *time-consistent tree-child networks* [21]. A notable exception is the *subnetwork distance* [4, Section 6.14.4], defined as the symmetric difference of the sets of *rooted subnetworks*, which is valid on all networks, and can be computed in polynomial-time for bounded-degree networks.

However, contrary to the Robinson-Fould metric on trees, none of the measures mentioned above allow to outline a single *common sub-structure* in input networks (at least not directly). A natural way to obtain metrics valid on the entire space of phylogenetics is to use *operational definitions*, i.e. to define a distance as the minimum number of a set of "edition operations" needed to transform one network into another. Note that, as mentioned earlier, the original definition of the Robinson-Foulds distance falls into this category. On networks, examples of such operations include *nearest-neighbor interchange* (NNI, [22]), *subtree prune and regraph* (SPR, [23]) or *cherry-picking operations* [24] on orchard networks. While cherry-picking operations have been successfully used to define and compute metrics on orchard networks [25], computing distances based on NNI and SPR operations is NP-hard even for trees [26, 27]. A remarkable aspect of operational distances is also that, if some operations "reduce" the network (as the edge contraction for Robinson-Foulds) and others "expand" it (as the edge expansion for Robinson-Foulds) then a natural notion of "maximum common reduced network" emerges as a way to compare networks. We see it as the best possible case of "exhibiting common structure" using a metric. Interestingly, both cherry-picking operations [25] and SPR [28] allow to define such a maximum common structure. Another example of this philosophy is the computation of *maximum agreement sub-networks* [29], although their definition is not operational.

In this work, we generalize the original definition of Robinson-Foulds, by defining and studying an operational metric for phylogenetic networks based on edge contractions and expansions. We first prove the connectivity of the space of networks having the same leaf sets under these operations. Then, we use them to define two possible ways of comparing networks: (1) through the minimum number of contractions and expansions connecting two networks and (2) based on the size of a maximum common contraction. We show that the former defines a metric, and the latter a *semi-metric*, i.e. a dissimilarity measure verifying all properties of a distance except the triangle inequality. Given our purpose to outline common structures between networks, we focus on the computation of a maximum common contraction (MCC) semi-metric. To do so, we first provide a characterization of contracted networks in terms of a *witness structure*, as well as a result regarding the *diameter* of the maximum common contraction semi-metric. We also prove its NP-hardness under different conditions: when restricted to bounded-degree networks and a common contraction of size ≤ 3 (number of non-leaf nodes), and when one of the networks has only 5 leaves and the other has unbounded degree. However, we also provide a polynomial-time algorithm for the case of weakly galled trees.

Related works outside of phylogenetics Perhaps the closest related work in its philosophy is [30], which defines a distance between (isomorphism classes of) undirected, unlabeled graphs. However, no algorithmic results are given. There does exist a rich line of work in algorithmic graph theory on contraction problems in undirected, unlabeled graphs [31–35]. The typical problem of interest in this literature is that of deciding whether a graph H is the contraction of a graph G (either with H fixed or not), not computing a common contraction between them. To the knowledge of the authors, no attention has been given to this problem so far from an algorithmic perspective. This is surprising in the light of the fact that the *maximum common subgraph* problem, which is similar in philosophy (but completely different

in its definition), has been the subject of some work [36, 37]. We also highlight that we forbid contractions that create directed cycles, which differs from most previous work. We also assume that leaves are uniquely labeled in the networks. This is an important difference, as deciding whether an unlabeled tree is the contraction of another unlabeled tree is NP-hard [35], whereas this is polynomial-time solvable when leaves are labeled, as this is equivalent to computing the Robinson-Foulds distance.

Contraction-expansion distance

Preliminaries

This article uses standard directed graph terminology (nodes, edges, in-neighbors, out-neighbors, acyclic graphs, ...), as can for instance be found in [38]. In particular, we use the notion of DAG (directed acyclic graph) and topological ordering. A *topological ordering* of a DAG $\mathcal{N}$ is a permutation $\sigma = (u_1, \ldots, u_n)$ of $V(\mathcal{N})$ such that $(u_i, u_j) \in E(\mathcal{N})$ implies $i < j$. Hence, no backwards arc and thus backwards path is possible. Such an ordering always exists in a DAG, and conversely, a directed graph allowing for a topological ordering is a DAG. We write $[n]$ for the set of integers going from 1 to n. A *phylogenetic network* $\mathcal{N} = (V, E)$ is a directed acyclic graph (DAG) such that exactly one node (the *root*) has no in-neighbors, and nodes with no out-neighbor (the *leaves*) have in-degree 1. We may write *network* for short. A non-leaf node is called an *internal node*. We use $V(\mathcal{N}), E(\mathcal{N}), I(\mathcal{N})$, and $L(\mathcal{N})$ to denote, respectively, the set of nodes, edges, internal nodes, and leaves of $\mathcal{N}$. If there is an edge from u to v, we may write either (u, v) or $u \to v$. A *reticulation* is a node of in-degree two or more. In a network, an in-neighbor of a node is called a *parent*, and an out-neighbor is a *child*. The set of parents and children of a node $u \in V(\mathcal{N})$ are respectively denoted by $\Gamma^-_{\mathcal{N}}(u)$ and $\Gamma^+_{\mathcal{N}}(u)$. Note that a node may have multiple parents, and that we allow a root with a single child, as well as nodes with a single parent and a single child. We say that u *reaches* a leaf ℓ if there exists a directed path from u to ℓ. The set of leaves that u reaches in $\mathcal{N}$ is denoted $D_{\mathcal{N}}(u)$ (or $D(u)$ for short). If $\mathcal{N}$ is a tree, such a set $D(u)$ is sometimes called a *clade* or *cluster*.

To continue, developing metrics between networks requires in particular a notion of *equality* between them. To that end, we use the notion of graph isomorphism, augmented with leaf preservation. Specifically, two networks $\mathcal{N}_1 = (V_1, E_1)$ and $\mathcal{N}_2 = (V_2, E_2)$ are said *isomorphic*, denoted $\mathcal{N}_1 \sim \mathcal{N}_2$, if there exists a bijection $\phi : V_1 \to V_2$ such that (a) $\forall u, v \in V_1$, $(u, v) \in E_1$ if and only if $(\phi(u), \phi(v)) \in E_2$ and (b) $\forall u \in L(\mathcal{N}_1), \phi(u) = u$. Given this notion, we recall the definition of a *metric* (or *distance*) on phylogenetic networks. A function d, associating a real value to an arbitrary pair of input networks, is a *metric* if $\forall \mathcal{N}_1, \mathcal{N}_2$: (positivity) $d(\mathcal{N}_1, \mathcal{N}_2) \geq 0$; (identity) $d(\mathcal{N}_1, \mathcal{N}_2) = 0$ if and only if $\mathcal{N}_1 \sim \mathcal{N}_2$; (symmetry) $d(\mathcal{N}_1, \mathcal{N}_2) = d(\mathcal{N}_2, \mathcal{N}_1)$; and (triangle inequality) $\forall \mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3\, d(\mathcal{N}_1, \mathcal{N}_3) \leq d(\mathcal{N}_1, \mathcal{N}_2) + d(\mathcal{N}_2, \mathcal{N}_3)$.

Remark 1 (graph isomorphism) Note that graph isomorphism is polynomial-time solvable on *bounded-degree* graphs [39], and therefore on bounded-degree networks as well.[1]

Note that it is often the identity criteria which has limited the applicability of metrics to sub-families of networks. For instance, [4, Figure 6.28] gives an example of a pair of networks that are different, while their *hardwired cluster*, *softwired cluster* and *tree containment* distances are 0. A similar example is given in [19] for the μ -distance.

Contractions and expansions

We now define edge contractions and expansions, as illustrated in Fig. 1, and the two ways we compare networks using them.

Definition 1 (Contraction) Let $\mathcal{N} = (V, E)$ be a network and let $(u, v) \in E$. The contraction operation $c(u, v, w)$ on $\mathcal{N}$, where $w \notin V$, yields the directed graph $\mathcal{N}' = (V', E')$ such that:

1. $V' = (V \setminus \{u, v\}) \cup \{w\}$;
2. $\Gamma^-_{\mathcal{N}'}(w) = \Gamma^-_{\mathcal{N}}(u) \cup \left(\Gamma^-_{\mathcal{N}}(v) \setminus \{u\}\right)$; and
3. $\Gamma^+_{\mathcal{N}'}(w) = \left(\Gamma^+_{\mathcal{N}}(u) \setminus \{v\}\right) \cup \Gamma^+_{\mathcal{N}}(v)$.

We denote by $\mathcal{N}/(u, v)$ the network obtained after applying the contraction $c(u, v, w)$ on $\mathcal{N}$.

A contraction $c(u, v, w)$ on a network $\mathcal{N}$ is said *admissible* if $\mathcal{N}/(u, v)$ is also a network, on the same set of leaves. This means that v is not a leaf, and that the contraction does not create a cycle. In particular, no edge adjacent to a leaf is ever contracted. The following proposition characterizes the set of admissible contractions in a network. This result can also be found in [40, Lemma 3].

Proposition 1 *A contraction c(u, v, w) applied to $\mathcal{N}$ creates a cycle if and only if there exists a directed path from u to v that does not use edge $u \to v$.*

[1] Isomorphism on networks is contained in the isomorphism problem on directed graphs, as each leaf may be replaced by a unique subgraph, implementing the same mapping constraint.

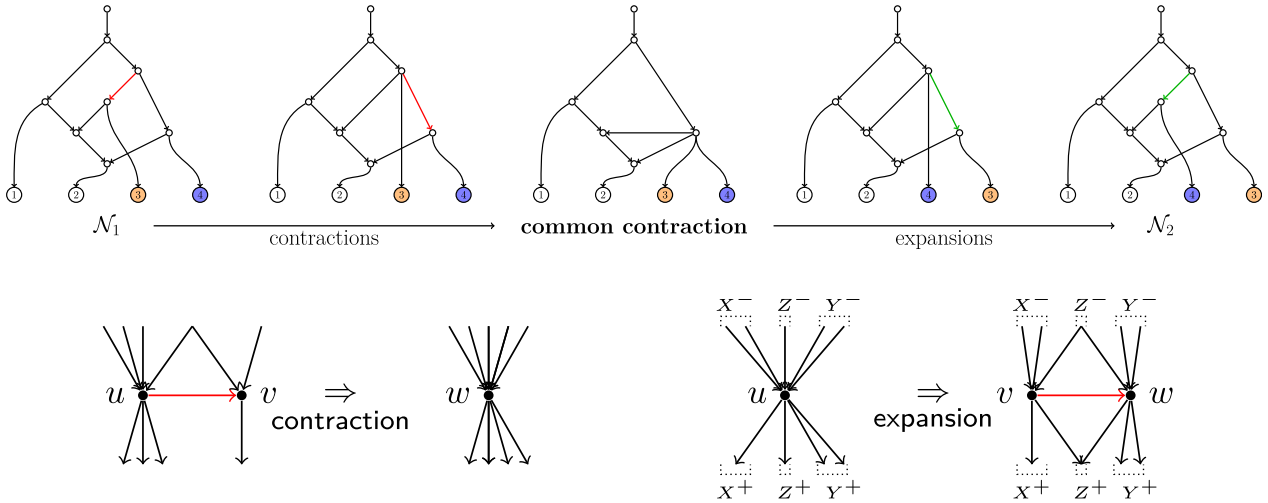

Fig. 1 (top) The transformation of a network into another through contractions, followed by expansions. Midway, the intermediate network is a maximum common contraction, which can be achieved from $\mathcal{N}_2$ by reversing the expansions. (bottom) Illustration of edge contractions (left) and expansions (right). For expansions, the sets $X^-, Y^-, Z^-, X^+, Y^+, Z^+$ specify how the neighbors of u are distributed to v and w. Note that contractions may delete cycles and expansions create them

Proof

If there is such a path $u = p_1 \to \cdots \to p_\ell = v$ in $\mathcal{N}$ (with $\ell \geq 3$), then contracting edge $u \to v$ into a newly introduced node w (i.e. operation $c(u, v, w)$) creates the cycle $p_1 = w \to \cdots \to p_\ell = w$. If there is no such path, we show that there is a topological order of $\mathcal{N}$ in which u, v are adjacent (in that order). Within such an order, by inserting w at the position of u, v and removing them, we get a valid topological order for $\mathcal{N}'$, the graph obtained by applying $c(u, v, w)$ to $\mathcal{N}$, which is therefore acyclic.

Let us therefore start from a topological order σ of $\mathcal{N}$, and let us denote by S the nodes of $\mathcal{N}$ ordered between u and v in $\mathcal{N}$. We show that S may always be reduced in size by a slight re-shuffling of σ, showing that it may be empty.

To start with, if no node in S is an out-neighbor of u, then u may be moved to make it adjacent to v while keeping the topological order valid. If this is not the case, then let x be an out-neighbor of u in S. By assumption (no path between u and v not using $u \to v$), there may not be a path between x and v. Consider the largest node y in S (according to the topological ordering) such that there is no path from y to v in $\mathcal{N}$. Note that such a y must exist since, in particular, x is in S and does not reach v (we may have $x = y$). If y is the last node of S according to σ, then y may be moved after v while maintaining a topological ordering, because there is no $y \to v$ arc. Otherwise, let $z \neq y$ be a node after y in S. By the definition of y, z must have a path to v. There may therefore not be an edge from y to such a z, as it would yield a path from y to v. Because this is true for every node z of S that occurs after y, in this case as well, y may safely be moved after v in the topological ordering. In both cases, this reduces S by 1. This process may be repeated until S is empty. □

A network $\mathcal{M}$ is said to be a *contraction* of a network $\mathcal{N}$ if there exists a series of admissible contractions yielding, when applied on $\mathcal{N}$, a network isomorphic to $\mathcal{M}$. Also, $\mathcal{M}$ is a *common contraction* of two networks $\mathcal{N}_1$ and $\mathcal{N}_2$ if it is both a contraction of $\mathcal{N}_1$ and $\mathcal{N}_2$.

As for expansions, they essentially transform a node into an edge. The definition is slightly more involved, as we must specify how the in and out-neighbors are split. Note that expansions may simply be seen as the inverse operation of contractions.

Definition 2 (Expansion) Let $\mathcal{N} = (V, E)$ be a network and let $u \in V$. The expansion operation $e(u, v, w, X^-, Y^-, Z^-, X^+, Y^+, Z^+)$, such that $\Gamma^-_{\mathcal{N}}(u) = X^- \cup Y^- \cup Z^-$, every node of $\Gamma^-_{\mathcal{N}}(u)$ occurs in exactly one of X^-, Y^-, Z^-, $\Gamma^+_{\mathcal{N}}(u) = X^+ \cup Y^+ \cup Z^+$, and every node of $\Gamma^+_{\mathcal{N}}(u)$ occurs in exactly one of X^+, Y^+, Z^+, yields the network $\mathcal{N}' = (V', E')$ such that:

1. $V' = V \setminus \{u\} \cup \{v, w\}$, where v, w are two new nodes;
2. $\Gamma^-_{\mathcal{N}'}(v) = X^- \cup Z^-$, $\Gamma^-_{\mathcal{N}'}(w) = Y^- \cup Z^- \cup \{v\}$, $\Gamma^+_{\mathcal{N}'}(v) = X^+ \cup Z^+ \cup \{w\}$ and $\Gamma^+_{\mathcal{N}'}(w) = Y^+ \cup Z^+$.

The sets $X^-, Y^-, Z^-, X^+, Y^+, Z^+$ specify how the neighbors of the original node u are distributed among the newly created ones v and w. Nodes in X^-/X^+ are attributed to v only, those in Y^-/Y^+ to w only, and those in Z^+/Z^- to both. Note that, compared to the definition of these operations in the case of trees [14], we need to specify more information as to how the neighbors of the original node u are "split" between v and w. It is quite straightforward to see that, if an expansion replaces u with (v, w), then the contraction $c(v, w, u)$ reverses it. Conversely, a contraction $c(u, v, w)$ that yields node w can be reversed with the expansion that specifies the X, Y, Z parameters as the appropriate sets of previous in and out-neighbors of u and v. As for contractions, we define *admissible expansions* on a network $\mathcal{N}$ on L leaves as those that yield a phylogenetic network on L leaves after application (no creation of a second root, of new leaves, or cycles).

Based on these definitions, we define two ways of comparing phylogenetic networks. The first one is the *contraction-expansion distance*, defined as the least number of contractions and expansions required to transform one network into another. The second one is a dissimilarity measure based on the *maximum common contraction* of two networks (or more precisely, the distance to a common contraction). Although we will see that the latter does not verify the triangle inequality, and therefore does not qualify as a "metric", we still make it the main subject matter of this article, given the common structure it outlines between two networks.

Definition 3 (contraction-expansion distance) Given two networks $\mathcal{N}_1$ and $\mathcal{N}_2$ over the same leaf sets, the contraction-expansion distance $d_{\text{CE}}(\mathcal{N}_1, \mathcal{N}_2)$ is the minimum length of a sequence of admissible contractions and expansions transforming $\mathcal{N}_1$ into a network $\mathcal{N}_2'$ such that $\mathcal{N}_2' \sim \mathcal{N}_2$.

As for the dissimilarity measure based on maximum common contraction, we call it δ_{MCC}, emphasizing that it is not a distance by not using d.

Definition 4 (MCC dissimilarity measure) Given two phylogenetic networks $\mathcal{N}_1$ and $\mathcal{N}_2$ over the same leaf sets, the maximum common contraction (MCC) dissimilarity measure $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2)$ is defined as $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) = |I(\mathcal{N}_1)| + |I(\mathcal{N}_2)| - 2|I(\mathcal{M})|$ where $\mathcal{M}$ is a common contraction of $\mathcal{N}_1, \mathcal{N}_2$ of maximum size.

Remark 2 As a common contraction also provides a sequence of contractions and expansions connecting both networks (by inverting one of the list of contractions into expansions), we have $d_{\text{CE}}(\mathcal{N}_1, \mathcal{N}_2) \leq \delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2)$.

Our main algorithmic problem of interest, formulated as a decision problem, is the following: given a pair of networks $\mathcal{N}_1$ and $\mathcal{N}_2$ on the same leafset and an integer k, is there a common contraction of size larger than k?

Maximum Common Network Contraction (MCNC)

Input: Networks $\mathcal{N}_1, \mathcal{N}_2$, integer k

Question: Is there a common contraction of $\mathcal{N}_1$ and $\mathcal{N}_2$ with more than k internal nodes?

Note that it is equivalent to asking whether $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) \leq k'$, with $k' = |I(\mathcal{N}_1)| + |I(\mathcal{N}_2)| - 2k$. We next establish important properties of d_{CE} and δ_{MCC}.

Properties of d_{CE} and δ_{MCC}

We first establish that admissible contractions and expansions connect the set of all phylogenetic networks on the same set of leaves. Within it, the *star network* denotes the network consisting of a single non-leaf node to which all leaves are connected.

Proposition 2 *Any phylogenetic network $\mathcal{N}$ may be contracted into the star network by a series of admissible contractions.*

Proof

We argue that the contraction of the edge between the root r of $\mathcal{N}$ and its first non-leaf out-neighbor in any topological ordering is always admissible. Let σ be any topological ordering of $V(\mathcal{N})$, and let u be the first non-leaf out-neighbor of the root r that occurs in σ (if no such out-neighbor exists, then $\mathcal{N}$ is already a star). If there exists a directed path between r and u that does not use edge $r \to u$, then any node on this path must be between r and u in σ, which goes against the definition of u. Therefore, by Proposition 1, $r \to u$ is admissible. It follows that we can always find a contraction of an edge incident to the root, and thus $\mathcal{N}$ can then be contracted into the star network by picking such an edge as long as there are non-root internal nodes. □

The corollary below uses Proposition 2 and the star network to show that one can always transform a network into another.

Corollary 3 *Given any two networks $\mathcal{N}_1$ and $\mathcal{N}_2$ over the same leafset, there always exists a common contraction, and there always exists a series of admissible contractions and expansions that transforms $\mathcal{N}_1$ into $\mathcal{N}_2' \sim \mathcal{N}_2$.*

Proof

Through Proposition 2, let C_1 and C_2 two sequences of contraction operations transforming $\mathcal{N}_1$ and $\mathcal{N}_2$ into two isomorphic star networks on the same set of leaves as $\mathcal{N}_1$ and $\mathcal{N}_2$. This shows that a common contraction exists. Moreover, by reversing C_2 into a sequence of expansions, appending it to C_1, and tracking the isomorphism ϕ along the resulting sequence of contractions and expansions, we get a sequence of admissible operations transforming $\mathcal{N}_1$ into $\mathcal{N}_2' \sim \mathcal{N}_2$. □

Note that in the proof above, the length of the sequence is $|I(\mathcal{N}_1)| + |I(\mathcal{N}_2)| - 2$, from which we deduce the following remark.

Remark 3 Given two networks $\mathcal{N}_1$ and $\mathcal{N}_2$, $|I(\mathcal{N}_1)| + |I(\mathcal{N}_2)| - 2$ is an upper bound on the distance.

We will see further down in this section that this value is in fact (under a varied set of constraints) the *diameter* of δ_{MCC} over the set of networks with fixed numbers of internal nodes.

It can then be shown that d_{CE} is a metric. Due to its "operational" definition, it is relatively straightforward. In particular, the triangle inequality is satisfied, since given three networks $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$, we can always transform $\mathcal{N}_1$ into $\mathcal{N}_2$, then $\mathcal{N}_2$ into $\mathcal{N}_3$.

Proposition 4 *d_{CE} is a metric on the set of phylogenetic networks with the same leaf sets.*

Proof

Positivity. This is trivially met since one cannot make a negative amount of operations. *Identity.* By definition, if $\mathcal{N}_1$ and $\mathcal{N}_2$ are isomorphic, $d_{\text{CE}}(\mathcal{N}_1, \mathcal{N}_2) = 0$. Conversely, if $\mathcal{N}_1$ and $\mathcal{N}_2$ are not isomorphic, then at least one operation is required to modify $\mathcal{N}_1$ into a network isomorphic to $\mathcal{N}_2$, and $d_{\text{CE}}(\mathcal{N}_1, \mathcal{N}_2) > 0$.

Symmetry. Let Σ be a sequence of admissible contractions and expansions of minimum length transforming $\mathcal{N}_1$ into $\mathcal{N}_2' \sim \mathcal{N}_2$. By inverting Σ into Σ' and tracking the isomorphism between $\mathcal{N}_2'$ and $\mathcal{N}_2$ along Σ', we get a sequence of admissible operations of length $|\Sigma|$ transforming $\mathcal{N}_2$ into a network $\mathcal{N}_1'$ isomorphic to $\mathcal{N}_1$.

Triangle inequality. Let $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$ be three networks on the same leafset. Since any of these networks can be transformed into the other by Corollary 3, one can transform $\mathcal{N}_1$ into $\mathcal{N}_2$, then transform $\mathcal{N}_2$ into $\mathcal{N}_3$, which justifies $d_{\text{CE}}(\mathcal{N}_1, \mathcal{N}_3) \leq d_{\text{CE}}(\mathcal{N}_1, \mathcal{N}_2) + d_{\text{CE}}(\mathcal{N}_2, \mathcal{N}_3)$. □

As for δ_{MCC}, it does verify identity ($\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) = 0$ if and only if their common contraction is themselves, i.e. they are isomorphic), positivity, and symmetry (by definition). However, Fig. 2A shows an example in which $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_3) = 10$, while $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) + \delta(\mathcal{N}_2, \mathcal{N}_3) = 4 + 2 = 6$, i.e. not verifying the triangle inequality. Therefore, δ_{MCC} is a *semi-metric*. We also highlight the difference between d_{CE} and δ_{MCC} by showing on Fig. 2B an example of two networks for which $d_{\text{CE}}(\mathcal{N}_1, \mathcal{N}_2) = 2$, whereas $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) = 6$.

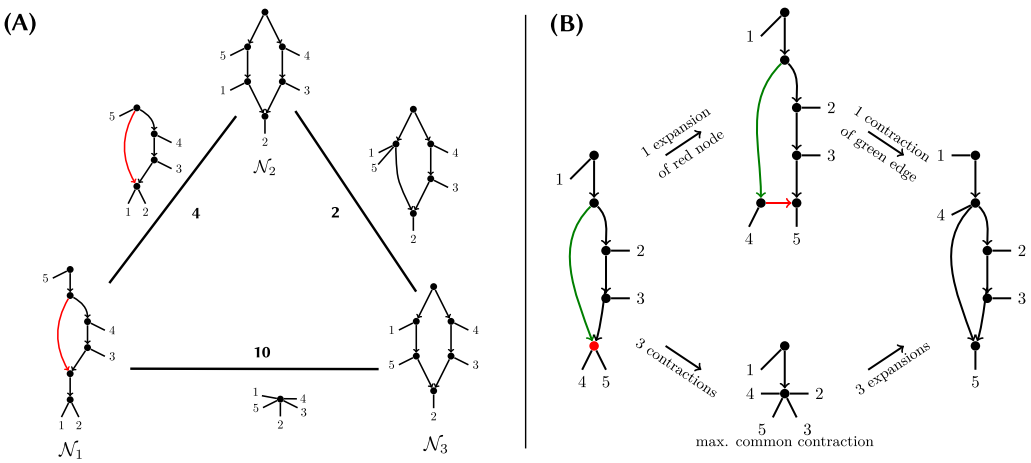

Fig. 2 **A** Examples of 3 networks for which $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_3) > \delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) + \delta_{\text{MCC}}(\mathcal{N}_2, \mathcal{N}_3)$. Indeed, as shown on the figure, there is a common contraction of size 4 between $\mathcal{N}_1$ and $\mathcal{N}_2$, and of size 5 between $\mathcal{N}_2$ and $\mathcal{N}_3$. However, the only possible common contraction between $\mathcal{N}_1$ and $\mathcal{N}_3$ is the star network. This is due to the edge highlighted in $\mathcal{N}_1$ not being admissible, enforcing the contraction of the whole cycle. Since all networks have 6 internal nodes, we have $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_3) = 12 - 2 = 10$ whereas $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) = 12 - 8 = 4$ and $\delta_{\text{MCC}}(\mathcal{N}_2, \mathcal{N}_3) = 12 - 10 = 2$. **B** An example further highlighting the difference between d_{CE} and δ_{MCC}, as we have there two networks $\mathcal{N}_1, \mathcal{N}_2$ such that $d_{\text{CE}}(\mathcal{N}_1, \mathcal{N}_2) = 2$ (following the top path) whereas $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) = 6$ (following the bottom path)

Remark 4 If we allow contractions that create cycles, then we could show that expansions and contractions can commute. This would imply that contractions can always be carried out before expansions, and therefore $d_{\mathrm{CE}} = \delta_{\mathrm{MCC}}$. The fact that δ_{MCC} is a distance on unlabeled, undirected graphs with the same number of nodes has also been shown [30].

Witness structure to a contraction

So far, we have only looked at contractions from an operational point of view, in which a network is gradually changed into its contraction. For the purpose of easing formal proofs in the following sections, we borrow from the undirected graph contraction literature (see for instance [34]) the concept of *witness structure* of a contraction. The main idea is that, for a network $\mathcal{N}$ with contraction $\mathcal{M}$, a node v of $\mathcal{M}$ corresponds to a connected set of nodes in $\mathcal{N}$, that we call W_v. The intuition is that the nodes of W_v are contracted together to become v. Contractions impose W_v to be connected if we ignore edge directions (i.e. weakly connected), and the sets of nodes $\{W_v\}_{v \in I(\mathcal{M})}$ must collectively partition $V(\mathcal{N})$.

Definition 5 Given two phylogenetic networks $\mathcal{N}$ and $\mathcal{M}$ on the same set of leaves, we call an *$\mathcal{M}$-witness structure in $\mathcal{N}$* any partition $\mathcal{W} = \{W_u \mid u \in I(\mathcal{M})\}$ of the internal nodes of $\mathcal{N}$ such that:

(1) $\forall u \in I(\mathcal{M})$, W_u is a non-empty weakly connected sub-graph of $\mathcal{N}$.
(2) $\forall u, v \in I(\mathcal{M}) \times I(\mathcal{M})$, $(u, v) \in E(\mathcal{M})$ iff $\exists x \in W_u, y \in W_v$ such that $(x, y) \in E(\mathcal{N})$.
(3) $\forall u \in L(\mathcal{M})$, the parent p_u of u in $\mathcal{N}$ must be in $W_{p'_u}$, where p'_u is the parent of u in $\mathcal{M}$.

Note that as $I(\mathcal{N})$ and $I(\mathcal{M})$ are the sets of internal nodes, conditions (1) and (2) do not enforce an agreement on the placement of the leaves between the contraction of $\mathcal{N}$ and $\mathcal{M}$. To obtain the equivalence between the presence of an $\mathcal{M}$-witness structure and the contractibility of $\mathcal{N}$ into $\mathcal{M}$, condition (3) is therefore needed.

In the case of undirected graphs, the equivalence between the presence of an $\mathcal{M}$-witness structure and the contractibility into $\mathcal{M}$ is trivial. In our case though, we need to guarantee that, although the connectivity of the sets in a witness structure is only required in a weak sense (i.e. ignoring directions), the presence of a $\mathcal{M}$ -witness structure in $\mathcal{N}$ is still equivalent to $\mathcal{M}$ being a contraction of $\mathcal{N}$, with all intermediary networks being acyclic. This can be proved using similar ideas as in Proposition 2.

Proposition 5 *Let $\mathcal{M}$ and $\mathcal{N}$ two networks over the same set of leaves. Then $\mathcal{M}$ is a contraction of $\mathcal{N}$ if an only if there exists a $\mathcal{M}$-witness structure in $\mathcal{N}$.*

Proof

From contraction sequence to $\mathcal{W}$. We work by induction on the length of the contraction sequence. For an empty contraction sequence from $\mathcal{N}$ to $\mathcal{M}$, $\mathcal{N}$ and $\mathcal{M}$ are isomorphic. An $\mathcal{M}$-witness structure in $\mathcal{N}$ is given by $W_u = \{\phi(u)\}$ for all u internal node of $\mathcal{M}$, and ϕ isomorphism from $\mathcal{M}$ to $\mathcal{N}$.

Let now $C = c_1, \ldots, c_p$ be a contraction sequence from $\mathcal{N}$ to $\mathcal{M}$. Let us call $\mathcal{M}'$ the network obtained by applying the first $p-1$ contractions to $\mathcal{N}$. By the induction hypothesis, there is a $\mathcal{M}'$-witness structure $\mathcal{W}'$ in $\mathcal{N}$. Let $c(x, y, z)$ be the p-th contraction. We remove W_x and W_y from $\mathcal{W}'$, and add to it a new set $W_z = W_x \cup W_y$. We call the result $\mathcal{W}$ and argue it is a $\mathcal{M}$-witness structure in $\mathcal{N}$. Let us verify each point from the definition. (1) is verified as W_x and W_y were both weakly connected, and are connected together by edge $x \to y$. (2) is inherited from $\mathcal{W}'$ for any pair of nodes not involving z. For u internal node of $\mathcal{M}$, $u \to z$ is an edge if and only if $u \to x$ or $u \to y$ were edges, which is equivalent to the existence of some edge from W_u to W_x or W_u to W_y, in turn equivalent to the existence of an edge between W_u and W_z. The case of an edge outgoing from z is treated similarly. As for (3), any leaf u whose parent is x or y in $\mathcal{M}'$ now has z has a parent, and indeed, denoting by $p_{\mathcal{N}}(u)$ the parent of u in $\mathcal{N}$, $p_{\mathcal{N}}(u) \in W_x$ or $p_{\mathcal{N}}(u) \in W_y$ implies $p_{\mathcal{N}}(u) \in W_z$.

From $\mathcal{W}$ to contraction sequence Given an $\mathcal{M}$-witness structure in $\mathcal{N}$ called $\mathcal{W}$, we prove that each set of $\mathcal{W}$ may be contracted into a single node. The result will then trivially be a network isomorphic to $\mathcal{M}$ (if p_u is the node obtained from contracting W_u then $\phi : p_u \to u$ is an isomorphism). We prove the result for each u by induction on $|W_u|$.

Consider therefore $W_u \in \mathcal{W}$, for u internal node of $\mathcal{M}$, and the sub-graph H_u induced by W_u in $\mathcal{N}$. We show that there is always an admissible contraction in H_u. Indeed, H_u must be acyclic since a cycle in H_u would be a cycle of $\mathcal{N}$. Consider the first node x in a topological ordering of H_u, and y its first neighbor (in H_u) according to the same topological ordering. Note that unless W_u is already a single node, such a y must exist, as otherwise x

has both no in-neighbor and no out-neighbor in H_u, and thus H_u is not weakly connected. By Proposition 1, the contraction of $x \to y$ is admissible if and only if there is no directed path from x to y not using edge $x \to y$. Such a path within H_u is not possible, as then y would not be the first neighbor of x in a topological ordering. Outside of H_u, such a path is not possible either: let $u_1, \ldots, u_t$ be, in order, the list of nodes in $\mathcal{M}$ whose corresponding set $(W_{u_1}, \ldots, W_{u_t})$ is visited on such a path. By equivalence between the presence of an edge between W_u and W_v and the presence of $u \to v$ in $\mathcal{M}$, $u, u_1, \ldots, u_t, u$ is a directed cycle of $\mathcal{M}$, which is not possible. Overall, there is always an admissible contraction in any set of a $\mathcal{M}$-witness structure. As the result is still a $\mathcal{M}$-witness structure, now in $\mathcal{N}/(x \to y)$, and with $|W_u|$ reduced by 1, W_u is contractible to a point by the induction hypothesis. We do the same for each W_u to conclude the proof. □

Diameter of δ_{MCC}

The maximum value a distance method can produce when comparing phylogenetic networks, called the diameter, is crucial to its practical use. Indeed, it enables the normalization of distance values, allowing to quantify how two networks are similar relative to a worst possible case. Specifically, given sets of networks F_1, F_2, we define the *diameter over* F_1, F_2 for δ_{MCC} as

$$\mathrm{diam}_{F_1,F_2} = \max_{\mathcal{N}_1,\mathcal{N}_2 \in F_1 \times F_2} \delta_{\mathrm{MCC}}(\mathcal{N}_1, \mathcal{N}_2)$$

A natural choice for F_1, F_2 could be the set of all networks over a given number of leaves. However, as we shall see, this yields a value of $+\infty$. To avoid it, we look at the diameter obtained when restricting F_1 and F_2 to having fixed (but possibly different) numbers of internal nodes m and m' (equivalently, in binary networks, fixed numbers of reticulations).

In this setting, we show below that diam_{F_1,F_2} is $m + m' - 2$, corresponding to the case where the common contraction between both networks is the *star network*, as defined in Sect. 2.3. Note that this value matches the upper bound outlined in Remark 3. In other words, even when restricting $\mathcal{N}_1$ and $\mathcal{N}_2$ to having fixed numbers of nodes, it is still possible to find a pair meeting this upper bound. To prove the result, we will need the following Lemma:

Lemma 6 *If a network $\mathcal{M}$ is a contraction of a network $\mathcal{N}$, with witness structure $\mathcal{W} = \{W_u \mid u \in \mathcal{M}\}$, then the root of $\mathcal{N}$ must be in the witness set W_r associated to the root r of $\mathcal{M}$.*

Proof

First, note that W_r is non-empty by definition of a witness structure. Suppose the Lemma is wrong, and consider the first node x of W_r in a topological ordering of $\mathcal{N}$. By assumption, x is not the root. It must therefore have an in-neighbor, which cannot be in W_r (it would be before x in the topological ordering otherwise). By definition of a witness structure, this means r has an in-neighbor in $\mathcal{M}$, which cannot be the case, hence a contradiction. □

Proposition 7 Let ℓ, m, m' be integer numbers with $\ell > 3$, m, $m' > 1$. Then there are networks $\mathcal{N}, \mathcal{N}'$ with ℓ leaves such that $|I(\mathcal{N})| = m, |I(\mathcal{N}')| = m'$, and $\delta_{MCC}(\mathcal{N}, \mathcal{N}') = m + m' - 2$.

Proof

The structure of the proof is as follows: We first define two binary trees T and T' with ℓ leaves, and show that their maximum common contraction is the star network. Then, we describe how to modify T and T' so that their numbers of internal nodes become respectively m and m', while their maximum common contraction remains the star network.

Consider the caterpillar trees T, T' in Fig. 3(left). Specifically, T consists of a directed path $P = p_1, \ldots, p_{\ell-1}$ with $\ell - 1$ nodes, in which $\forall i \in [\ell - 1]$, p_i is the parent of leaf i, and $p_{\ell-1}$ is also the parent of leaf ℓ. As for T', it is built similarly (we call the path P'), except that leaves 1 and ℓ are swapped. We argue first that their maximum common contraction M is a star network. In a second time, we will describe how to modify them to get networks $\mathcal{N}, \mathcal{N}'$ with numbers of internal nodes equal to m and m'.

In the case of T and T', Lemma 6 implies that in any common contraction, leaf 1 must be a child of the root (as it is the case in T), as well as leaf ℓ (from T'). Therefore, in both T and T', the parents of leaves 1 and ℓ are both in the same witness set (associated to the root of the contraction). As this witness set must be connected, and there is only one path, consisting of all internal nodes, connecting the parents of leaves 1 and ℓ (both in T and T'), the witness set of the root contains all internal nodes. The maximum common contraction of T and T' is therefore the star network.

Now, we want to construct $\mathcal{N}$ and $\mathcal{N}'$ by modifying T and T' in order to reach numbers of internal nodes m and m'. We do so by either adding reticulated edges or

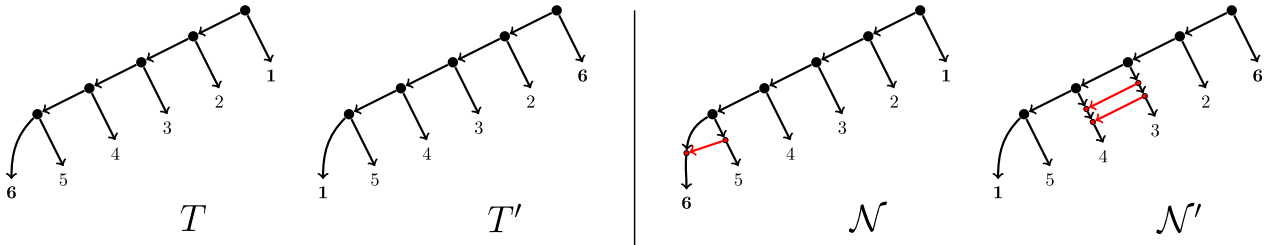

Fig. 3 (left) Two caterpillar trees over 6 leaves whose common contraction is the star network. One way of seeing it is that in T, leaf 1 is a child of the root, and therefore this must be the case in a contraction of T as well. In T', the only way to achieve this is to contract the parent of leaf 1 into the root, making the star network the only possible common contraction. (right) Reticulation edges have been added in T, T' between different pairs of leaves ($P_{\mathcal{N}} = \{(5,6)\}$ and $P_{\mathcal{N}'} = \{(3,4)\}$ in the notations of the proof), in order to get $\mathcal{N}, \mathcal{N}'$ with respectively 7 and 9 internal nodes, while maintaining the property that the star network is the only possible common contraction. Note that reaching numbers of internal nodes lower than 5 is also possible, by contracting a edges in T, T'

contracting some edges, in a way that the maximum common contraction remains the star network (see Fig. 3 (right)). To be more specific, we distinguish several cases for the construction of $\mathcal{N}$ from T and $\mathcal{N}'$ from T'.

If $m = \ell - 1$, then $\mathcal{N}$ is simply T. Likewise, if $m' = \ell - 1$, then $\mathcal{N}' = T'$. If $m < \ell - 1$, then $\mathcal{N}$ is obtained from T by contracting the last $\ell - 1 - m$ edges of P. Likewise, if $m' < \ell - 1$, $\mathcal{N}'$ is obtained from T' by contracting the last $\ell - 1 - m'$ edges of P'. If $m > \ell - 1$, and the difference is an even number, then $\mathcal{N}$ is constructed from T by taking the leaves $(\ell - 1, \ell)$ (their parent is $p_{\ell-1}$), and adding "reticulation edges" between them. Specifically, starting from T, the edges $(p_{\ell-1}, \ell - 1)$ and $(p_{\ell-1}, \ell)$ are removed. Instead, two paths $P_{\ell-1}$ and P_ℓ starting both at $p_{\ell-1}$, and ending at $\ell - 1$ and ℓ, are introduced. Without counting $p_{\ell-1}$, ℓ and $\ell - 1$, P_ℓ and $P_{\ell-1}$ consist of $(m - (\ell - 1))/2$ nodes. We add a (reticulation) edge from the i-th node of $P_{\ell-1}$ to the i-th node of P_l, for $i \in [(m - (\ell - 1))/2]$. This yields a network with $\ell - 1 + 2(m - (\ell - 1))/2 = m$ nodes. If $m' > \ell - 1$, we build $\mathcal{N}'$ from T' similarly, except that we pick any edge (u, v) of the main path P' distinct from $(\ell - 1, \ell)$ (we may, as $\ell > 3$), and the paths P_u and P_v (of length $(m' - (\ell - 1))/2$) that are introduced respectively start at the parent of u and the parent of v. Finally, if $m > \ell - 1$ and the difference is odd, we build $\mathcal{N}$ as in the even case, except that one of the edges of P is contracted (any will do). If $m' > \ell - 1$ and the difference is odd, we also build $\mathcal{N}'$ as in the even case, and contract one edge of P'. In any case, we obtain networks $\mathcal{N}$ and $\mathcal{N}'$ on ℓ leaves with respectively m and m' internal nodes.

We now argue that in all cases, the star network is the only possible common contraction between $\mathcal{N}$ and $\mathcal{N}'$ (which implies $\delta_{\mathrm{MCC}}(\mathcal{N}, \mathcal{N}') = m + m' - 2$). To start with, note that in $\mathcal{N}'$, in all cases ($m' < \ell - 1$ or not) the parent of ℓ is the root.

Let $\mathcal{M}$ be a common contraction of $\mathcal{N}$ and $\mathcal{N}'$ and $\mathcal{W}$ the witness structure of the contractibility of $\mathcal{M}$ into $\mathcal{N}$. The witness set associated to the root of $\mathcal{M}$ in $\mathcal{N}$ is denoted W. In $\mathcal{N}$, both the root (by Lemma 6) and the parent of ℓ must be in W.

If $m \leq \ell - 1$, then $\mathcal{N}$ is a contraction of T, and the set of all internal nodes of $\mathcal{N}$ constitutes the only directed path starting at the root and finishing at the parent of ℓ. All of it must be included in W, yielding the star network as the only possible contraction.

If $m > \ell - 1$, then let us call x_ℓ the parent of ℓ in $\mathcal{N}$ (it is the penultimate node of P_ℓ). Note that any node of P_ℓ or $P_{\ell-1}$ is on some directed path from the root of $\mathcal{N}$ to x_ℓ. Indeed, consider the i-th nodes on the paths $P_{\ell-1}$ and P_ℓ: by taking the (potentially contracted on one edge) path to $p_{\ell-1}$ (starting point of P_ℓ and $P_{\ell-1}$) then going to the i-th node of $P_{\ell-1}$, taking the edge to the i-th node of P_ℓ, and going to x_ℓ, we have a directed path visiting them both.

To conclude, we argue that *any directed path* from the root of $\mathcal{N}$ to x_ℓ must lie entirely in W. Indeed, if it is not included in W, then some vertices in these paths are in witness sets associated to other vertices of $\mathcal{M}$, which yields a directed cycle starting and ending at the root in $\mathcal{M}$, hence a contradiction. As any node on the path from the root of $\mathcal{N}$ to $p_{\ell-1}$ is on every path from the root to x_ℓ, we have that all internal nodes of $\mathcal{N}$ are in W, and the only possible common contraction between $\mathcal{N}$ and $\mathcal{N}'$ is the star network. □

A consequence of Proposition 7 is that if F_1, F_2 are both simply defined as the set of all networks over a given

number of leaves, diam_{F_1,F_2} is $+\infty$ (indeed, if it was equal to a finite value M, then we could for instance choose $m = M$ and $m' = 2$ to exceed it). Specifying the number of internal nodes is therefore required to get a finite value.

In addition, note that in this proof, each reticulated edge that is introduced adds exactly one reticulation to a network. Therefore, the construction also allows to specify the number of reticulations of each networks while ensuring their common contraction is the star network. The reticulation edges could also be added between several disjoint pairs of leaves in each network, allowing to specify the *level* of both networks, as well as the number of reticulations up to a limit of $level \times \#ofleaves/2$ (as with such a construction, one may require 2 leaves per biconnected component). In all of these cases, having two networks whose common contraction is a star (and therefore $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) = |I(\mathcal{N}_1)| + |I(\mathcal{N}_2)| - 2$) is possible, showing the generality of this value as a diameter.

NP-hardness of finding maximum common contractions

In this section, we prove that the Maximum Common Network Contraction problem is NP-hard, through reductions from the Set Splitting problem (SP12 in [41]).

Set Splitting

Input: A set X, a family $\mathcal{A}$ of subsets of X

Question: Is there a bipartition X_1, X_2 of X such that $\forall S \in \mathcal{A}, S \cap X_1 \neq \emptyset$ and $S \cap X_2 \neq \emptyset$?

If $(X, \mathcal{A})$ is a yes-instance, we say it is *splittable*. We then call X_1, X_2 the *split* of $(X, \mathcal{A})$. Note that Set Splitting may be seen as Positive Not-All-Equal-SAT, i.e. CNF satisfiability with the additional requirements that (1) negations are completely absent ("positive", also sometimes called "monotone" in the literature) and (2) at least one variable is set to false in each clause.

We provide two reductions from Set Splitting to Maximum Common Network Contraction, described in Sects. 3.1 and 3.2. The first one has three interesting features. First, it shows that Maximum Common Network Contraction remains NP-hard when k (the size of the common contraction) is equal to 3. Second, noting that Positive NAE-SAT remains NP-hard with 3 variables per-clauses (Positive-NAE-3-SAT) and at most 4 occurrences of each variable in all clauses [42], it shows that MCNC is NP-hard on phylogenetic networks of bounded degree. Third, when not putting any constraint on the number of occurrences of each element, it allows importing lower bounds for Positive-NAE-3-SAT conditional to ETH (Exponential-Time Hypothesis) [43, Proposition 5.1]. As for the second one, it shows that MCNC remains hard even when one of the networks is a path on 4 nodes, each parent to one leaf. Not only is this simple network a tree, but it also has a constant number of leaves.

For both of the reductions, the intuition is the following. We build a "big network" $\mathcal{N}_1$ and a "small network" $\mathcal{N}_2$ (See Figs. 4 and 5). To get a common contraction of $\mathcal{N}_1$ and $\mathcal{N}_2$ of a given specified size, we will have most of the vertices of $\mathcal{N}_1$ constrained to belonging to specific witness sets corresponding to vertices of $\mathcal{N}_2$. Only some vertices (representing the set X of the Set Splitting instance) will have a choice. As for the connectivity of these witness sets, it helps enforce the "split", i.e. that $S \cap X_1 \neq \emptyset$ and $S \cap X_2 \neq \emptyset$ for all $S \in \mathcal{A}$.

Hardness with common contraction of size $k = 3$

Given an instance $(X, \mathcal{A})$ of Set Splitting, we build two binary phylogenetic networks in the following way (see Fig. 4 for an illustration on an example). First, we introduce two leaves ℓ_S and ℓ'_S for each set $S \in \mathcal{A}$. We also add to this leaf-set two couples of leaves $\{\ell_1, \ell'_1\}$ and $\{\ell_2, \ell'_2\}$, and a single leaf ℓ_r, whose positions in $\mathcal{N}_1, \mathcal{N}_2$ will allow us to enforce specific contractions.

As for the networks, let us start with $\mathcal{N}_1$. Its purpose is to encode the relationships between the elements of X and the sets in $\mathcal{A}$. Its internal node set is composed of one node t_x for each $x \in X$, two nodes u_S and v_S for each $S \in \mathcal{A}$, a root r_1, and other nodes named $r'_1, a_1, a'_1, b_1, b'_1$. We write $U = \{u_S\}_{S \in \mathcal{A}}$ and $V = \{v_S\}_{S \in \mathcal{A}}$. For all $S \in \mathcal{A}$, the parents of ℓ_S and ℓ'_S are u_S and v_S, respectively. ℓ_1 and ℓ'_1 share a parent a'_1, whose own parent is a_1 (see Fig. 4). The same connectivity pattern is set between ℓ_2, ℓ'_2, b'_1 and b_1. The root has a child r'_1 which is the parent of a_1 and b_1, while the other child is the first of a series of intermediary nodes (white circles on the left on Fig. 4), called $W^O[r_1]$, ensuring that $r_1 \rightsquigarrow u_S$, $\forall S \in \mathcal{A}$. Denoting $\mathcal{A}$ as $\{S_1, \ldots, S_m\}$, $W^O[r_1]$ consists of a path $p_1 \to p_2 \to \ldots \to p_{m-1}$ of $m-1$ nodes, such that for $i \in [m-1]$, $p_i \to u_{S_i}$ is an edge. Also add edges $r_1 \to p_1$, and $p_{m-1} \to u_{S_m}$. In addition, we add a leaf ℓ_r, such that p_{m-1} is the parent of ℓ_r. In a similar manner, denoting $X = \{1, \ldots, n\}$, we add a set of nodes $W^O[a_1]$ consisting of a path $q_1 \to \ldots \to q_{n-1}$, such that $q_i \to t_i$ is an edge for $i \in [n-1]$, and add edges $a_1 \to q_1, q_{n-1} \to t_n$. Likewise, we add another set of nodes $W^I[b_1]$ consisting of a path $q'_{n-1} \to q'_{n-2} \to \ldots \to q'_1$ such that $t_i \to q'_i$ is an edge for $i \in [n-1]$. Finally, we add edges $t_n \to q'_{n-1}$ and $q_1 \to b_1$. The purpose of these paths is to implement directed paths with bounded degree nodes. Finally and most importantly, we add an edge from u_S to t_x, and from t_x to v_S, if and only if $x \in S$.

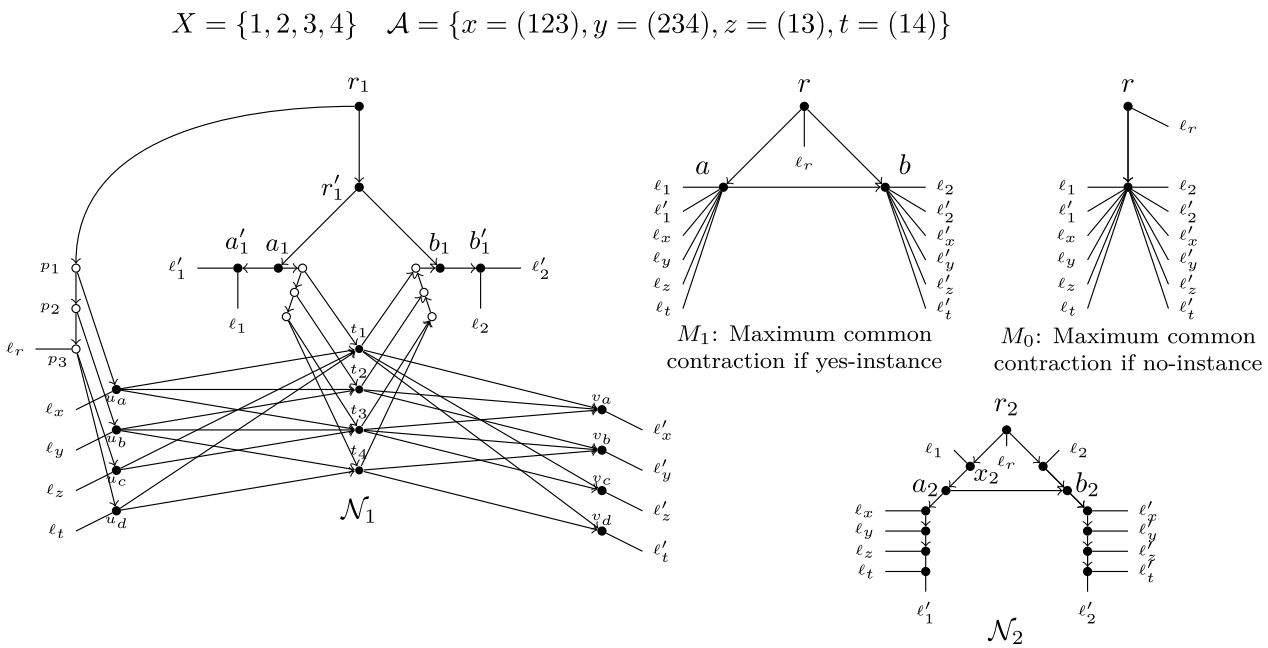

Fig. 4 Illustration of our first reduction from Set Splitting to Maximum Common Network Contraction on an example

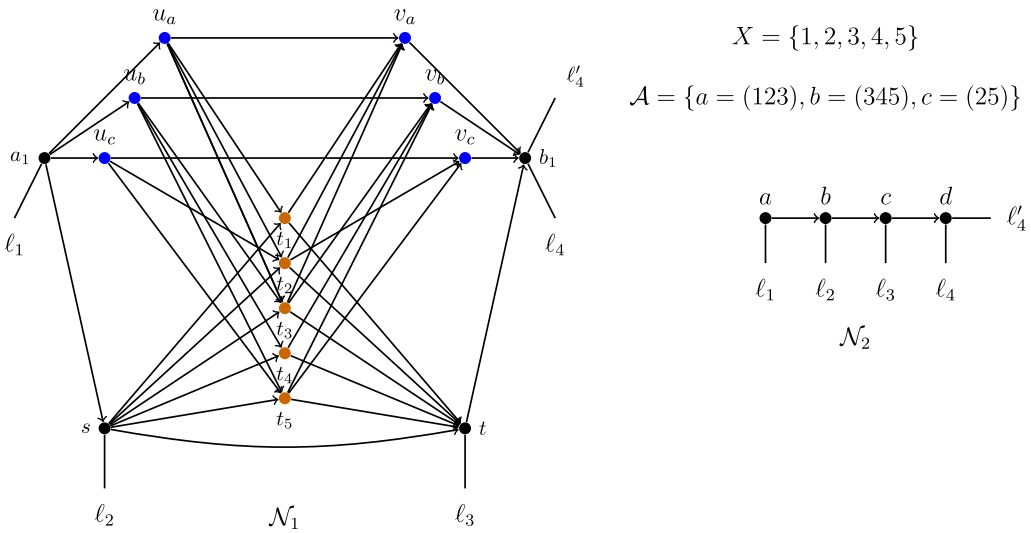

Fig. 5 Illustration of our second reduction, also from Set Splitting. The blue nodes represent the sets of $\mathcal{A}$, and the orange nodes the elements of X. We use this construction in Theorem 11, to prove that even when restricted to networks on 5 leaves, Maximum Common Network Contraction is NP-hard

As for $\mathcal{N}_2$, it is a much simpler network. Its main feature is the presence of a "separation" between $\{\ell_S \mid S \in \mathcal{A}\}$ and $\{\ell'_S \mid S \in \mathcal{A}\}$, emulating the purpose of "splitting" each element of $\mathcal{A}$. Similarly to $\mathcal{N}_1$, it contains nodes r_2 (root) and a_2, b_2. The root r_2 is the parent of ℓ_r, and two non-leaf children, which are themselves the parents of ℓ_1 and a_2, and ℓ_2 and b_2, respectively. a_2 and b_2 are connected by an edge $a_2 \to b_2$. Finally, a_2 b_2 are the parents of, respectively, a path of nodes with children $\{\ell_S \mid S \in \mathcal{A}\} \cup \{\ell_1, \ell'_1\}$ and $\{\ell'_S \mid S \in \mathcal{A}\} \cup \{\ell_2, \ell'_2\}$ (see Figure).

We prove below that $\mathcal{N}_1$ and $\mathcal{N}_2$ have a common contraction of size ≥ 3 (more specifically, the network M_1 depicted on Fig. 4) if and only if $(X, \mathcal{A})$ is splittable.

Theorem 8 *Maximum Common Network Contraction is NP-hard, even when restricted to* $k = 3$

Proof

Necessary contractions in $\mathcal{N}_2$ We start the proof by showing that a common contraction of $\mathcal{N}_1$ and $\mathcal{N}_2$ is at most the network depicted as M_1 on Fig. 4. To that end, in order to keep arguments short, we use $p(\ell)$ to denote the parent of a leaf ℓ. Indeed, note that ℓ_1 and ℓ'_1 share a parent in $\mathcal{N}_1$, whereas this is not the case in $\mathcal{N}_2$. Therefore, to get to any common contraction of $\mathcal{N}_1$ and $\mathcal{N}_2$, one of the paths between $p(\ell_1)$ and $p(\ell'_1)$ in $\mathcal{N}_2$ must be contracted to a single node. There are two such paths: one going through b_2 and r_2, and one going through the common neighbor of a_2 and $p(\ell_1)$, which we call x_2. Both use the path from a_2 to $p(\ell'_1)$, which must be contracted to a point. In addition, we argue that edge $p(\ell_1) \to x_2$ must be contracted. Indeed, if it is not, then the path from x_2 to ℓ_1 going through b_2 an r must be contracted, but this would create a cycle (in the form of a self-loop). The same is true of leaves ℓ_2 and ℓ'_2.

The result of the application of these necessary contractions on $\mathcal{N}_2$ is no other than the network M_1 depicted on Fig. 4. It is composed of three internal nodes r, a, b, three edges $r \to a$, $r \to b$ and $a \to b$, the set of leaves $\{\ell_S \mid S \in \mathcal{A}\} \cup \{\ell_1, \ell_2\}$ (resp. $\{\ell'_S \mid S \in \mathcal{A}\} \cup \{\ell'_1, \ell'_2\}$) are connected to a (resp. b), and leaf ℓ_r connected to r. A common contraction of $\mathcal{N}_1$ and $\mathcal{N}_2$ is therefore either M_1 or a contraction of M_1.

NP-hardness proof We use the fact that M_1 is a contraction of $\mathcal{N}_1$ if and only if there is a M_1-witness-structure in $\mathcal{N}_1$ (Proposition 5). Recall that this is the case if one can partition the internal nodes of $\mathcal{N}_1$ into three sets V_r, V_a, V_b such that: (1) there exists edges going from V_r to V_a, V_r to V_b, and V_a to V_b, (2) all other edges are internal to V_a, V_b, V_r, (3) leaves $\{\ell_S \mid S \in \mathcal{A}\} \cup \{\ell_1, \ell_2\}$ are connected to V_a, leaves $\{\ell'_S \mid S \in \mathcal{A}\} \cup \{\ell'_1, \ell'_2\}$ to V_b and ℓ_r to V_r, and (4) V_r, V_a, V_b each induce weakly-connected graphs in $\mathcal{N}_1$. We now prove that M_1 is a contraction of $\mathcal{N}_1$ if and only if $(X, \mathcal{A})$ is splittable.

$(X, \mathcal{A})$ **splittable** $\Rightarrow$ M_1 **common contraction.** Let us first suppose that $(X, \mathcal{A})$ is splittable, and let X_1, X_2 be a bipartition of X such that $\forall S \in \mathcal{A}$, $S \cap X_1 \neq \emptyset$ and $S \cap X_2 \neq \emptyset$. We first contract $W^O[r_1]$ into r_1, $W^O[a_1]$ into a_1 and $W^I[b_1]$ into b_1. Now, $\forall x \in X$, there is an edge $a_1 \to t_x$ and an edge $t_x \to b_1$, as well as edges $u_S \to t_x$ and $t_x \to v_S$ if and only if $x \in S$. We now contract t_x into a_1 $\forall x \in X_1$, and t_x into b_1 $\forall x \in X_2$. Due to the fact that $\forall S \in \mathcal{A}$, $X_1 \cap S \neq \emptyset$, each u_S is now connected to a_1 (likewise, each v_S is connected to b_1). We contract these edges, as well as $r_1 \to r'_1$, $a'_1 \to a_1$ and $b_1 \to b'_1$, to finish the contractions and obtain M_1.

M_1 **common contraction** $\Rightarrow$ $(X, \mathcal{A})$ **splittable.** Conversely, let us suppose that $\mathcal{N}_1$ is contractible to M_1. There must therefore exist V_a, V_b, V_r verifying properties (1),(2),(3),(4) stated above. To start with, because of the presence of the leaves ℓ_S, we must have $u_S \in V_a$ (likewise, $v_S \in V_b$) $\forall S \in \mathcal{A}$. Also, $a'_1 \in V_a$ and $b'_1 \in V_b$. As V_a is connected, and a_1 is on all paths from a'_1 to any u_S, we also have $a_1 \in V_a$ (and $b_1 \in V_b$). This implies that each t_x must be in V_a or V_b, but not V_r, as all paths from t_x to r_1 (which must be in V_r) are cut by nodes in V_a or V_b. Finally, note that all of $W^O[r_1]$ must be included in V_r, as ℓ_r must have its parent in V_r, and all paths from p_{m-1} (the last node of $W^O[r_1]$ and parent of ℓ_r) to r_1 not using $W^O[r_1]$ are cut by vertices that must be in V_a (namely, $\{u_S\}_{S \in \mathcal{A}}$). The only way to connect p_{m-1} to r_1 is through $W^O[r_1]$ (which is a single path).

Let us now define $X_1 = \{x \in X \mid t_x \in V_a\}$ and $X_2 = \{x \in X \mid t_x \in V_b\}$. For all $S \in \mathcal{A}$, as $u_S \in V_a$, and all paths from u_S to a_1, excluding those that use $W^O[r_1] \cup \{r_1\} \subseteq V_r$, go through some t_x for $x \in S$, we must have some $x \in S \cap X_1$ to ensure the connectivity of V_a. Likewise, since $v_S \in V_b$, there must be some $x \in S \cap X_2$ to ensure the connectivity of V_b. X_1, X_2 is therefore a valid split of $(X, \mathcal{A})$. □

Restricting the instances of Set Splitting to having sets of size 3 and exactly 4 total occurrences of an element x in $\mathcal{A}$ (a variant that it still NP-hard [42]), we get the following.

Theorem 9 *Maximum Common Network Contraction is NP-hard, even with* $k = 3$ *and networks of degree* ≤ 10.

Proof

We use the same reduction as above. Note that in the network $\mathcal{N}_1$ constructed therein, all nodes except $\{t_x\}_{x \in X}$ and $\{u_S, v_S\}_{S \in \mathcal{A}}$ have (in+out) degree at most 4 (achieved by the parent of ℓ_r). Each t_x has degree 2 times its number of occurrences in sets of $\mathcal{A}$, to which we add 2 to account for the edge from $W^O[a_1]$ and to $W^I[b_1]$. With the restriction of Set Splitting under which we work, this yields a degree of 10. As for each u_S, their degree is exactly 5 (1 edge from $W^O[r_1]$, 1 to ℓ_S, and 3 to t_x nodes). Likewise, each v_S has degree 4. From [42], we get the result. □

The complexity remains open for binary networks (i.e. with in+out degree=3 for every node). Finally, we can argue that the number of nodes and edges of $\mathcal{N}_1$ and $\mathcal{N}_2$ produced by our reduction is linear with respect to $|X| + |\mathcal{A}|$. The correspondence with Positive-NAE-SAT allows to import ETH-based lower bounds [43, Proposition 5.1]:

Theorem 10 *Assuming the Exponential Time Hypothesis, Maximum Common Network Contraction cannot be solved in time* $2^{o(|V(\mathcal{N}_1)|+|E(\mathcal{N}_1)|)}$.

Proof

Proposition 5.1 in [43] states that, $\forall k \geq 3$ and assuming ETH, Positive-NAE-k-SAT cannot be solved in time $2^{o(n+m)}$ (they use the term "Monotone" as we use "positive" in this article, i.e. signifying the complete absence of negations).

We recall the equivalence between a Set Splitting instance $(X, \mathcal{A})$ and a Positive-NAE-k-SAT instance: each element of X is a variable and each set in $\mathcal{A}$ is a clause. Therefore $n = |X|$ and $m = |\mathcal{A}|$, and given an instance of Positive-NAE-k-SAT, our reduction constructs a pair of network $\mathcal{N}_1, \mathcal{N}_2$, in which $|U| = |V| = m$ and $|\{t_x\}_x| = n$. To these we need to add $2m + 5$ leaves, $2(n-1)$ nodes in $W^O[a_1]$ and $W^I[b_1]$, and $m - 1$ nodes in $W^O[r_1]$. The remaining nodes constitute a set of constant size. Thus $|V(\mathcal{N}_1)| = O(n + m)$. Likewise, there are $\leq 2km$ edges between U, V and $\{t_x\}_{x\in X}$, $\leq m$ edges between $W^O[r_1]$ and U, and $\leq m$ edges between U, V and the leaves. To finish, there are $\leq n$ edges between $W^O[a_1]$, $W^I[b_1]$ and $\{t_x\}_{x\in X}$, the remaining set of edges being of constant size.

Overall $|V(\mathcal{N}_1)| + |E(\mathcal{N}_1)| = O(n + m)$. Therefore if we had an algorithm solving Maximum Common Network Contraction in time $2^{o(|V(\mathcal{N}_1)|+|E(\mathcal{N}_1)|)}$, it could be used to solve Positive-NAE-k-SAT in time $2^{o(n+m)}$, which is not possible under ETH. □

Hardness with $\mathcal{N}_2$ a path with 5 leaves

We still reduce from Set Splitting, but proceed a bit differently. This time, $\mathcal{N}_2$ is a path on 4 nodes a, b, c, d, to which are respectively connected leaves ℓ_1, ℓ_2, ℓ_3 and $\{\ell_4, \ell'_4\}$ (see Fig. 5 for an illustration). As for $\mathcal{N}_1$, it is composed of 4 internal nodes a_1, s, t, b_1 to which are respectively connected ℓ_1, ℓ_2, ℓ_3 and $\{\ell_4, \ell'_4\}$. As previously, two nodes u_S and v_S are introduced for each $S \in \mathcal{A}$, and one node t_x for each $x \in X$. There are edges from a_1 to s, from a_1 to each u_S, from each u_S to the corresponding v_S, from each v_S to b_1, from s to t, from t to b_1, from s to each t_x, and from each t_x to t. Finally, to encode the instance, there are edges from u_S to t_x and t_x to v_S if and only if $x \in S$. This construction yields the following.

Theorem 11 *Maximum Common Network Contraction is NP-hard, even on networks with 5 leaves, one of them being a tree and* $k = 4$.

Proof

We prove that a Set Splitting instance $(X, \mathcal{A})$ is splittable if and only if, with the construction detailed above and illustrated on Fig. 5, $\mathcal{N}_2$ is a contraction of $\mathcal{N}_1$. This is equivalent to asking whether $(\mathcal{N}_1, \mathcal{N}_2, k)$ is a yes-instance of MCNC with $k = 4$.

$(X, \mathcal{A})$ **splittable** $\Rightarrow$ $\mathcal{N}_2$ **is a contraction of** $\mathcal{N}_1$ Given X_1, X_2 split for $(X, \mathcal{A})$, we describe a sequence of edge contractions on $\mathcal{N}_1$ yielding $\mathcal{N}_2$. We first contract $\{t_x \mid x \in X_1\}$ into s and $\{t_x \mid x \in X_2\}$ into t. These contractions are admissible, as no other directed path from s to t_x other than edge (s, t_x) exist (Proposition 1). Then, since $\forall S \in A$, $S \cap X_1 \neq \emptyset$, there is now $\forall S \in \mathcal{A}$ an edge from u_S to s, which we contract (one can easily check that it is also admissible). Likewise, we contract each v_S into t. The result is $\mathcal{N}_2$.

$\mathcal{N}_2$ **is a contraction of** $\mathcal{N}_1 \Rightarrow (X, \mathcal{A})$ **splittable** By Proposition 5 and Definition 5, $\mathcal{N}_2$ is a contraction of $\mathcal{N}_1$ if and only if there exists an $\mathcal{N}_2$-witness-structure in $\mathcal{N}_1$, i.e. a partition V_a, V_b, V_c, V_d of the internal nodes of $\mathcal{N}_1$, such that: (1) each of these sets is weakly connected, (2) there are edges from V_a to V_b, V_b to V_c and V_c to V_d, but all other edges in $\mathcal{N}_1$ are internal to each set, and (3) the parent of ℓ_1 (resp. $\ell_2, \ell_3, \ell_4, \ell'_4$) in $\mathcal{N}_1$ is in V_a (resp. V_b, V_c, V_d). With such a partition, we must have $a_1 \in V_a, s \in V_b, t \in V_c$ and $b_1 \in V_d$. If any u_S is in V_a, then the undirected distance from V_a to V_d is at most 2 because u_S is at undirected distance 2 from b_1 in $\mathcal{N}_1$ and contractions cannot increase that distance, whereas that distance must be 3. Therefore, all the non-leaf neighbors of a_1 are in another witness set, and $V_a = \{a_1\}$. By the same argument, no v_S can be in V_d and $V_d = \{b_1\}$. From a similar distance argument, if $u_S \in V_c$ then the distance between V_c and V_a is incorrect, and we get $\{u_S\}_{S\in\mathcal{A}} \subseteq V_b$ and $\{v_S\}_{S\in\mathcal{A}} \subseteq V_c$. Each t_x can

only be in V_b or V_c. We define $X_1 = \{x \in X \mid t_x \in V_b\}$ and $X_2 = \{x \in X \mid t_x \in V_c\}$. If for some $S \in \mathcal{A}$, $S \cap X_1 = \emptyset$, then u_S is not connected to the other nodes in V_b. Therefore $\forall S \in \mathcal{A}$, $S \cap X_1 \neq \emptyset$, and likewise for X_2. X_1, X_2 is indeed a split for $(X, \mathcal{A})$. □

On weakly galled-trees, clades, and contraction sequences

Given the hardness of finding maximum contractions for general networks, we searched for classes of networks that could be solved in polynomial time. We found that weakly galled trees, defined below, formed such a class. Our algorithm is slightly involved though, so before diving into it, we explain the main obstacle to overcome when designing algorithms for specific classes. We then describe a set of reduction rules that can be used to simplify instances.

Recall that for $u \in V(\mathcal{N})$, we write $D_{\mathcal{N}}(u)$ for the *clade* of u, i.e., the set of leaves of $\mathcal{N}$ reached by u (we recommend [40] for extensive results on the structural properties of such clusters). In the case of two trees T_1, T_2, the maximum contraction subtree is easy to construct: for each $v \in V(T_1)$ such that $\forall w$ node of T_2, $D_{T_2}(w) \neq D_{T_1}(v)$, we contract the edge between v and its parent in T_1. This algorithm works for three reasons, among others: (1) after contracting an edge from T_1, the result T_1' is still a tree and the algorithm can proceed with the new instance; (2) the set of clades of T_1' is a subset of the clades of T_1; and (3) a clade shared by both trees is always in the maximum contraction subtree. Although these properties may appear trivial, they allow the algorithm to stay applicable after any number of contractions.

This is not as straightforward in the case of networks. For (1), if we start from a galled tree (in which no two cycles contain a common node), a contraction may result in a network that is not a galled tree (see [44] for precise definitions). By applying a contraction, we therefore go outside of the initial network class. As for (2), a contraction may create new clades. Consider $\mathcal{N}_2$ in Fig. 6. There is no $u \in V(\mathcal{N}_2)$ such that $D_{\mathcal{N}_2}(u) = \{d, e, f\}$. However, if we contract one of the edges incoming to the parent of d, resulting in $\mathcal{N}_2'$, we create a node w satisfying $D_{\mathcal{N}_2'}(w) = \{d, e, f\}$. This new clade is actually part of a maximum contraction with $\mathcal{N}_1$. Then for (3), again in Fig. 6, there are $u \in V(\mathcal{N}_1)$ and $v \in V(\mathcal{N}_2)$ satisfying $D_{\mathcal{N}_1}(u) = D_{\mathcal{N}_2}(v) = \{d, e\}$. If we insist on keeping $\{d, e\}$ as a clade, we must contract large portions of the cycles as shown on the top-right, which is suboptimal as shown on the bottom-right.

Weakly galled trees

To address point (1) above, the so-called *weakly galled trees* appear the most appropriate to study, as they form the simplest class that is closed under contractions, to our knowledge. We define this class and extend the notion of clades to it.

A *reticulation cycle* of a network $\mathcal{N}$, or *cycle* for short, is a set of nodes C formed by a pair of directed paths $r \to u_1 \to \ldots \to u_p \to t$ and $r \to v_1 \to \ldots \to v_q \to t$ that only intersect at nodes r and t, and such that t is the only node of C with two in-neighbors from C. We call r the root of C, t its reticulation, and the other nodes its internal nodes. A network is a *weakly galled tree* if no pair of reticulation cycles have an edge in common. In

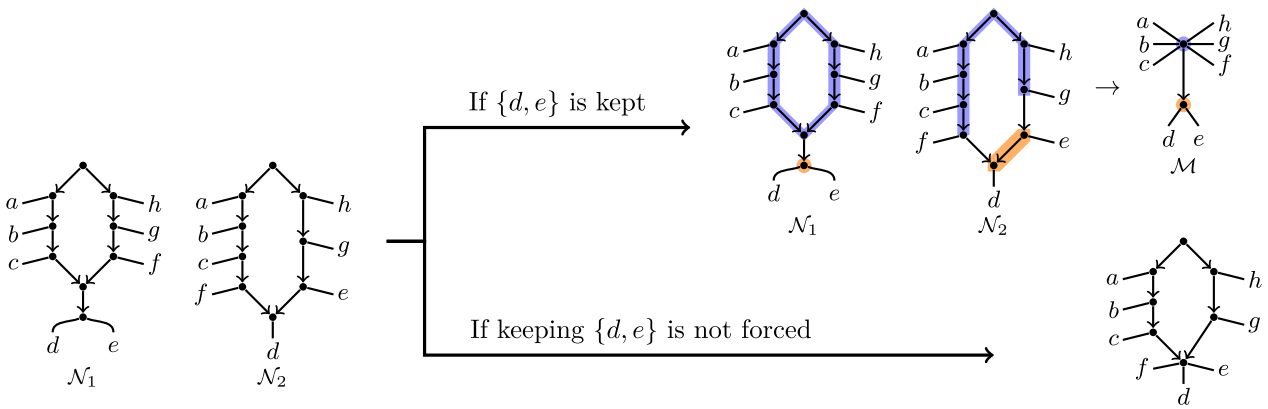

Fig. 6 An example showing that not all common $D(u)$ sets should be preserved. If we enforce keeping the clade $\{d, e\}$ (which is common to both networks), then we get a common contraction with only 2 internal nodes. On the other hand, without enforcing to keep $\{d, e\}$, a larger common contraction is possible. The intuition is that since f is "misplaced", keeping $\{d, e\}$ enforces moving f through the long "upwards" path (depicted in blue)

[44], the following properties were shown on weakly galled trees:

- every reticulation of $\mathcal{N}$ has an in-degree of exactly 2, and is the reticulation of exactly one cycle;
- every cycle C contains exactly one reticulation node of $\mathcal{N}$.

In other words, there is a 1-to-1 correspondence between cycles and reticulations. Note that the internal nodes of such a cycle C have exactly one child in C. We note that a node v could be the root of multiple cycles (which all intersect only at v), and that v could be internal to a cycle, and also be the root of another cycle. The closure property follows from the undirected version of this class, namely cactus graphs, see [45].

Proposition 12 *If $\mathcal{M}$ is a contraction of a weakly galled tree $\mathcal{N}$, then $\mathcal{M}$ is also a weakly galled tree.*

Proof
It is sufficient to argue that the statement holds if $\mathcal{M}$ is obtained from a single contraction on $\mathcal{N}$, as further contractions will inductively preserve membership to the network class. In [45, Observation 5], the authors show that if an undirected graph G is a cactus, then contracting any edge also yields a cactus, where a cactus is an undirected graph in which no two cycles share an edge. In our case, the underlying undirected graph of a weakly galled tree $\mathcal{N}$ is a cactus. After contracting an edge of $\mathcal{N}$, the underlying undirected graph must thus be a cactus. Therefore, no two cycles share an edge in this undirected graph, which also holds after orienting the edges, and thus the resulting network is a weakly galled tree. □

One can verify that a contraction $c(u, v, w)$ cannot increase the number of reticulations. Therefore, by Proposition 12, the number of cycles in a weakly galled tree can only decrease when applying a contraction. In fact, the only way to remove a cycle C is if it has three nodes, that is, a single internal node x that gets contracted with the root or reticulation.

1-clades and 2-clades
To overcome the difficulties regarding clades mentioned above, we next distinguish two types of clades in a weakly galled tree $\mathcal{N}$.

- *(1-clades)* A node $u \in V(\mathcal{N})$ is a *1-clade node* if u is not an internal node of a cycle, in which case $D(u)$ is called a *1-clade*. Note that a 1-clade node u can be the root of a cycle, but only if u is not internal to another cycle.
- *(2-clades)* Let C be a cycle of $\mathcal{N}$ with reticulation t. A pair of distinct nodes u, v of C is called a *2-clade pair* if u and v are on distinct paths that form the cycle, or if one of u, v is the reticulation of C and the other is internal. The set $D(u) \cup D(v)$ is called a 2-clade.

Note that if u does not belong to a cycle, or is a reticulation, or is a root of a cycle while not being internal to another cycle (which may happen in weakly galled trees), then u is a 1-clade node. Also note that multiple nodes may represent the same 1-clade, for instance a reticulation u with a single child v. Examples of 1-clades and 2-clades may be found on Fig. 2: $\{1, 2, 3, 4\}$ is a 1-clade of $\mathcal{N}_1$, whose 1-clade node is the root of the cycle of $\mathcal{N}_1$, and $\{1, 2, 3\}$ is a 2-clade of $\mathcal{N}_2$, with the parents of 1 and 3 as a 2-clade pair. Observe that in Fig. 6, $\{d, e, f\}$ is not a 1-clade of $\mathcal{N}_1$, but is a 2-clade if we take u as the parent of f, and v as the reticulation, thereby addressing the aforementioned problem with standard clades. We let $\Sigma_1(\mathcal{N})$ denote the set of all 1-clades of $\mathcal{N}$, and $\Sigma_2(\mathcal{N})$ denote the set of all 2-clades. These new types of clades are, in some sense, preserved under contractions.

Proposition 13 *(clade-set conservation) If $\mathcal{M}$ is a contraction of a network $\mathcal{N}$, then $\Sigma_1(\mathcal{M}) \subseteq \Sigma_1(\mathcal{N}) \cup \Sigma_2(\mathcal{N})$ and $\Sigma_2(\mathcal{M}) \subseteq \Sigma_2(\mathcal{N})$.*

Proof
It is enough to show that the statement holds if $\mathcal{M}$ is obtained from $\mathcal{N}$ by applying a single contraction $c(u, v, w)$, as further contractions will also produce networks whose clade-set is a subset of $\mathcal{M}$. So suppose that $\mathcal{M}$ is obtained by contracting (u, v) into the new node w. To start with, note that since w inherits the out-neighbors of u and v, we have $D_M(w) = D_{\mathcal{N}}(u) \cup D_{\mathcal{N}}(v) = D_{\mathcal{N}}(u)$ (the latter because u reaches every leaf that v reaches). In the remainder of the proof, we successively consider a 1-clade of $\mathcal{M}$, and to show that it belongs to $\Sigma_1(\mathcal{N}) \cup \Sigma_2(\mathcal{N})$, and a 2-clade of $\mathcal{M}$, to show that it belongs to $\Sigma_2(\mathcal{N})$.

1-clade: Let S be a 1-clade of $\mathcal{M}$. We have $S = D_{\mathcal{M}}(x)$ for some x 1-clade node of $\mathcal{M}$. If $x \neq w$, then x is also a node of $\mathcal{N}$, different from u and v. We argue it is also a 1-clade node in $\mathcal{N}$. Indeed, if it is not, then it is internal to a cycle in $\mathcal{N}$. Since x is not u nor v, no edge incident to x is contracted, and x is still part of a cycle in $\mathcal{M}$ (to see this, note that the only way to remove a cycle of $\mathcal{N}$ is if it has a single internal node which is part of the

contraction). Moreover, it is not hard to see that x is still internal to that cycle after contracting, which is a contradiction. Therefore $D_{\mathcal{N}}(x)$ is a 1-clade of $\mathcal{N}$. This means that in $\mathcal{N}$, x either reaches both u, v, or none of them (if not, x would reach v but not u, in which case v would be a reticulation and x would be internal to the cycle with that reticulation). This implies that $D_{\mathcal{N}}(x) = D_{\mathcal{M}}(x) = S$, as the contraction does not alter the leaves that x reaches.

If $x = w$, then $S = D_{\mathcal{M}}(x) = D_{\mathcal{M}}(w) = D_{\mathcal{N}}(u)$. Either u was a 1-clade node of $\mathcal{N}$, and S is indeed a 1-clade of $\mathcal{N}$ and we are done, or u was internal to a cycle in $\mathcal{N}$. Consider the latter case. If v was the reticulation of the cycle, then u, v was a 2-clade pair of $\mathcal{N}$, with $D_{\mathcal{N}}(u) \cup D_{\mathcal{N}}(v) = S$. In that case, $S \in \Sigma_2(\mathcal{N})$. If v was not the reticulation, then both u, v were internal to a cycle and one can easily see that w is also internal, and thus not a 1-clade. It follows that every 1-clade of M is a 1-clade or 2-clade of $\mathcal{N}$.

2-clade: Let us now look at the other case, where S is a 2-clade of $\mathcal{M}$. Let x, y be the corresponding 2-clade node pair, and C the cycle they stand on. As cycles may only be deleted, and not created, by contractions, C is also a cycle in $\mathcal{N}$, possibly with a difference of one edge (if u, v was an edge in C). Suppose that $w \notin \{x, y\}$, so that x, y are distinct from u, v and thus are also in $\mathcal{N}$. Then x, y also form a 2-clade pair in $\mathcal{N}$. If both x, y either reach both u, v or none of u, v in $\mathcal{N}$, then the contraction does not change the leaves reached by x and y and our result holds. Suppose that, say, x reaches v but not u in $\mathcal{N}$. Then v is the reticulation of C and u is its parent on the other side of x. In that case, $y \notin \{u, v\}$ implies that y is an ancestor of u on C and reaches both u, v. We get that $D_{\mathcal{M}}(x) \cup D_{\mathcal{M}}(y) = (D_{\mathcal{N}}(x) \cup D_{\mathcal{N}}(u)) \cup D_{\mathcal{N}}(y)$ which is equal to $D_{\mathcal{N}}(x) \cup D_{\mathcal{N}}(y)$ since $D_{\mathcal{N}}(u)$ is included in $D_{\mathcal{N}}(y)$. Thus $S \in \Sigma_2(\mathcal{N})$.

If $x = w$ or $y = w$, say $x = w$, then we look at a few cases. If w is internal to C in $\mathcal{M}$, then through case-checking we see that u was also internal in C in $\mathcal{N}$. Therefore u, y is a 2-clade node pair, and $S = D_{\mathcal{M}}(w) \cup D_{\mathcal{M}}(y) = D_{\mathcal{N}}(u) \cup D_{\mathcal{N}}(y)$ is in $\Sigma_2(\mathcal{N})$. If w is the reticulation of C in $\mathcal{M}$, then either v or u were the reticulation of C in $\mathcal{N}$ (to see this, expand w and note that either v has u plus another parent, or u receives the parents of w). If v was the reticulation, then either u was on the same path as y or not. If u was not on the same path as y, then u, y is a 2-clade node pair in $\mathcal{N}$ and we conclude as above. If u was on the same side as y, then $D_{\mathcal{N}}(u) \subseteq D_{\mathcal{N}}(y)$, and v, y is a 2-clade node pair with clade S in $\mathcal{N}$. Last, if u was the reticulation, then u, y is a valid 2-clade pair node in $\mathcal{N}$, and we have $S = D_{\mathcal{N}}(u) \cup D_{\mathcal{N}}(y) \in \Sigma_2(\mathcal{N})$. □

Since new 1-clades or 2-clades cannot be created through contractions, it follows that a 1-clade or a 2-clade that is present in one network $\mathcal{N}_1$, but not in another network $\mathcal{N}_2$, needs to be removed to get a common contraction. In trees, a unique contraction can achieve this, but in networks this is much less obvious. In the case of 2-clades, we must choose between two possible contractions (one for each side of the cycle) that eliminate the 2-clade. Even for 1-clade nodes that are the root of a cycle, we may have to choose between the left or right child to contract. Still, we can argue that if we look at clades close to the root, some contractions can be forced unambiguously in a pre-processing step, which is shown very useful in the next section. Recall that a *reduction rule* alters a given instance if certain conditions are met. We say that a rule is *safe* if, given networks $\mathcal{N}_1, \mathcal{N}_2$, the rule contracts an edge $(u, v) \in E(\mathcal{N}_1) \cup E(\mathcal{N}_2)$ that is contracted in *any* sequence of contractions leading to a maximum common contraction (note that in the following, the roles of $\mathcal{N}_1$ and $\mathcal{N}_2$ can be swapped to reduce $\mathcal{N}_2$ instead).

Rule 1. If the root r_1 of $\mathcal{N}_1$ has a non-reticulation child u such that $D(u)$ is a 1-clade of $\mathcal{N}_1$, but is not a 1-clade nor a 2-clade of $\mathcal{N}_2$, then contract (r_1, u).

Rule 2. If the root r_1 of $\mathcal{N}_1$ has a child u that is an internal node of a cycle, such that every 2-clade containing u is not a 1-clade nor a 2-clade of $\mathcal{N}_2$, then contract (r_1, u).

Lemma 14 *Rule 1 and Rule 2 are safe.*

Proof

Let us argue Rule 1 first. Let $\mathcal{M}$ be any maximum common contraction of $\mathcal{N}_1$ and $\mathcal{N}_2$. Note that since u is not a reticulation, there is only one path from r_1 to u and contracting (r_1, u) is therefore admissible. If (r_1, u) is never contracted to transform $\mathcal{N}_1$ into $\mathcal{M}$, then $D_{\mathcal{N}_1}(u)$ must be a 1-clade of $\mathcal{M}$. Since $\mathcal{M}$ is a contraction of $\mathcal{N}_2$, by Proposition 13, $D_{\mathcal{N}_1}(u)$ is also a 1-clade or a 2-clade of $\mathcal{N}_2$, contradicting the conditions of the rule.

Let us now show that Rule 2 is safe. Let $\mathcal{M}$ be a maximum common contraction of $\mathcal{N}_1$ and $\mathcal{N}_2$. We use the witness structure interpretation of $\mathcal{M}$. That is, let $W_1, W_2, \ldots, W_p$ be the partition of $V(\mathcal{N}_1)$, and $W'_1, \ldots, W'_p$ the partition of $V(\mathcal{N}_2)$ that result in $\mathcal{M}$. Assume without loss of generality that $r_1 \in W_1$. If $u \in W_1$ as well, then $\mathcal{M}$ can be reached after contracting (r_1, u) and the rule is safe. So assume that $r_1 \in W_1$ and $u \in W_2$, again without loss of generality.

Let C be the cycle of $\mathcal{N}_1$ in which u is internal. Let $r_1 = v_1, v_2, \ldots, v_k$ be the path of C that does not contain u, where v_k is the reticulation of C. We claim that $v_k \notin W_1$. Indeed, if $v_k \in W_1$, there are two paths from r_1 to v_k that can be used to connect them in W_1, and the one that uses u cannot be used. If we had $r_1, v_k \in W_1$, after contracting all nodes in the same W_i's, there would be an edge from the node representing W_1 to that representing W_2, because of the edge (r_1, u), and the latter node representing W_2 would then have a path to the node representing W_1, because $v_k \in W_1$. This creates a cycle in $\mathcal{M}$ and thus a contradiction. Hence, there is some $i \in [k-1]$ such that $v_i \in W_1$ but $v_{i+1} \notin W_1$. This implies that the edges (r_1, u) and (v_i, v_{i+1}) of $\mathcal{N}_1$ are not contracted to obtain $\mathcal{M}$, implying in turn that $D_{\mathcal{N}_1}(u) \cup D_{\mathcal{N}_1}(v_{i+1})$ is a 1-clade or a 2-clade of $\mathcal{M}$. By Proposition 13, this is also a 1-clade or 2-clade of $\mathcal{N}_2$, which contradicts the conditions of the rule. □

Note that if, for a 1-clade node u of $\mathcal{N}_1$, the set $D(u)$ is in $\mathcal{N}_2$ as a 2-clade, we may still have to contract the edge between u and its (unique) parent. Hence, an algorithm to find common contractions cannot simply keep the common 1-clades and 2-clades, as is done on trees. An example is given in Fig. 7. A common 2-clade may also need to be removed.

Unicity of clades To finish this section, we prove a useful Proposition about the limited number of ways a set of leaves may be represented as a 1-clade or 2-clade in a weakly galled trees. We show that this number is constant, which will be used in the analysis of our algorithm, in the next section. In its proof, we use the following Proposition:

Proposition 15 *Given a weakly galled tree $\mathcal{N}$ without degree-2 nodes and a set of leaves S, there is:*

1. *At most two 1-clade nodes u such that $D(u) = S$*
2. *At most one 2-clade pair u, v such that $D(u) \cup D(v) = S$*

Proof

The structure of the proof is the following. Given a set of leaves S, we first prove point 1. This is done by considering two hypothetical nodes u, v such that $D(u) = D(v)$. We show that the only case where we can have $u \neq v$ is if one of them is a reticulation, and the other its only child. When we are not in this case, we show that necessarily, $u = v$. To do so, we use a case disjunction based on whether u and v are comparable or not. We then turn to point 2, where we consider 2 hypothetical 2-clade pairs u, v and w, x, and show that if $D(u) \cup D(v) = D(w) \cup D(x)$, then $\{u, v\} = \{w, x\}$.

Point 1 (1-clade case) Assume that there are distinct nodes u, v that are 1-clade nodes such that $D(u) = D(v) = S$. This is possible if v is the unique child of u, so assume this is not the case. Suppose that u, v are incomparable, that is, neither is a descendant of the other. Let P be any path from u to a leaf ℓ. Because $D(v) = D(u)$, v also reaches ℓ, and any path P' from v

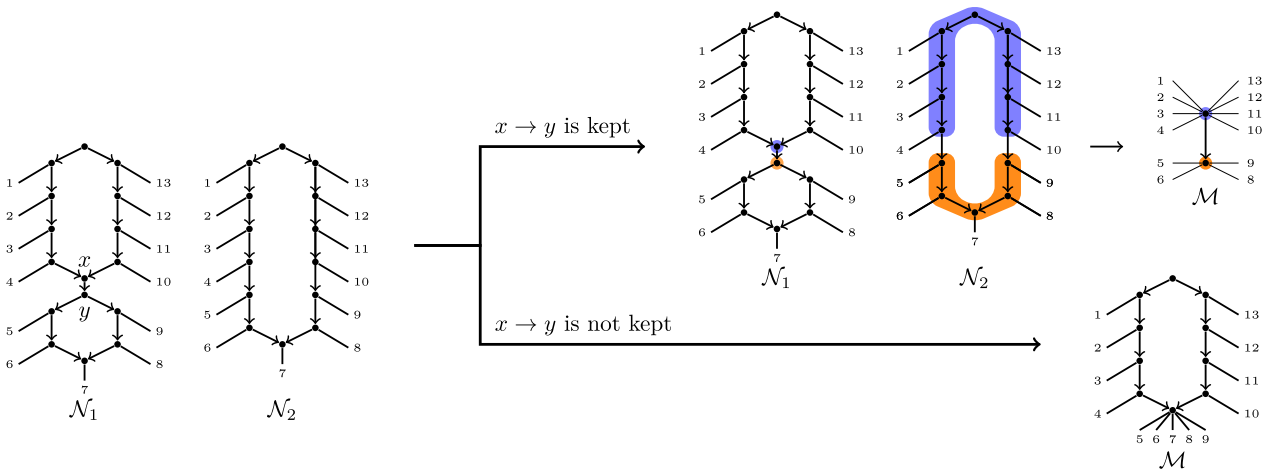

Fig. 7 Example of an instance where keeping a common 1-clade is not optimal. Indeed, if the edge incoming to the root of the second cycle in $\mathcal{N}_1$ is kept (corresponding to the 1-clade $\{5, 6, 7, 8, 9\}$), then its two ends belong to two different sets W_u and W_v of an $\mathcal{M}$-witness structure, with $\mathcal{M}$ common contraction of $\mathcal{N}_1$ and $\mathcal{N}_2$. One of them, say u, must reach exactly $\{5, 6, 7, 8\}$ in $\mathcal{M}$, and W_u must be weakly connected. The only possibility to achieve this in $\mathcal{N}_2$ is to contract the cycle into a single edge, as depicted by the orange and blue shaded areas on the top path. Without forcing the preservation of $\{5, 6, 7, 8\}$ (bottom path), we get a larger common contraction. This example highlights the difference between computing a maximum common contraction of two trees (where one can simply keep the clades in common) and two networks

to ℓ must intersect P because, in particular, both P and P' must contain the parent $p(\ell)$ of ℓ. The first node t at which P and P' intersect is a reticulation since it receives a distinct in-neighbor from both paths. Note that $t \notin \{u, v\}$ since u, v are incomparable and they both reach t. Moreover, u and v must have a common ancestor w (in particular, the root is such an ancestor). This means that u and v are part of some cycle whose reticulation is t. By incomparability, u and v are internal to that cycle, and thus cannot be 1-clade nodes, a contradiction.

Next suppose that u and v are comparable, and without loss of generality that u reaches v. If u has only one child w, then u is a reticulation or the root and under our current assumption (that v is not the only child of u) we have that $w \neq v$. Also, $D(u) = D(w)$ and w cannot be a reticulation (if it was, its in-neighbors would be on the cycle of w, including u — as the latter is a 1-clade, u would root that cycle and have two children). In that case, redefine u to be w (v still descends from u). In that manner, we may assume that u has at least two children p and q. If u is not the root of any cycle, then $D(p)$ and $D(q)$ are disjoint, and, as v descends from u, $D(v)$ cannot contain both $D(p)$ and $D(q)$. So assume that u is the root of some cycle C that contains p and q and a reticulation t. Note that v is not internal on C as it is 1-clade. If $v \neq t$ or if v does not descend from t, then t must reach leaves that v does not. So we may assume that v is t or descends from it. In that case, assume without loss of generality that q is internal to C. Then q has some child q' not on C. Moreover, q' reaches some leaf that t, and thus v, does not reach. Thus $D(u) \neq D(v)$, and therefore only two nodes can have the same 1-clade.

Point 2 (2-clade case) As for Point 2, let u, v be a 2-clade pair such that $D(u) \cup D(v) = S$. Let us call C the cycle they belong to, r the root of C and t its reticulation. Consider w, x another 2-clade pair such that $D(w) \cup D(x) = S$, belonging to cycle C' with root r' and reticulation t'. We will show that w, x can only be equal to u, v. The discussion below first shows that we must have $t = t'$ and $C = C'$, and then concludes. We rule out $t \neq t'$ by contradiction, and distinguishing cases based on whether t, t' are comparable or not. To start with, recall that in a 2-clade pair, at least one vertex must be internal to its cycle. Without loss of generality, say that u and w are internal, and thus have a child outside of their respective cycles.

Case 1: $C \neq C'$. Suppose that $C \neq C'$, and thus $t \neq t'$. If t is an ancestor of t', the child of u outside of C must reach a leaf that w and x cannot reach (otherwise, there would be a cycle with two reticulations from u going through t then w or x, then to the intersection on the paths leading to that leaf). This is in contradiction with $D(u) \cup D(v) = D(w) \cup D(x)$. A similar argument applies if t' is an ancestor of t.

If t, t' are incomparable, we first show that $D(t) \cap D(t') = \emptyset$. Indeed, suppose $D(t) \cap D(t') \neq \emptyset$ and let ℓ be a leaf in that intersection. As in the proof of Point 1 above, let P be any path from t to ℓ, and P' any path from t' to ℓ. These two paths intersect at least at the parent of ℓ. The first node s at which they intersect must be a reticulation (as if it had only one in-neighbor, this node would be in both P and P', and s would not be the first intersection). In addition, t and t' also have a common ancestor (the root of the network for instance). The lowest common ancestor of t and t' is therefore the root of a cycle whose reticulation is S. This is not possible in a weakly galled network, in which reticulations (such as t and t') belong to exactly one cycle. Therefore if t and t' are incomparable, $D(t) \cap D(t') = \emptyset$. We show that this implies that the roots r and r' of C and C' must also be incomparable. Indeed, if r can reach r' while t and t' are incomparable, then either r' is on the cycle C, in which case C, C' intersect only at r' (here r' could be equal to r or internal to C), or r' is reachable from r by going to a node of C (possibly r) then following a path of edges not on C. In any case, there is a leaf in $D(t)$ (and therefore in $D(u) \cup D(v)$) unreachable by both the internal vertices or the reticulation of C', and in particular w and x. This is a contradiction. A similar argument applies if r is reachable from r'. However, if r and r' are incomparable, then by the same argument applied to t and t' (taking a leaf reachable by both, and a common ancestor), we get that r and r' are part of the same cycle, which is a contradiction. Therefore, if $C \neq C'$, in all cases, $D(w) \cup D(x)$ cannot be equal to $D(u) \cup D(v)$. So we may assume that $t = t'$ and $C = C'$.

Case 2: $C = C'$. Suppose, without loss of generality, that among u, v, w, x, the node u is at minimum distance from r on C. If $u \notin \{w, x\}$, then the child of u outside of C reaches a leaf that neither w nor x can reach, because w, x are either below u or on the other side of C. So assume $u = w$. Then consider the node among v, x that is the closest to r, say v. If we assume $v \neq x$, then v must be internal in C, and the child of v outside of C must reach a leaf that x cannot reach (because x descends from v on C) and that w cannot reach (because it is on the other side of the cycle). Therefore, $\{u, v\} = \{w, x\}$. □

A dynamic programming algorithm for weakly galled trees

We now show that computing the minimum number of contractions needed to achieve a common contraction for a pair of weakly galled trees $\mathcal{N}_1, \mathcal{N}_2$ is feasible in polynomial time. Let C be a cycle of $\mathcal{N} \in \{\mathcal{N}_1, \mathcal{N}_2\}$ with root r. It will be convenient to use a cyclic ordering of its nodes. Let $r, v_1, v_2, \ldots, v_l, r$ be the sequence of nodes obtained by traversing the cycle C when seen as an undirected graph, starting at r and choosing one of its neighbors v_1 arbitrarily. We view this as traversing the cycle counter-clockwise. For a node u of C, we write $u+1$ for the node that succeeds u in this sequence, and $u-1$ for the node that precedes u (note that this is also well-defined for r, as it has only one predecessor, which is v_ℓ, and one successor, which is v_1). For u, w in C, we write $u \preceq_C w$ if $u = w$ or u precedes w in the sequence, and we drop the C subscript if clear. As a special case, for every u of C both $u \preceq_C r$ and $r \preceq_C u$. For $u \preceq v$, we denote by $|v-u|$ the number of edges on the path from u to v, following the cyclic ordering. If $v \prec u$, then $|v-u| = 0$.

Some useful subnetworks

Let $\mathcal{N}$ be a weakly galled tree. In our recurrences, we will make use of the following networks that can be obtained from $\mathcal{N}$. They are all displayed in Fig. 8.

- $\mathcal{N}\langle u\rangle$ and $\mathcal{N}\langle \overset{\bullet}{u}\rangle$ (Fig. 8A): for u a 1-clade node, $\mathcal{N}\langle u\rangle$ is the network induced by u and all the nodes that u reaches. $\mathcal{N}\langle \overset{\bullet}{u}\rangle$ is obtained from $\mathcal{N}\langle u\rangle$ by adding a new node r and the edge $r \to u$. The circle above u represents a "dangling" root with a single child u.
- $\mathcal{N}\langle C, u, v\rangle$ (Fig. 8B): for C a cycle of $\mathcal{N}$ with reticulation t, and $u \prec t \prec v$, $\mathcal{N}\langle C, u, v\rangle$ is obtained from the network induced by any node reached by $u+1, v-1$ (inclusively), then adding a new node r_{uv} with children $u+1, v-1$.
  Such a sub-network can be seen as the result of contracting the upper part of the cycle between u and v into its root.
  It is possible that $u+1 = v-1$, in which case they are the reticulation t of the cycle. In this case, notice that $\mathcal{N}\langle C, u, v\rangle$ is equal to $\mathcal{N}\langle \overset{\bullet}{t}\rangle$. It is also possible that $u = v$ is the root of C (which is why we need to specify C in the notation, since u can root multiple cycles). In this case, the cycle C is kept as is, but if u has children outside of C, they are removed.
- $\mathcal{N}^B\langle u, v\rangle$ (Fig. 8C): for u, v a 2-clade node pair, or $u = v$ a reticulation, $\mathcal{N}^B\langle u, v\rangle$ is the subnetwork obtained by (1) removing any node not descending from u or v and (2) contracting the path from u to v going through the reticulation of the cycle. Note that u or v could be the reticulation. The B stands for *bottom*.
- $\mathcal{N}^L\langle u, v\rangle$ (Fig. 8D): for u, v a pair of internal nodes belonging to the same side of a cycle, $\mathcal{N}^L\langle u, v\rangle$ is the network induced by all the nodes reached by u, but that are not reached by $v+1$. For technical reasons, an extra leaf ℓ_T is added, with v as its parent (which simulates that there existed a lower portion of the cycle). Note that this network includes u and

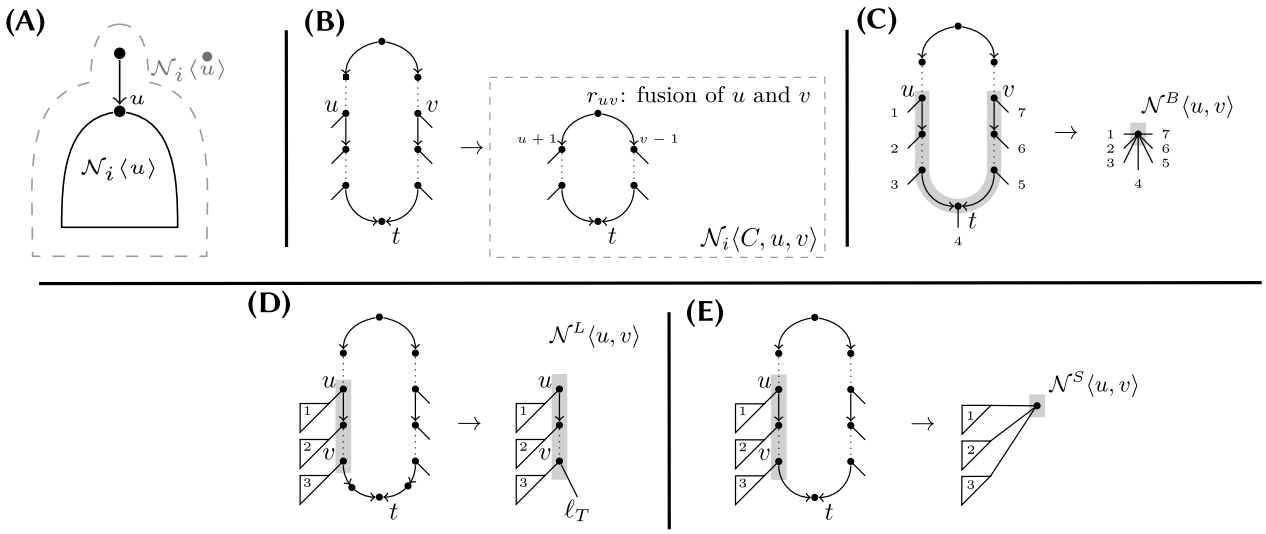

Fig. 8 Sub-networks used in our polynomial algorithm for weakly galled trees. The dynamic programming is primarily based on $\mathcal{N}_i\langle \overset{\bullet}{u}\rangle$ and $\mathcal{N}_i\langle C, u, v\rangle$ ((A) and (B)) on the figure. The others are intermediate networks on which reduction rules are applied to fall back to $\mathcal{N}_i\langle \overset{\bullet}{u}\rangle$ and $\mathcal{N}_i\langle C, u, v\rangle$

excludes $v+1$. Moreover, if v is not u or one if its descendants, we assume that $\mathcal{N}^L\langle u,v\rangle$ is a network with no node. The L stands for *lateral*.

- $\mathcal{N}^S\langle u,v\rangle$ (Fig. 8E): for u, v pair of nodes belonging to the same side of a cycle, $\mathcal{N}^S\langle u,v\rangle$ is obtained from $\mathcal{N}^L\langle u,v\rangle$ by removing ℓ_T and contracting the path from u to v. If v does not descend from u, this is the empty network. The S stands for *side*.

Join subnetworks Let $\mathcal{N}_1, \mathcal{N}_2$ be two networks with distinct leaf sets. We write $\mathcal{N}_1 * \mathcal{N}_2$ for the network obtained by identifying the roots of $\mathcal{N}_1$ and $\mathcal{N}_2$ (or, to explain this differently, we create a new root that inherits all the children of the roots of $\mathcal{N}_1$ and $\mathcal{N}_2$, and delete the two previous roots). If $\mathcal{N}$ is a weakly galled tree, we say that a network $\mathcal{N}'$ is a *join subnetwork of* $\mathcal{N}$ if $\mathcal{N}' = \mathcal{N}^1 * \ldots * \mathcal{N}^p$, with each $\mathcal{N}^i$ either of the form $\mathcal{N}\langle \overset{\bullet}{u}\rangle$ for some $u \in V(\mathcal{N})$, or $\mathcal{N}\langle C,u,v\rangle$ for some cycle C of $\mathcal{N}$ and u, v on C. If $p=1$, then $\mathcal{N}'$ is called a *prime join subnetwork* of $\mathcal{N}$, and otherwise it is a *composite join subnetwork* of $\mathcal{N}$. Note that by the definition of $*$, no two $\mathcal{N}^i$ subnetworks may have a leaf in common. Two join subnetworks $\mathcal{N}_1 = \mathcal{N}_1^1 * \cdots * \mathcal{N}_1^p$ and $\mathcal{N}_2 = \mathcal{N}_2^1 * \cdots * \mathcal{N}_2^q$ are said to be *matching* if $p=q$ and $\forall i \in [p], L(\mathcal{N}_1^i) = L(\mathcal{N}_2^i)$.

Our dynamic programming tables will contain join subnetworks as entries. The following will be useful to argue that applying contractions keeps us in the realm of join subnetworks.

Lemma 16 *Let $\mathcal{N}$ be a weakly galled tree and let $\mathcal{N}'$ be a join subnetwork of $\mathcal{N}$. Then applying to $\mathcal{N}'$ any sequence of contractions of edges outgoing from its root yields a join subnetwork of $\mathcal{N}$.*

Proof

It suffices to argue that a single contraction preserves the desired form, as in turn, applying any number of them will do so. Let r be the root of $\mathcal{N}'$ and suppose that edge $r \to w$ is contracted. Denote $\mathcal{N}' = \mathcal{N}^1 * \ldots * \mathcal{N}^p$. Assume without loss of generality that w belongs to $\mathcal{N}^1$. Suppose first that $\mathcal{N}^1 = \mathcal{N}\langle \overset{\bullet}{w}\rangle$ for some $w \in V(\mathcal{N})$ (see Fig. 8 for the illustrations of the different notions of subnetworks). Notice that because $\mathcal{N}$ is a weakly galled tree, $\mathcal{N}\langle w\rangle$ is itself a composite join subnetwork of $\mathcal{N}$, as it can be obtained by joining its 1-clade children under a common root, and joining the cycles that w is a root of. In other words, we may write $\mathcal{N}\langle w\rangle = \mathcal{N}_w^1 * \ldots * \mathcal{N}_w^q$. Then, after contracting $r \to w$, the network $\mathcal{N}'$ becomes $\mathcal{N}_w^1 * \ldots * \mathcal{N}_w^q * \mathcal{N}^2 * \ldots * \mathcal{N}^p$, which is a join subnetwork of $\mathcal{N}$, as desired. This takes care of the case where $\mathcal{N}^1$, the first network joined to obtain $\mathcal{N}'$, has the form $\mathcal{N}\langle w\rangle$.

So suppose that $\mathcal{N}^1 = \mathcal{N}\langle C,u,v\rangle$ for some cycle C and nodes u, v. Then $w = u+1$ or $w = v-1$. Either way, if the cycle is still present after contracting $r \to u$, $\mathcal{N}^1$ can be replaced with $\mathcal{N}\langle u+1, v\rangle$ or $\mathcal{N}\langle u, v-1\rangle$, in which case the contraction yields another join subnetwork (as the other subnetworks $\mathcal{N}^2, \ldots, \mathcal{N}^p$ are unaltered). If the cycle is contracted to an edge, then $\mathcal{N}^1$ becomes $\mathcal{N}\langle \overset{\bullet}{t}\rangle$, with t the reticulation of the cycle. Again, this preserves the join subnetwork form. □

The last but not least ingredient before we can describe our procedure is to use Rules 1-2 from the previous section in order to reach join subnetworks with corresponding leaf sets.

Lemma 17 *Suppose that Rules 1-2 are not applicable to $\mathcal{N}_1$ or $\mathcal{N}_2$. Then the networks can be written as matching join subnetworks $\mathcal{N}_1 = \mathcal{N}_1^1 * \ldots * \mathcal{N}_1^p$ and $\mathcal{N}_2 = \mathcal{N}_2^1 * \ldots * \mathcal{N}_2^p$ such that $L(\mathcal{N}_1^i) = L(\mathcal{N}_2^i)$ for every $i \in [p]$.*

Proof

Let r_1 and r_2 be the roots of $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively. Notice that a network is a join subnetwork of itself, so we may write $\mathcal{N}_1 = \mathcal{N}_1^1 * \ldots * \mathcal{N}_1^p$ and $\mathcal{N}_2 = \mathcal{N}_2^1 * \ldots * \mathcal{N}_2^q$. If the superscripts can be arranged so that leaf sets are equal as desired in the statement, we are done, so assume this is not the case. Let us start with an observation:

Claim: If two leaves x and y from a network $\mathcal{N}$ belong to a 1-clade or 2-clade other than $L(\mathcal{N})$, then there is an (undirected) path between them that does not use the root.

Proof of Claim: In case of a 1-clade $S = D(u) \neq L(\mathcal{N})$, simply concatenate a path from x to u (which must exist as u reaches x) and a path from u to y. Likewise, in the case of a 2-clade $S = D(u) \cup D(v)$, use u and v, then the reticulation of the cycle that u and v belong to, as intermediate points to get a path from x to y not using the root. □

The matching condition holds if $\forall i \in [p]$, $\exists j$ such that $L(\mathcal{N}_1^i) = L(\mathcal{N}_2^j)$. Therefore, if it is not verified, there exists $i \in [p]$ such that $\forall j \in [q]$, $L(\mathcal{N}_1^i) \neq L(\mathcal{N}_2^j)$. Since $\{L(\mathcal{N}_2^j)\}_{j\in[p]}$ forms a partition of the common leafset

of $\mathcal{N}_1$ and $\mathcal{N}_2$, there must exist j such that, in addition $L(N_1^i) \cap L(\mathcal{N}_2^j) \neq \emptyset$. Let us pick such a pair i, j and denote $A = L(\mathcal{N}_1^i)$ and $B = L(\mathcal{N}_2^j)$. We have $A \neq B$ and $A \cap B \neq \emptyset$, and therefore $B \backslash A \neq \emptyset$ or $A \backslash B \neq \emptyset$. We analyze the case $B \setminus A \neq \emptyset$, but the arguments apply symmetrically to the other. Note that $\mathcal{N}_2^j$ is either rooted at a node with a single child ($\mathcal{N}_2^j = \mathcal{N}_2\langle\overset{\bullet}{v}\rangle$ for some v), or rooted at a cycle ($\mathcal{N}_2^j = \mathcal{N}_2\langle C, u, v\rangle$ for some u, v).

Let us first look at $\mathcal{N}_2^j = \mathcal{N}_2\langle\overset{\bullet}{v}\rangle$, i.e. the case where the root r_2 of $\mathcal{N}_2^j$ has a single child v. Let $x \in A \cap B$ and $y \in B \backslash A$. We have that v is a child of r_2 in $\mathcal{N}_2$ and, moreover, $D_{\mathcal{N}_2}(v) = B$ is a 1-clade of $\mathcal{N}_2$ containing both x an y. However, in $\mathcal{N}_1$, all paths between x and y go through r_1, and by contraposing the Claim above, x and y cannot share a clade, implying that B cannot be a 1-clade nor a 2-clade of $\mathcal{N}_1$. Therefore, Rule 1 is applicable to $r_2 \to v$, a contradiction.

So consider the case where $\mathcal{N}_2^j = \mathcal{N}_2\langle C, u, v\rangle$, i.e. $\mathcal{N}_2^j$ rooted at a cycle C. Let t be the reticulation of C and let x be a leaf that t reaches (in $\mathcal{N}_2$ and $\mathcal{N}_2^j$). Suppose for now that $x \in A$. Then let $y \in B \setminus A$, which again exists. Let v be a non-reticulation child of r_2 in $\mathcal{N}_2$ that reaches y (which exists), and note that v also reaches x. Therefore, any 2-clade involving v contains x and y. Again because all paths between x and y in $\mathcal{N}_1$ go through r_1, such a 2-clade cannot be in $\mathcal{N}_1$, in which case Rule 2 can be applied to $r_2 \to v$, a contradiction. So suppose that $x \notin A$. Then let $y \in A \cap B$. As before, we let v be a non-reticulation child of r_2 that reaches y. This v also reaches x, and we get the same contradiction. □

A dynamic programming algorithm

We next describe our dynamic programming algorithm. Its overall workflow is sketched in Fig. 9, relying on the concept of join subnetworks. The present section details the "decomposition" and "recursive equations" displayed in Fig. 9.

We define two functions f_P and f_C, linked together by recursive equations. They both compute, given two join subnetworks $\mathcal{N}_1', \mathcal{N}_2'$ of $\mathcal{N}_1$ and $\mathcal{N}_2$, the minimum number of contractions needed on both networks to achieve a common contraction. The difference is that $f_P(\mathcal{N}_1', \mathcal{N}_2')$ assumes that the given pair are *prime* join subnetworks (i.e. for $i = 1, 2$, $\mathcal{N}_i' = \mathcal{N}_i\langle\overset{\bullet}{v}\rangle$ for some v, or $\mathcal{N}_i' = \mathcal{N}_i\langle C, u, v\rangle$ for some cycle C and nodes u, v), whereas $f_C(\mathcal{N}_1', \mathcal{N}_2')$ is given an arbitrary pair of join subnetworks (which may be composite or prime). The role of f_C is to apply rules 1 and 2, in order to get a matching pair of join-subnetworks, and then call f_P on the resulting pairs of prime subnetworks. Our value of interest is $f_C(\mathcal{N}_1, \mathcal{N}_2)$.

For join subnetworks $\mathcal{N}_1', \mathcal{N}_2'$, we start with the simple base cases.

Base cases. If $L(\mathcal{N}_1') \neq L(\mathcal{N}_2')$ or if one of $\mathcal{N}_1', \mathcal{N}_2'$ is not acyclic, then $f_C(\mathcal{N}_1', \mathcal{N}_2') = f_P(\mathcal{N}_1', \mathcal{N}_2') = \infty$ as no common contraction is possible. If none of the above holds and if $\mathcal{N}_1'$ and $\mathcal{N}_2'$ each have no node, or each has a single node, put $f_C(\mathcal{N}_1', \mathcal{N}_2') = f_P(\mathcal{N}_1', \mathcal{N}_2') = 0$.

Main entries f_C. If $\mathcal{N}_1'$ and $\mathcal{N}_2'$ are both prime join subnetworks, then $f_C(\mathcal{N}_1', \mathcal{N}_2') = f_P(\mathcal{N}_1', \mathcal{N}_2')$ (defined later

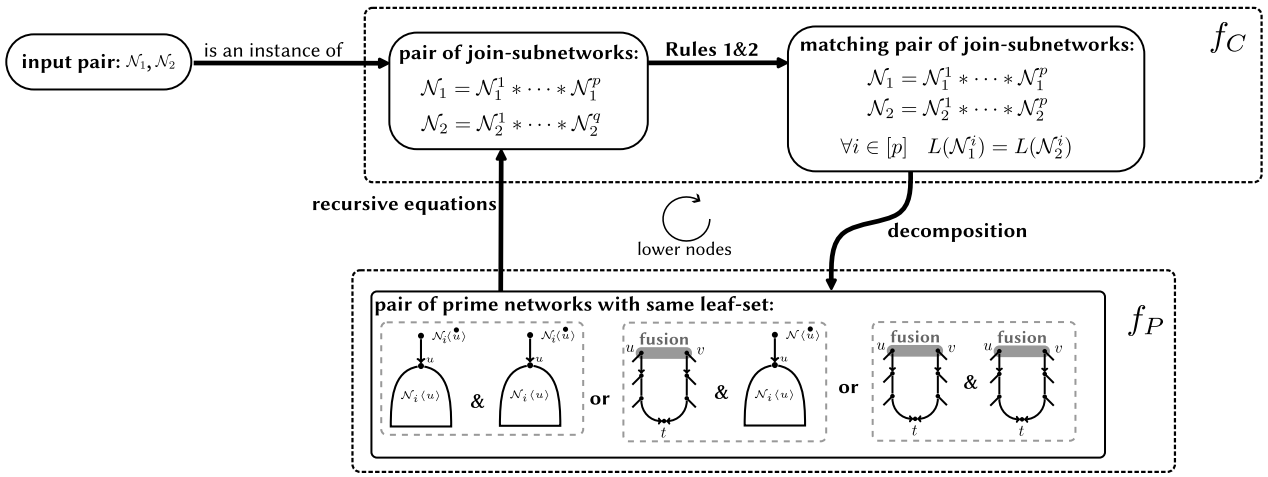

Fig. 9 Overview of our algorithm for weakly galled trees. The final answer is $f_C(\mathcal{N}_1, \mathcal{N}_2)$. To compute it, f_C applies rules 1 and 2, which detect necessary edge contractions at the roots of the networks. When these rules cannot be applied anymore, one can decompose the networks into matching pairs of prime subnetworks, which are given to f_P. Values of f_P are in turn computed with recursive equations, requiring values of f_C, on subnetworks rooted at lower nodes

below). Otherwise, suppose that at least one of $\mathcal{N}_1'$ or $\mathcal{N}_2'$ is a composite join subnetwork of $\mathcal{N}_1$ or $\mathcal{N}_2$. Then proceed as follows:

- As long as one of Rule 1 or Rule 2 is applicable to $\mathcal{N}_1'$ or $\mathcal{N}_2'$, we apply it. Let $\mathcal{N}_1'', \mathcal{N}_2''$ be the resulting pair of networks, which are join subnetworks by Lemma 16, and suppose that Rule 1-2 enforced k contractions. Then we have $f_C(\mathcal{N}_1', \mathcal{N}_2') = k + f_C(\mathcal{N}_1'', \mathcal{N}_2'')$
- If no rule is applicable to either network, by Lemma 17, we have matching join subnetworks $\mathcal{N}_1' = \mathcal{N}_1^1 * \ldots * \mathcal{N}_1^p$ and $\mathcal{N}_2' = \mathcal{N}_2^1 * \ldots * \mathcal{N}_2^p$, with $L(\mathcal{N}_1^i) = L(\mathcal{N}_2^i)$ for each $i \in [p]$. We can solve each pair of prime subnetworks independently with $f_C(\mathcal{N}_1', \mathcal{N}_2') = \sum_{i=1}^{p} f_P(\mathcal{N}_1^i, \mathcal{N}_2^i)$.

Prime entries f_P. We now assume that both $\mathcal{N}_1'$ and $\mathcal{N}_2'$ are prime join subnetworks of $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively. By definition, $\mathcal{N}_1'$ must have the form $\mathcal{N}_1' = \mathcal{N}\langle \overset{\bullet}{u} \rangle$ or $\mathcal{N}_1' = \mathcal{N}\langle C, u, v \rangle$. The same holds for $\mathcal{N}_2'$. There are thus four combinations to check.

Case 1: $\mathcal{N}_1' = \mathcal{N}_1\langle \overset{\bullet}{u} \rangle$ and $\mathcal{N}_2' = \mathcal{N}_2\langle \overset{\bullet}{v} \rangle$ for $u \in V(\mathcal{N}_1)$, $v \in V(\mathcal{N}_2)$. We argue that in this case, $f_P(\mathcal{N}_1\langle \overset{\bullet}{u} \rangle, \mathcal{N}_2\langle \overset{\bullet}{v} \rangle) = f_C(\mathcal{N}_1\langle u \rangle, \mathcal{N}_2\langle v \rangle)$. This is because both networks start with a single edge $r_u \to u$ and $r_v \to v$ (respectively). Therefore, we argue that an optimal common contraction simply keeps both of these edges and defers the computation to $\mathcal{N}_1\langle u \rangle$ versus $\mathcal{N}_2\langle v \rangle$, which are join subnetworks (possibly prime) and can use other f_C entries.

Case 2: $\mathcal{N}_1' = \mathcal{N}_1\langle \overset{\bullet}{u} \rangle$ and $\mathcal{N}_2' = \mathcal{N}_2\langle C, v, w \rangle$ for $u \in V(\mathcal{N}_1)$, and cycle C of $\mathcal{N}_2$ and $v, w \in V(\mathcal{N}_2)$ belonging to C. In this case, we argue that:

$$f_P(\mathcal{N}_1\langle \overset{\bullet}{u} \rangle, \mathcal{N}_2\langle C, v, w \rangle) = \min \begin{cases} f_C(\mathcal{N}_1\langle u \rangle, \mathcal{N}_2^B\langle v+1, w-1 \rangle) + |w - v| - 2, \\ f_C(\mathcal{N}_1\langle u \rangle, \mathcal{N}_2\langle C, v, w \rangle) + 1 \end{cases}$$

Here, $\mathcal{N}_1\langle \overset{\bullet}{u} \rangle$ starts with a single edge $r_u \to u$, while $\mathcal{N}_2\langle C, u, v \rangle$ starts with a root of a cycle. The minimization distinguishes two cases: either (a) edge $r_u \to u$ is conserved, in which case the cycle must be contracted to a single edge, with the root having one child, or (b) the edge is contracted.

Case 3: $\mathcal{N}_1' = \mathcal{N}_1\langle C, u, v \rangle$ and $\mathcal{N}_2' = \mathcal{N}_2\langle \overset{\bullet}{w} \rangle$ for a cycle C of $\mathcal{N}_1$, $u, v \in V(\mathcal{N}_1)$ and $w \in V(\mathcal{N}_2)$. This case is symmetric to the previous case, and is handled the same way.

Case 4: $\mathcal{N}_1' = \mathcal{N}_1\langle C_1, u, v \rangle$ and $\mathcal{N}_2' = \mathcal{N}_2\langle C_2, w, x \rangle$, for a cycle C_1 of $\mathcal{N}_1$ and nodes $u, v \in C_1$, and cycle C_2 of $\mathcal{N}_2$ and $w, x \in C_2$. This is the most complicated case, which is illustrated in Fig. 10. We denote the roots of C_1 and C_2 as r_{uv}, r_{wx} and their reticulations by t_1, t_2. Let us consider the following possibilities for $f_P(\mathcal{N}_1\langle C_1, u, v \rangle, \mathcal{N}_2\langle C_2, w, x \rangle)$: either we contract an edge incident to u, v, w, or x on the cycles, or not. In the latter case, we treat these incident edges as "kept", and we must handle the bottom of the cycles. For clarity, this value is described in a separate and temporary entry $f_B(a, b, c, d)$, which represents the best scenario if we contract the subpath of $\mathcal{N}_1\langle C_1, u, v \rangle$ from a to b into t_1, and the subpath of $\mathcal{N}_2\langle C_2, w, x \rangle$ from c to d into t_2. We get the equation below. We recall first that given a network N and an edge (x_1, x_2) of N, $N/(x_1, x_2)$ denotes the network obtained from N by contracting (x_1, x_2).

$$f_P(\mathcal{N}_1\langle C_1, u, v \rangle, \mathcal{N}_2\langle C_2, w, x \rangle) = \min \begin{cases} f_C(\mathcal{N}_1\langle C_1, u, v \rangle / (r_{uv}, u+1), \mathcal{N}_2\langle C_2, w, x \rangle) + 1 \\ f_C(\mathcal{N}_1\langle C_1, u, v \rangle / (r_{uv}, v-1), \mathcal{N}_2\langle C_2, w, x \rangle) + 1 \\ f_C(\mathcal{N}_1\langle C_1, u, v \rangle, \mathcal{N}_2\langle C_2, w, x \rangle / (r_{wx}, w+1)) + 1 \\ f_C(\mathcal{N}_1\langle C_1, u, v \rangle, \mathcal{N}_2\langle C_2, w, x \rangle / (r_{wx}, x-1)) + 1 \\ \min_{\substack{u < a \le t_1 \le b < v \\ w \le c \le t_2 \le d \le x \\ D(a) \cup D(b) = D(c) \cup D(d)}} f_B(a, b, c, d) \end{cases}$$

Note that the first four entries create a composite join subnetwork, as one root inherits the prime subnetworks going out of its contracted child. In the event that one of these contractions is inadmissible, a cycle is created and the base case returns ∞.

An f_B entry needs to consider the contractions at the bottom of the cycle, plus the join subnetworks created at the bottom, plus the contractions needed on the two lateral parts of the cycles. The latter requires comparing subnetworks of the form $\mathcal{N}_1^L\langle i, j \rangle$ and $\mathcal{N}_2^L\langle p, q \rangle$, for some pairs i, j and p, q of internal nodes to a cycle of respectively $\mathcal{N}_1$ and $\mathcal{N}_2$, and belonging to the same side of that cycle. For clarity, we denote by $f_L(i, j, p, q)$ the minimum number of contractions needed to obtain a common contraction of $\mathcal{N}_1^L\langle i, j \rangle$ and $\mathcal{N}_2^L\langle p, q \rangle$). We get

$$f_B(a, b, c, d) = |b - a| + |d - c| + f_C(\mathcal{N}_1^B\langle a, b \rangle, \mathcal{N}_2^B\langle c, d \rangle) + \min \begin{cases} f_L(u+1, a-1, w+1, c-1) + f_L(v-1, b+1, x-1, d+1) \\ f_L(u+1, a-1, x-1, d+1) + f_L(v-1, b+1, w+1, c-1) \end{cases}$$

The latter minimization occurs because we do not know which sides of the cycles should be put in correspondence. It only remains to describe how to compute these lateral subnetworks, i.e. entries $f_L(i, j, p, q)$. To start with, if $L(\mathcal{N}_1^L\langle i, j \rangle) \neq L(\mathcal{N}_2^L\langle p, q \rangle)$, we set $f_L(i, j, p, q) = +\infty$. Then, in the case of equality, if $i > j$ (which implies $p > q$ given the constraint $L(\mathcal{N}_1^L\langle i, j \rangle) = L(\mathcal{N}_2^L\langle p, q \rangle)$, the corresponding subnetworks are empty and $f_L(i, j, p, q) = 0$. For i, j on the same side of the $\mathcal{N}_1$ cycle and p, q on the same side of the $\mathcal{N}_2$ cycle, we consider two cases: either there is an edge $(k, k+1)$ between i, j and a

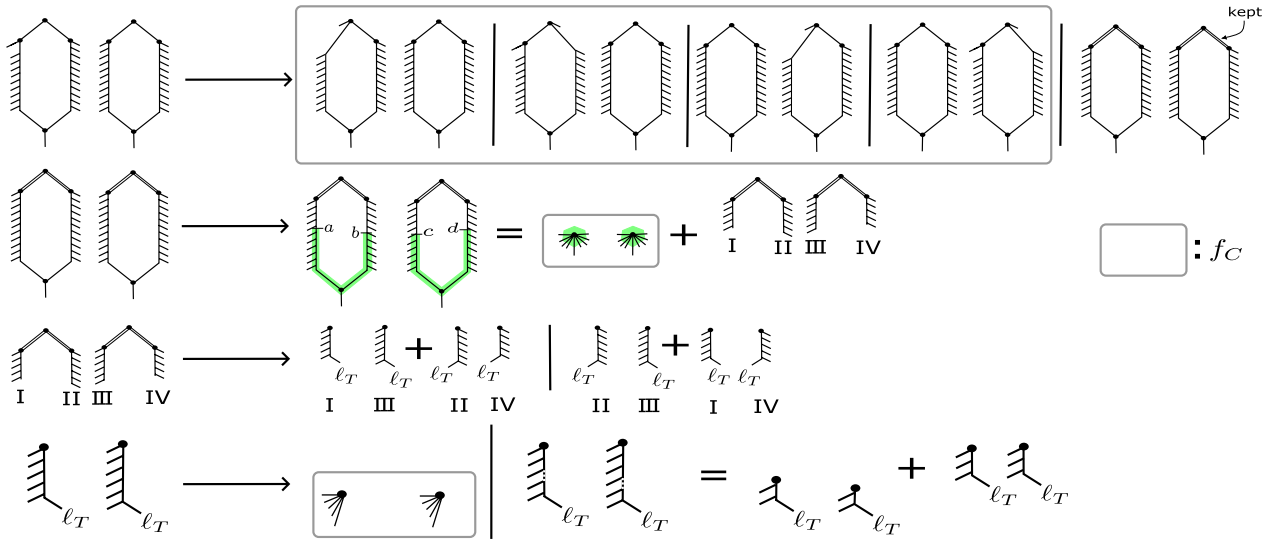

Fig. 10 The handling of Case 4. The gray boxes represent calls to f_C entries, and vertical bars separate possible cases. First row: The left and right children of the first network are $u+1, v-1$, and the left and right children of the second are $w+1, x-1$. In the gray box, the four ways of contracting an edge incident to a root (in bold edges is the untouched network). If we do not apply those, we keep these edges. Second row: we choose a, b, c, d and contract the corresponding paths and defer the computation to $f_B(a,b,c,d)$, which requires optimizing over the $\mathcal{N}_1^B, \mathcal{N}_2^B$ bottom networks, plus what is above a, b, c, d. Third row: the two ways of matching the lateral networks, considered by f_B. Fourth row: given two lateral networks for an f_L entry, we either contract the whole lateral paths to get $\mathcal{N}^S$ networks, or split the paths and solve two smaller lateral networks

corresponding edge $(k', k'+1)$ between p, q that can be kept, or not (in the proofs, the extra leaf ℓ_T is there to enforce that if $(k, k+1)$ is kept on the $\mathcal{N}_1'$ lateral path, the corresponding edge in $\mathcal{N}_2'$ is on the other lateral path and not elsewhere). In the former case, the edge splits the computation into two subproblems, and in the latter we must contract everything. This yields:

$$f_L(i,j,p,q) = \min \begin{cases} \min\limits_{\substack{i \le k < j \\ p \le k' < q}} f_L(i,k,p,k') + f_L(k+1,j,k'+1,q), \\ f_C(\mathcal{N}_1^S\langle i,j\rangle, \mathcal{N}_2^S\langle p,q\rangle) + |j-i| + |p-q| \end{cases}$$

Note that if $i = j$, there is no k such that $i \le k < j$ and the above recurrence only consider the second line, which makes a call to an f_C entry. The chain of recurrences can stop here, since the $\mathcal{N}^S$'s are join subnetworks, which can make use of other f_C entries. Although not trivial, we can argue that a polynomial number of entries is sufficient to compute $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) = f_C(\mathcal{N}_1, \mathcal{N}_2)$.

Theorem 18

The value of $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2)$ between two weakly galled trees without degree-2 nodes can be computed in $O(n^5)$, with $n = \max(|V(\mathcal{N}_1)|, |V(\mathcal{N}_2)|)$.

Proof

The proof is split in two parts: first we analyze the time complexity, and then argue the correctness of the algorithm.

► **Time complexity.**

We first argue the time complexity to compute $f_C(\mathcal{N}_1, \mathcal{N}_2) = \delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2)$.

We assume a recursive algorithm that starts with a request to calculate $f_C(\mathcal{N}_1, \mathcal{N}_2)$, and computes only smaller entries when they are required by our recurrences. We store in memory any f_C, f_P entry and f_L that is computed for future reuse (memoization). Note that f_B has a slightly different status: it is a mere calculus intermediate between f_P and f_L. Thanks to memoization, when f_C,f_P, or f_L entries require other entries, we assume they can access their value in time $O(1)$. The set of all possible composite join subnetworks has exponential size, so it is important to emphasize that we only calculate the necessary f_C,f_P, and f_L entries encountered. We need to bound the number of such entries to compute through the whole execution, multiplied by the time needed to compute each individual entry.

Time to compute all f_P entries Let us start with f_P. We argue that f_P is called on $O(n^2)$ entries, each

requiring at most time $O(n^2)$ to compute. To start with, note that f_P is only called by f_C on pairs of matching sub-networks, which in particular have the same set of leaves. As detailed below, the equality of the sets of leaves is crucial to bounding the number of entries of f_P that need to be stored. The entries of f_P are of four types, which we argue separately.

- **Case 1** - $f_P(\mathcal{N}_1\langle\overset{\bullet}{u}\rangle, \mathcal{N}_2\langle\overset{\bullet}{v}\rangle)$: by equality of the sets of leaves, $D(u) = D(v)$. By Proposition 15, there are at most 2 nodes v verifying this constraint for a given u. Since the number of u's is at most the number of nodes of $\mathcal{N}_1$, and since each u can have at most 2 corresponding v's, there are therefore $O(n)$ of these entries. Each of them takes $O(1)$ to compute (only need to fetch the required f_C value).
- **Case 2** - $f_P(\mathcal{N}_1\langle\overset{\bullet}{u}\rangle, \mathcal{N}_2\langle C, v, w\rangle)$: here, u, v, w must verify $D(u) = D(v+1) \cup D(w-1)$. Note that we may have $v+1 \neq w-1$ (in which case they form a 2-clade pair) or $v+1 = w-1$ (in which case they are both the reticulation of C). In any case, by Proposition 15, given u, there is only $O(1)$ possible 2-clade-pair v, w verifying $D(u) = D(v+1) \cup D(w-1)$. Therefore, given u, there is only $O(1)$ possible C, u, v such that $\mathcal{N}_1\langle\overset{\bullet}{u}\rangle, \mathcal{N}_2\langle C, v, w\rangle$ is a pair of matching sub-networks. As f_P is only called on such pairs, this yields $O(n)$ possible f_P entries of this type that need to be stored, each of them being computed in $O(1)$ time.
- **Case 3** - $f_P(\mathcal{N}_1\langle C, u, v\rangle, \mathcal{N}_2\langle\overset{\bullet}{w}\rangle)$: takes time $O(n)$ by a symmetric argument.
- **Case 4** - $f_P(\mathcal{N}_1\langle C_1, u, v\rangle, \mathcal{N}_2\langle C_2, w, x\rangle)$: since we must have $D(u+1) \cup D(v-1) = D(w+1) \cup D(x-1)$, as above, there is only $O(1)$ possibilities for w, x given u, v. As argued for case 2, this gives $O(n^2)$ possible entries. For each of them, the first 4 cases of the minimization only require time $O(1)$. For the fifth case which needs to minimize over a, b, c, d, the constraint $D(a) \cup D(b) = D(c) \cup D(d)$ restricts (by Proposition 15) c, d to one possibility given a, b. However, this time a, b, c, d are not given together as input. In order to find c, d given a, b in time $O(1)$, we use a pre-processing with $O(n^5)$ complexity, iterating over all pairs of 2-clade pairs a, b, c, d, comparing in time $O(n)$ their sets of descendants, and storing for each a, b the $O(1)$ pairs c, d such that $D(a) \cup D(b) = D(c) \cup D(d)$. Given that, for each of the $O(n^2)$ entries to f_P, the minimization over the possible a, b, c, d takes time $O(n^2)$ to compute, corresponding to the iteration over a, b.

Overall, the time complexity of computing f_P is dominated by Case 4, with an $O(n^4)$ complexity. However, this complexity is reached by assuming a pre-processing step taking $O(n^5)$.

Time to compute all f_L entries We continue with f_L. Let us recall that $f_L(i,j,p,q)$ is defined as $\delta_{\text{MCC}}(\mathcal{N}_1^L\langle i,j\rangle, \mathcal{N}_2^L\langle p,q\rangle)$. We call an entry to f_L *valid* if $L(\mathcal{N}_1^L\langle i,j\rangle) = L(\mathcal{N}_2^L\langle p,q\rangle)$. We argue that given i, j, there is only one pair p, q such that i, j, p, q is a valid entry to f_L. Indeed, let x, y be a pair of nodes of $\mathcal{N}_2$ such that $L(\mathcal{N}_2^L\langle p,q\rangle) = L(\mathcal{N}_2^L\langle x,y\rangle)$. We show that, necessarily, $p = x$ and $q = y$. We restrict our attention to the case $L(\mathcal{N}_2^L\langle p,q\rangle) \neq \emptyset$, as otherwise computing $\delta_{\text{MCC}}(\mathcal{N}_1^L\langle i,j\rangle, \mathcal{N}_2^L\langle p,q\rangle)$ is trivial.

Let us call C the cycle to which p, q belong, r its root, and t its reticulation. Likewise, C' is the cycle containing x, y, and r', t' are its root and reticulation. Note that p, q and x, y are all internal nodes in their respective cycles (potentially with $p = q$ or $x = y$). As internal nodes, and because $\mathcal{N}_1$ and $\mathcal{N}_2$ are assumed not to have degree-2 nodes, p, q reach leaves (through an edge outside of C) that the reticulation t cannot reach. Likewise t reaches leaves that children of p, q outside of C cannot reach. Note that these leaves are not in $L(\mathcal{N}_2^L\langle p,q\rangle) = D(p) \backslash D(q+1)$.

Suppose $C \neq C'$, and let us first examine the case where r' is reachable from p, i.e. where the cycle C' descends from p in $\mathcal{N}_2$. We further distinguish the cases wether r' is equal to p, is reachable from the child of p within the cycle C (i.e. descends from $p+1$), or is reachable from a child of p outside the cycle C.

If r' descends from $p+1$, then p has a child outside C that reaches leaves not reached by r', and thus the sets of leaves $L(\mathcal{N}_2^L\langle p,q\rangle)$ and $L(\mathcal{N}_2^L\langle x,y\rangle)$ cannot be equal. If $p = r'$, then x, y are on a cycle rooted at p, and $D(x) \backslash D(y+1)$ excludes leaves reached by p that must be in $D(p) \setminus D(q+1)$. Finally, if r' descends from a child of p outside of C, the same argument applies.

Therefore r' is not reachable from p, and thus not reachable from q. Note also that if t, x or y are reachable from p, then r' must be reachable from p, which we just showed cannot happen.

In particular x and p must be incomparable. However, since we supposed $L(\mathcal{N}_2^L\langle p,q\rangle) = L(\mathcal{N}_2^L\langle x,y\rangle)$, they have at least a leaf ℓ that they can both reach. Consider a path P from p to ℓ an a path P' from x to ℓ, and the first node s of P that belongs to $P \cap P'$. Such a node exists, as the

parent of ℓ is on both P and P'. In addition, it must be a reticulation (as otherwise its only parent would be an earlier node of P in $P \cap P'$). Likewise, consider the paths from the root to x and p, and the last node belonging to both. This node is the root of a cycle, whose reticulation is s, containing both x and p. This is in contradiction with $C \neq C'$ in weakly-galled networks.

Therefore $C = C'$. If x, y is not on the same side as p, q, then x reaches some leaves p, q cannot reach, which is a contradiction. Therefore x, y and p, q are on the same side of C. If p is strictly above x, it can reach leaves that x cannot, therefore $p = x$. Likewise, if q is strictly above y, too many leaves are excluded. Therefore $p = x$ and $q = y$

Since i, j determine p, q, this gives a $O(n^2)$ number of valid entries to f_L.

However, some of the calls from f_P to f_L might not be valid. To recognize valid entries in time $O(1)$, we iterate as a pre-processing step over the $O(n^4)$ possible entries of f_L, check for each of them in $O(n)$ whether it is valid or not by comparing reachable leaf sets ($O(n^5)$ total complexity), and store the result. The total time spent in f_L is then $O(1)$ for each invalid entry, and $O(n^2)$ for each valid entry, per the iteration over k, k'. A rough upper bound on the number of invalid entries is $O(n^4)$, and we have argued an $O(n^2)$ upper bound on the number of valid entries, yielding $O(n^4)$ in total. The worst-case complexity of computing f_L entries is therefore $O(n^5)$, dominated by the pre-processing.

Time to compute all f_C entries To finish, we analyze the complexity of computing f_C entries. The first step of f_C is to compare (in $O(n)$) the leaf-sets of the two input networks, to check the validity of the entry. We argue that there are $O(n^2)$ valid entries to f_C. To see this, note that f_C is only called by f_P and f_L (in addition to the original main call). The first 3 cases of f_P, and f_L call f_C a constant number of times, and only when they are handling a valid entry. As analyzed before, there are $O(n^2)$ of these calls. The only case yielding more calls to f_C is the last minimization case of Case 4 of f_P, where (through the intermediate f_B), $O(n^4)$ calls are given to f_C. However, these are of the form $f_C(\mathcal{N}_1^B\langle a, b\rangle, \mathcal{N}_2^B\langle c, d\rangle)$. These calls are only valid if $D(a) \cup D(b) = D(c) \cup D(d)$. As argued for f_P, and thanks to Proposition 15, there is only one possibility for c, d given a, b and this constraint. There are therefore $O(n^4)$ calls to f_C, with $O(n)$ spent on each of them to check the validity. However, only $O(n^2)$ of them are valid and pass this step. It remains to account for the complexity of applying Rules 1 &2 exhaustively, which we argue in the following can be done in $O(n^3)$ for each entry (yielding $O(n^5)$ in total).

To begin with, we assume that as a first step, we compute and store the $D(u)$ sets of both networks $\mathcal{N}_1'$ and $\mathcal{N}_2'$. This can be done naively in time $O(n^3)$ by taking, for each node u, the union of the $D(v)$ sets for the children v of u (so, n nodes, each requiring $O(n)$ unions in time $O(n)$ each, yielding cubic time). Then, we assume that these are stored in a data structure that allows querying for the existence of a 1-clade or 2-clade in time $O(n)$. This can be achieved in multiple ways, for instance storing them in a hash table (which takes time $O(n)$ to hash one set of leaves), or by bucket-sorting the elements of the clades by their leaf identifiers, then interpreting them as strings and storing them in a trie in which each node contains $O(n)$ pointers, which can be done without adding to the $O(n^3)$ complexity.

Let us now analyze the complexity of checking whether Rule 1 or 2 applies. There are $O(n)$ 1-clade nodes that are non-reticulation children of r_1, and also $O(n)$ 2-clades that contain a child u of r_1 (to see this, note that u is internal to at most one cycle, and if u, v is a 2-clade pair, then v occurs in at most two 2-clade pairs to evaluate, the other possibly occurring for the other child of r_1 in the cycle). For every such 1-clade or 2-clade, we can query our stored sets of 1-clades and 2-clades of $\mathcal{N}_2'$ to see if it is present, each query taking time $O(n)$. Thus, checking whether one rule should be applied takes time $O(n^2)$. Since we may apply $O(n)$ rules before reaching matching join subnetworks, we spend time up to $O(n^3)$ to reduce the networks.

Overall running time To conclude, we spend time $O(n^5)$ computing f_P (due to the pre-processing), $O(n^5)$ computing f_L (also due to the pre-processing) and $O(n^5)$ computing f_C. This gives the overall worst-case complexity of $O(n^5)$.

► Correctness of the algorithm

We next argue that the recurrences are correct. Let $\mathcal{N}_1', \mathcal{N}_2'$ be join subnetworks of $\mathcal{N}_1$ and $\mathcal{N}_2$, respectively. We show by induction on $|V(\mathcal{N}_1')| + |V(\mathcal{N}_2')|$ that the recurrences given above are correct. To this end, denote by $OPT := OPT(\mathcal{N}_1', \mathcal{N}_2')$ the minimum number of contractions needed to obtain a common contraction $\mathcal{M}$ of $\mathcal{N}_1'$ and $\mathcal{N}_2'$. We show that for any $\mathcal{N}_1', \mathcal{N}_2'$, $f_C(\mathcal{N}_1', \mathcal{N}_2') = OPT$. Through the recursions, it implies also showing that, in the case that $\mathcal{N}_1', \mathcal{N}_2'$ are prime, $f_P(\mathcal{N}_1', \mathcal{N}_2') = OPT$ also holds.

The base cases ($L(\mathcal{N}_1') \neq L(\mathcal{N}_2')$, $\mathcal{N}_1'$ or $\mathcal{N}_2'$ not acyclic, empty networks, or both networks having a single node) are trivial to verify, so we assume that they do not cover $\mathcal{N}_1', \mathcal{N}_2'$. So as an inductive hypothesis, we assume that $OPT(\mathcal{N}_1'', \mathcal{N}_2'') = f_C(\mathcal{N}_1'', \mathcal{N}_2'')$ for any pair of join sub-networks of size smaller than $\mathcal{N}_1', \mathcal{N}_2'$, i.e. $|V(\mathcal{N}_2'')| + |V(\mathcal{N}_1'')| < |V(\mathcal{N}_2')| + |V(\mathcal{N}_1')|$ (with the same hypothesis for f_P when the networks are prime). In other words, the induction hypothesis applies as soon as the size of at least one of the two networks is reduced by 1.

Suppose first that one of the subnetworks is composite. If Rule 1 or Rule 2 applies to either network, we contract k edges incident to either roots until no such rule applies, resulting in join subnetworks $\mathcal{N}_1'', \mathcal{N}_2''$ by Lemma 16. Since these rules are safe by Lemma 14, these k contractions are also required to obtain OPT, implying that $OPT = k + OPT(\mathcal{N}_1'', \mathcal{N}_2'') = k + f_C(\mathcal{N}_1'', \mathcal{N}_2'')$ by induction. Since this is our value of $f_C(\mathcal{N}_1', \mathcal{N}_2')$, this case is verified.

Next suppose that no rule applies. Then by Lemma 17, $\mathcal{N}_1' = \mathcal{N}_1^1 * \ldots * \mathcal{N}_1^p, \mathcal{N}_2 = \mathcal{N}_2^1 * \ldots * \mathcal{N}_2^p$, with $L(\mathcal{N}_1^i) = L(\mathcal{N}_2^i)$. It is easily seen that after applying contractions to each pair $\mathcal{N}_1^i, \mathcal{N}_2^i$ separately to make them isomorphic, one obtains a common contraction of $\mathcal{N}_1', \mathcal{N}_2'$ by identifying the roots of the separate subnetworks. This justifies

$$f_C(\mathcal{N}_1', \mathcal{N}_2') = \sum_{i=1}^{p} f_P(\mathcal{N}_1^i, \mathcal{N}_2^i) \geq OPT.$$

Moreover, the solution with OPT contractions must, at the very least, apply contractions to make each $\mathcal{N}_1^i, \mathcal{N}_2^i$ subnetwork isomorphic, and, because the subnetworks are edge-disjoint and only intersect at the root, using induction justifies

$$OPT \geq \sum_{i=1}^{p} f_P(\mathcal{N}_1^i, \mathcal{N}_2^i) = f_C(\mathcal{N}_1', \mathcal{N}_2').$$

This shows that the f_C entries are correct, assuming that all f_P entries are correct.

Suppose that $\mathcal{N}_1'$ and $\mathcal{N}_2'$ are prime join subnetworks, and consider $f_P(\mathcal{N}_1', \mathcal{N}_2')$. We handle the same set of cases as in the recurrences.

Case 1: $\mathcal{N}_1' = \mathcal{N}_1\langle \overset{\bullet}{u} \rangle, \mathcal{N}_2' = \langle \overset{\bullet}{v} \rangle$. In this case, $\mathcal{N}_1'$ consists of a root r_1 with a single child u, which is the root of a subnetwork of $\mathcal{N}_1$, and $\mathcal{N}_2'$ has a root r_2 with a single child v, which roots a subnetwork of $\mathcal{N}_2$. In that case, we may assume that the edges $r_1 \to u$ and $r_2 \to v$ are kept, so we may ignore them. In other words, we can find a sequence of contractions to make $\mathcal{N}_1\langle u \rangle$ and $\mathcal{N}_2\langle v \rangle$ isomorphic, then attach r_1 and r_2 to the resulting networks to make them isomorphic. Since OPT cannot do better than this, this justifies $f_P(\mathcal{N}_1\langle \overset{\bullet}{u} \rangle, \mathcal{N}_2\langle \overset{\bullet}{v} \rangle) = f_C(\mathcal{N}_1\langle u \rangle, \mathcal{N}_2\langle v \rangle)$. Importantly, we observe that $\mathcal{N}_1\langle u \rangle, \mathcal{N}_2\langle v \rangle$ are join subnetworks of $\mathcal{N}_1, \mathcal{N}_2$, so the corresponding f_C entry used in our recurrences is well-defined.

Case 2: $\mathcal{N}_1' = \mathcal{N}_1\langle \overset{\bullet}{u} \rangle, \mathcal{N}_2' = \mathcal{N}_2\langle C, v, w \rangle$. Here, $\mathcal{N}_1'$ starts with a single edge $r_1 \to u$, while $\mathcal{N}_2'$ starts with a node r_2 that is the root of a cycle, which is the fusion of v and w on cycle C.

Suppose that edge $r_1 \to u$ is conserved in the OPT scenario. Then $\mathcal{N}_2'$ must be modified so that its root r_2 becomes a node with a single child x. This means that the edges of C are contracted so that all children of the cycle nodes that are not on C become a child of x. After applying these contractions in the OPT scenario, r_2 has a single child, which is the root of the subnetwork $\mathcal{N}_2^B\langle C, v+1, w-1 \rangle$, which is a join subnetwork. In this case, we get that OPT is equal to the first case of our recurrence.

So suppose that $r_1 \to u$ is not conserved. Then after applying this contraction in the OPT scenario, by induction we have $OPT = f_C(\mathcal{N}_1\langle u \rangle, \mathcal{N}_2\langle C, v, w \rangle) + 1$. Noting that this f_C entry is well-defined, in this case, OPT is equal to the second case of our recurrence.

Since the recurrence takes the minimum of both cases, we get $f_P(\mathcal{N}_1\langle u \rangle, \mathcal{N}_2\langle C, v, w \rangle) \leq OPT$. Moreover, it is not hard to see that both cases of the recurrence either yield infinity, or result in a valid common contraction, which justify $f_P(\mathcal{N}_1\langle u \rangle, \mathcal{N}_2\langle C, v, w \rangle) \geq OPT$.

Case 3: is symmetric to case 2.

Case 4: $\mathcal{N}_1' = \mathcal{N}_1\langle C_1, u, v \rangle, \mathcal{N}_2' = \mathcal{N}_2\langle C_2, w, x \rangle$. Let t_1, t_2 be the respective reticulations of C_1, C_2. The purpose of the lengthy arguments that follow is to establish that $f_C(\mathcal{N}_1', \mathcal{N}_2') \leq OPT$.

Suppose that the OPT scenario contracts an edge incident to u, v, w, or x, say $(u, u+1)$ (the other cases are identical). Note that $u+1$ could not be a reticulation, as such a contraction would be forbidden. After contracting this edge in $\mathcal{N}_1\langle C_1, u, v \rangle$, we obtain the network $\mathcal{N}_1\langle C_1, u, v \rangle / (u, u+1)$, which is a join subnetwork by Lemma 16. We therefore get by induction that

$OPT = 1 + f_C(\mathcal{N}_1\langle C_1, u, v\rangle/(u, u+1), \mathcal{N}_2\langle C_2, w, x\rangle)$ as in our recurrences.

So, suppose that in the *OPT* scenario, no edge incident to u, v, w, x is contracted. If we let $W = \{W_1, \ldots, W_l\}$ be a partition of $\mathcal{N}_1\langle C_1, u, v\rangle$ that is a witness to the common contraction yielding *OPT*, and $Z = \{Z_1, \ldots, Z_l\}$ the witness partition of $\mathcal{N}_2\langle C_2, w, x\rangle$. By our assumption, $u+1$ and $v-1$ are not in the same set as the root of the first network, and $w+1, x-1$ are not in the same set as the root of the second network. This also means that the respective reticulation t_1, t_2 of C_1, C_2 are also in sets that differ from the roots. Let a, b be nodes of C_1 such that are in the same witness set as t_1, and such that $a-1$ and $b+1$ are not in this set (such a, b exist by the previous sentences). Define c, d as the analogous nodes of C_2, with respect to the set with t_2. Note that any of a and/or b could be equal to t_1, and c and/or d could be equal to t_2.

We get that in the *OPT* scenario, all the nodes on the path from a to b, and from c to d, following our cyclic orderings, are contracted. Denote by $\mathcal{N}_1''$ and $\mathcal{N}_2''$ networks obtained after applying these contractions. Then $OPT = |b-a| + |d-c| + OPT(\mathcal{N}_1'', \mathcal{N}_2'')$. Unfortunately, we cannot use induction by relating the latter term to an f_C entry, since it might not be a join subnetwork as it applies contractions not at the root.

We divide the argument into two subcases: either C_1 was contracted to a single edge, or not. We will assume for now that the entries $f_L(i, j, p, q)$ correctly store the minimum number of contractions needed to obtain a common contraction of the corresponding lateral subnetworks, and we shall prove this later on.

Subcase 1: C_1 **contracted to a single edge** Suppose first that $\mathcal{N}_1''$ consists of the root r_1 with a single child, i.e., C_1 got contracted into a single edge. This is only possible if $a = u+1, b = v-1$, and if C_2 is also contracted to a single edge, and thus that $c = w+1, d = x-1$. In that case, we may ignore the dangling root edges, and the optimal contraction only compares the subnetworks rooted at the nodes obtained from the contraction from a to b and from c to d. These subnetworks are $\mathcal{N}_1^B\langle a, b\rangle$ and $\mathcal{N}_2^B\langle c, d\rangle$, so that $OPT(\mathcal{N}_1'', \mathcal{N}_2'') = OPT(\mathcal{N}_1^B\langle a, b\rangle, \mathcal{N}_2^B\langle c, d\rangle)$. Because these subnetworks are join subnetworks, this is equal to $f_C(\mathcal{N}_1^B\langle a, b\rangle, \mathcal{N}_2^B\langle c, d\rangle)$ by induction. In other words, $OPT = |b-a| + |d-c| + f_C(\mathcal{N}_1^B\langle a, b\rangle, \mathcal{N}_2^B\langle c, d\rangle)$. Note that these terms are considered in the $f_B(a, b, c, d)$ entry of our recurrence when minimizing over a, b, c, d, and we would like to argue that $f_B(a, b, c, d) \leq OPT$. However, the f_B entry also adds a term from the f_L entries, so let us focus on the latter.

The considered entries $f_L(i, j, p, q)$ have $i = u+1, j = a-1$ or $i = v-1, j = b-1$. Either way, we are in the case where $a = u+1, b = v-1$, and j does not descend from i. According to our definition of the f_L terms, they all evaluate to 0 in $f_B(a, b, c, d)$. It follows that according to the recurrences,

$$f_C(\mathcal{N}_1', \mathcal{N}_2') \leq f_B(a, b, c, d) = |b-a| + |d-c| + f_C(\mathcal{N}_1^B\langle a, b\rangle, \mathcal{N}_2^B\langle c, d\rangle) + 0 = OPT$$

If $\mathcal{N}_2''$ consists of a root with a single child, a symmetric argument yields the same conclusion.

Subcase 2: C_1 **not contracted to a single edge.** Let us next suppose that in $\mathcal{N}_1''$, C_1 has not been contracted to a single edge, and that in $\mathcal{N}_2''$, C_2 is not contracted to a single edge. Then $\mathcal{N}_1''$ has root r_1 with the two children $u+1, v-1$, which remain in the optimal contraction, and $\mathcal{N}_2''$ has root r_2 with children $w+1, x-1$. Moreover, the parents of the respective reticulations after contracting the $a-b$ and $c-d$ paths are $a-1, b+1$ and $c-1, d+1$. This means that in the *OPT* scenario, we must make $|b-a| + |d-c|$ contractions, then make the bottom subnetworks $\mathcal{N}_1^B\langle a, b\rangle, \mathcal{N}_2^B\langle c, d\rangle$ isomorphic, then choose corresponding lateral subnetworks and make those isomorphic as well.

Noting that our expression for $f_B(a, b, c, d)$ incorporates all of the above, we would like to argue that

$$f_B(a, b, c, d) \leq OPT.$$

The main difficulty is that f_B uses the lateral subnetworks $\mathcal{N}_1^L$ and $\mathcal{N}_2^L$, which have an extra leaf that is not considered by *OPT* (and, as we shall see later on, this extra leaf is necessary for the recurrences to work properly). Consider the minimum common contraction $\mathcal{M}$ in which C_1 is not contracted to a single edge. Consider the witness structures $\mathcal{W}$ and $\mathcal{W}'$ of $\mathcal{N}_1\langle C_1, u, v\rangle$ and $\mathcal{N}_2\langle C_2, w, x\rangle$ that yield the optimal common contraction $\mathcal{M}$. For a node $y \in V(\mathcal{M})$, W_y and W_y' denote the part of $\mathcal{W}$ and $\mathcal{W}'$ that got contracted into y. Because we assume that $r_1 \to u+1$ and $r_1 \to v-1$ are not contracted, $\mathcal{M}$ has a root r with two children, with $W_r = \{r_1\}$ and $W_r' = \{r_2\}$. Also, r is the root of a cycle of $\mathcal{M}$ with reticulation t, and we must have $a, b \in W_t$ and $c, d \in W_t'$.

Let $r = z_0, z_1, z_2, \ldots, z_l = t$, $l \geq 1$, be one of the paths of the cycle of $\mathcal{M}$ rooted at r, such that $u+1 \in W_{z_1}$ (such a path must exist). We infer that $a-1 \in W_{z_{l-1}}$. We must then have either $w+1 \in W_{z_1}', c-1 \in W_{z_{l-1}}'$

or $x-1 \in W'_{z_1}, d+1 \in W'_{z_{l-1}}$. Assume that the former holds (the latter leads to an identical argument).

Denote by $V_1\langle u+1, a-1\rangle$ the set of all nodes of $\mathcal{N}_1$ reached by $u+1$ but not a, and denote $V_2\langle w+1, c-1\rangle$ analogously. Notice that the subnetwork induced by $V_1\langle u+1, a-1\rangle$ is the same as $\mathcal{N}_1^L\langle u+1, a-1\rangle$, except that the latter has the extra leaf child ℓ_T attached to $a-1$. Because edges $r_1 \to u+1$ and $a-1 \to a$ are kept, and because these edges cut $V_1\langle u+1, a-1\rangle$ from the rest of the network, any set of $\mathcal{W}$ that contains nodes $V_1\langle u+1, a-1\rangle$ only contains nodes from that set $V_1\langle u+1, a-1\rangle$ (i.e., no overlap with the exterior). The same holds for $V_2\langle w+1, c-1\rangle$.

This means that $\{W_i \in \mathcal{W} : W_i \subseteq V_1\langle u+1, a-1\rangle\}$ and $\{W'_i \in \mathcal{W} : W'_i \subseteq V_2\langle w+1, c-1\rangle\}$ form a witness structure to a common contraction of the subnetworks induced by $V_1\langle u+1, a-1\rangle$ and $V_2\langle w+1, c-1\rangle$. Moreover, $a-1 \in W_{z_{l-1}}$ and $c-1 \in W'_{z_{l-1}}$, meaning that both sets contract to the same node z_{l-1}. If we add a new leaf ℓ_T that is a child of $a-1, c-1$ in these induced subgraphs, and ℓ_T as a child of z_{l-1}, we see that that the same W_i and W'_i sets still form a witness structure of these networks.

All this setup allows us to conclude that if we use the optimal witness structure yielding $\mathcal{M}$, and restrict its subsets to the internal nodes on the lateral networks $\mathcal{N}_1^L\langle u+1, a-1\rangle$, $\mathcal{N}_2^L\langle w+1, c-1\rangle$, we get a witness structure of the latter two networks (even with the extra leaf). It follows that to make these lateral subnetworks isomorphic, the *OPT* scenario needs at least as many contractions as for the pair of networks $\mathcal{N}_1^L\langle u+1, a-1\rangle$, $\mathcal{N}_2^L\langle w+1, c-1\rangle$. The same holds for the lateral networks on the other side of the cycle. If we assume that the f_L entries correctly store the distances for all the lateral networks, we deduce

$$\begin{aligned} OPT \geq\ & |b-a| + |c-d| + \\ & f_C(\mathcal{N}_1^B\langle a, b\rangle, \mathcal{N}_2^B\langle c, d\rangle) + \\ & f_L(u+1, a-1, w+1, c-1) \\ & + f_L(v-1, b+1, x-1, d+1) \end{aligned}$$

which is greater or equal to $f_B(a, b, c, d)$.

We do need to assume that the recurrence equations for computing f_L entries (equal by definition to the minimum number of contractions needed to reach a common contraction of $\mathcal{N}_1^L\langle i, j\rangle$ and $\mathcal{N}_2^L\langle p, q\rangle$) are correct. So, we initiate a separate proof specifically for the $f_L(i, j, p, q)$ entries. We want to show that $f_L(i, j, p, q)$ is equal to the number OPT_L of contractions needed to achieve a common contraction $\mathcal{M}_L$ of $\mathcal{N}_1^L\langle i, j\rangle$ and $\mathcal{N}_2^L\langle p, q\rangle$.

Consider first the cases of $f_L(i, j, p, q)$ that do not make any recursive calls to other f_L entries. Notice that this occurs when $i = j$ or $p = q$, since there are no possible values of k or k' to iterate over. Suppose that $i = j$. Then, the root of $\mathcal{N}_1^L\langle i, j\rangle$ has, as children, all the children of i in $\mathcal{N}_1$ except $i+1$, and it also has the extra leaf ℓ_T as a child. Note that in this case, $\mathcal{N}_1^L\langle i, j\rangle$ is the same as $\mathcal{N}_1^S\langle i, j\rangle$, but with the extra leaf ℓ_T attached to the root. In $\mathcal{N}_2^L\langle p, q\rangle$, it is q that has this extra leaf child ℓ_T. To reach a common contraction, ℓ_T must be a child of the root of $\mathcal{N}_2^L\langle p, q\rangle$ and thus all the edges on the path from p to q must be contracted. Doing this results in the subnetwork $\mathcal{N}_2^S\langle p, q\rangle$, with the extra leaf added. At this point, that leaf ℓ_T is a child of the root in both networks and it can be ignored, and we have to compute a maximum common contraction between $\mathcal{N}_1^S\langle i, j\rangle$ and $\mathcal{N}_2^S\langle p, q\rangle$. The recurrence for $f_L(i, j, p, q)$ returns the number of contractions on these two networks (we invoke the induction hypothesis for the correctness of the f_C value), plus the number $|p-q|$ of contractions on the $p-q$ path. This covers the case $i = j$, and the case $p = q$ is identical.

So assume that $i \neq j$ and $p \neq q$. We follow the typical strategy for showing the correctness of dynamic programming equations, namely showing both $f_L(i, j, p, q) \leq OPT_L$ and $f_L(i, j, p, q) \geq OPT_L$. As these equations also involve f_C terms, the induction hypothesis on f_C we also be invoked.

Let $\mathcal{M}_L$ be such an optimal common contraction, and suppose that to get to $\mathcal{M}_L$, some edge $k \to k+1$ on the path between i and j is not contracted. Let us choose the smallest k such that this is the case. It means that all nodes from u to k (included) are contracted into a single point. We argue that there must exist k' in $[p, q]$ such that the edge $k' \to k'+1$ in $\mathcal{N}_2^L\langle p, q\rangle$ is not contracted either, and such that the edge separates the leaves in the same manner as $k \to k+1$.

In terms of $\mathcal{M}_L$-witness-structure, all nodes on the $i-k$ path are part of the same W_r, for some $r \in V(\mathcal{M}_L)$. On the other hand, $k+1$ belongs to W_s with $s \neq r$. Let A be the set of leaves reached by i but not $k+1$, and $B = L(\mathcal{N}_1^L\langle i, j\rangle) \setminus A$.

Let $\mathcal{W}'$ be a $\mathcal{M}_L$-witness structure in $\mathcal{N}_2^L\langle p, q\rangle$. To start with, since i is the root of $\mathcal{N}_1^L\langle i, j\rangle$, r is the root of $\mathcal{M}_L$, and the root p of $\mathcal{N}_2^L\langle p, q\rangle$ must be in W'_r, the correspondent of W_r in $\mathcal{W}'$.

Then, in $\mathcal{M}_L$, all paths from the leaf ℓ_T to r must encounter s, as it is the case in $\mathcal{N}_1^L\langle i,j\rangle$ that all paths from ℓ_T to W_r go through W_s. Therefore, in $\mathcal{N}_2^L\langle p,q\rangle$, as $p \in W'_r$, and ℓ_T is a child of q, there must be some node $\tilde{k} \in W'_s$ on the path from p to q. We pick $\tilde{k}$ as close to p as possible. To finish, we argue that $\tilde{k}-1$ must be in W'_r. Indeed, if it is not, then it belongs to some other witness set W_l. In $\mathcal{M}_L$, l would separate r from s, which is not the case. To conclude, edge $\tilde{k}-1 \to \tilde{k}$ is kept in $\mathcal{N}_2^L\langle p,q\rangle$. We also see that in $\mathcal{M}_L$, the set of leaves reached by s is B, from which it follows that the set of leaves reached by $\tilde{k}$ in $\mathcal{N}_2^L\langle p,q\rangle$ is also B. Since this is also the set of leaves reached by $k+1$, we get that both edges $k \to k+1$ and $k' \to k'+1 = \tilde{k}-1 \to \tilde{k}$ separate the lateral networks into two other lateral sub-networks with the same sets of leaves.

From this, it follows that $\mathcal{M}_L$ can be obtained by combining a common contraction of the lateral networks $\mathcal{N}_1^L\langle i,k\rangle$ and $\mathcal{N}_2^L\langle p,k'\rangle$, and of the networks $\mathcal{N}_1^L\langle k+1,j\rangle$ and $\mathcal{N}_2^L\langle k'+1,q\rangle$ (the fact that each of those lateral sub-networks has an extra leaf can be argued to not change the optimality as before). These common contractions cannot be greater than their optimal sizes, reached with respectively $f_L(i,k,p,k')$ and $f_L(k+1,j,k'+1,q)$ contractions. Therefore, we get in this case that $f_L(i,j,p,q) \leq OPT_L$.

If an edge on the $p-q$ path is not contracted, a symmetric argument yields the same result. Finally, if no edge on either $i-j$ or $p-q$ path is conserved, we may assume that they are all contracted, resulting in $\mathcal{N}_1^S$ and $\mathcal{N}_2^S$ sub-networks on both sides, and a number of contractions that is $|j-i|+|p-q|+f_C(\mathcal{N}_1^S\langle i,j\rangle, \mathcal{N}_2^S\langle p,q\rangle)$ by the induction hypothesis on f_C (note that the f_C entry is well-defined since $\mathcal{N}_1^S$ and $\mathcal{N}_2^S$ sub-networks are join sub-networks). Again, this case is considered in our recurrences and we get $f_L(i,j,p,q) \leq OPT_L$.

In all cases, we get $f_L(i,j,p,q) \leq OPT_L$. To finish the argument, it is not too difficult to see inductively that each case in the minimization of the f_L recurrence either yields $+\infty$, or yields an integer that corresponds to a scenario leading to a common contraction. This gives the complementary bound $f_L(i,j,p,q) \geq OPT_L$. □

Conclusion and discussion

We have generalized the original formulation of the Robinson-Foulds distance on phylogenetic trees, based on edge contractions and expansions, to phylogenetic networks. This required novel versions of contractions/expansions capable of creating and suppressing cycles. Both our measures d_{CE} and δ_{MCC} connect the whole space of networks on the same leaves. However, whereas in the case of trees, d_{CE} is equivalent to finding a maximum common contraction, that it is not the case for networks, making both measure require their own separate studies.

Our work poses several questions and open problems. From a *parameterized complexity* point of view, our reductions prove `Para-NP-hardness` (i.e. NP-hardness for a constant value of the parameter) of computing δ_{MCC} with "size of the common contraction plus maximum degree" as a parameter, as well as the number of leaves. However, other parameters, such as the *level* of the input networks, their *treewidth* [46], their *scanwidth* [47] as well as the maximum number of allowed contractions, could be of interest and will be the subject of future work. Regarding treewidth, a starting point could be the $O(|H|^{tw(G)+1}|G|)$ algorithm for determining if a bounded-degree graph H is a contraction of another graph G [35]. Although the *bounded-degree* constraint is unlikely to help, the complexity of computing δ_{MCC} on *binary networks* (in+out degree ≤ 3) is also open.

Funding
The authors acknowledge financial support from the NSERC/FRQNT NOVA program (Grant #327657).

Data availability
Not applicable.

Declarations

Competing interests
The authors declare no Conflict of interest nor Conflict of interest.

Received: 31 October 2024 Accepted: 15 May 2025
Published online: 01 October 2025

References

1. Abby SS, Tannier E, Gouy M, Daubin V. Lateral gene transfer as a support for the tree of life. Proc Natl Acad Sci. 2012;109(13):4962–7.
2. Koonin EV, Makarova KS, Aravind L. Horizontal gene transfer in prokaryotes: quantification and classification. Annu Rev Microbiol. 2001;55(1):709–42.
3. Ellstrand NC, Schierenbeck KA. Hybridization as a stimulus for the evolution of invasiveness in plants? Proc Natl Acad Sci. 2000;97(13):7043–50.
4. Huson DH, Rupp R, Scornavacca C. Phylogenetic networks: concepts, algorithms and applications. Cambridge University Press; 2010.
5. Bandelt H-J, Forster P, Röhl A. Median-joining networks for inferring intraspecific phylogenies. Mol Biol Evol. 1999;16(1):37–48.
6. Boc A, Diallo AB, Makarenkov V. T-rex: a web server for inferring, validating and visualizing phylogenetic trees and networks. Nucleic Acids Res. 2012;40(W1):573–9.
7. Cámara PG, Levine AJ, Rabadan R. Inference of ancestral recombination graphs through topological data analysis. PLoS Comput Biol. 2016;12(8):1005071.

8. Gusfield D, Eddhu S, Langley C. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. J Bioinform Comput Biol. 2004;2(01):173–213.
9. Solís-Lemus C, Ané C. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. PLoS Genet. 2016;12(3):1005896.
10. Forster P, Forster L, Renfrew C, Forster M. Phylogenetic network analysis of sars-cov-2 genomes. Proc Natl Acad Sci. 2020;117(17):9241–3.
11. Sánchez-Pacheco SJ, Kong S, Pulido-Santacruz P, Murphy RW, Kubatko L. Median-joining network analysis of sars-cov-2 genomes is neither phylogenetic nor evolutionary. Proc Natl Acad Sci. 2020;117(23):12518–9.
12. Forster P, Forster L, Renfrew C, Forster M. Reply to explaining phylogenetic network analysis of sars-cov-2 genomes. Proc Natl Acad Sci USA. 2020;117(23):12524.
13. Mavian C, Pond SK, Marini S, Magalis BR, Vandamme A-M, Dellicour S, Scarpino SV, Houldcroft C, Villabona-Arenas J, Paisie TK, et al. Sampling bias and incorrect rooting make phylogenetic network tracing of sars-cov-2 infections unreliable. Proc Natl Acad Sci. 2020;117(23):12522–3.
14. Robinson DF, Foulds LR. Comparison of phylogenetic trees. Math Biosci. 1981;53(1–2):131–47.
15. Cardona G, Llabrés M, Rosselló F, Valiente G, et al. The comparison of tree-sibling time consistent phylogenetic networks is graph isomorphism-complete. Sci World J. 2014;1: 254279.
16. Cardona G, Llabrés M, Rosselló F, Valiente G. Metrics for phylogenetic networks I: generalizations of the Robinson-Foulds metric. IEEE/ACM Trans Comput Biol Bioinf. 2008;6(1):46–61.
17. Moret BM, Nakhleh L, Warnow T, Linder CR, Tholse A, Padolina A, Sun J, Timme R. Phylogenetic networks: modeling, reconstructibility, and accuracy. IEEE/ACM Trans Comput Biol Bioinf. 2004;1(1):13–23.
18. Cardona G, Rosselló F, Valiente G. Tripartitions do not always discriminate phylogenetic networks. Math Biosci. 2008;211(2):356–70.
19. Bai A, Erdős PL, Semple C, Steel M. Defining phylogenetic networks using ancestral profiles. Math Biosci. 2021;332: 108537.
20. Cardona G, Pons JC, Ribas G, Martınez Coronado T. Comparison of orchard networks using their extended μ-representation. IEEE/ACM Trans Comput Biol Bioinform. 2024. https://doi.org/10.1109/TCBB.2024.3361390.
21. Cardona G, Llabrés M, Rosselló F, Valiente G. Metrics for phylogenetic networks ii: nodal and triplets metrics. IEEE/ACM Trans Comput Biol Bioinf. 2008;6(3):454–69.
22. Gambette P, Van Iersel L, Jones M, Lafond M, Pardi F, Scornavacca C. Rearrangement moves on rooted phylogenetic networks. PLoS Comput Biol. 2017;13(8):1005611.
23. Bordewich M, Linz S, Semple C. Lost in space? Generalising subtree prune and regraft to spaces of phylogenetic networks. J Theor Biol. 2017;423:1–12.
24. Landry K, Teodocio A, Lafond M, Tremblay-Savard O. Defining phylogenetic network distances using cherry operations. IEEE/ACM Trans Comput Biol Bioinf. 2022;20(3):1654–66.
25. Landry K, Tremblay-Savard O, Lafond M. A fixed-parameter tractable algorithm for finding agreement cherry-reduced subnetworks in level-1 orchard networks. J Comput Biol. 2023.
26. Bordewich M, Semple C. On the computational complexity of the rooted subtree prune and regraft distance. Ann Comb. 2005;8:409–23.
27. DasGupta B, He X, Jiang T, Li M, Tromp J, Zhang L. On computing the nearest neighbor interchange distance. Computing. 1998;23(22):21–6.
28. Klawitter J. Spaces of phylogenetic networks. PhD thesis, ResearchSpace@ Auckland. 2020.
29. Choy C, Jansson J, Sadakane K, Sung W-K. Computing the maximum agreement of phylogenetic networks. Theoret Comput Sci. 2005;335(1):93–107.
30. Zelinka B. Contraction distance between isomorphism classes of graphs. Časopis pro pěstování matematiky. 1990;115(2):211–6.
31. Brouwer AE, Veldman HJ. Contractibility and np-completeness. J Graph Theory. 1987;11(1):71–9.
32. Levin A, Paulusma D, Woeginger GJ. The computational complexity of graph contractions I: polynomially solvable and np-complete cases. Netw Int J. 2008;51(3):178–89.
33. Kamiński M, Paulusma D, Thilikos DM. Contractions of planar graphs in polynomial time. In: Algorithms–ESA 2010: 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I 18, 2010;pp. 122–133. Springer.
34. Belmonte R, Golovach PA, Hof P, Paulusma D. Parameterized complexity of three edge contraction problems with degree constraints. Acta Informatica. 2014;51(7):473–97.
35. Matoušek J, Thomas R. On the complexity of finding iso-and other morphisms for partial k-trees. Discret Math. 1992;108(1–3):343–64.
36. Kriege N, Kurpicz F, Mutzel P. On maximum common subgraph problems in series-parallel graphs. Eur J Comb. 2018;68:79–95.
37. Akutsu T, Melkman AA, Tamura T. Improved hardness of maximum common subgraph problems on labeled graphs of bounded treewidth and bounded degree. Int J Found Comput Sci. 2020;31(02):253–73.
38. Diestel R. Graph Theory. Springer; 2016.
39. Luks EM. Isomorphism of graphs of bounded valence can be tested in polynomial time. J Comput Syst Sci. 1982;25(1):42–65.
40. Hellmuth M, Schaller D, Stadler PF. Clustering systems of phylogenetic networks. Theory Biosci. 2023;142(4):301–58.
41. Garey MR, Johnson DS. Computers and Intractability vol. 174. freeman San Francisco. 1979.
42. Darmann A, Döcker J. On a simple hard variant of not-all-equal 3-sat. Theoret Comput Sci. 2020;815:147–52.
43. Antony D, Cao Y, Pal S, Sandeep R. Switching classes: characterization and computation. 2024. arXiv preprint arXiv:2403.04263
44. Rosselló F, Valiente G. All that glisters is not galled. Math Biosci. 2009;221(1):54–9.
45. Agrawal A, Kanesh L, Saurabh S, Tale P. Paths to trees and cacti. Theoret Comput Sci. 2021;860:98–116.
46. Marchand B, Ponty Y, Bulteau L. Tree diet: reducing the treewidth to unlock fpt algorithms in RNA bioinformatics. Algorithms Mol Biol. 2022;17(1):8.
47. Berry V, Scornavacca C, Weller M. Scanning phylogenetic networks is np-hard. In: SOFSEM 2020: Theory and Practice of Computer Science: 46th International Conference on Current Trends in Theory and Practice of Informatics, SOFSEM 2020, Limassol, Cyprus, January 20–24, 2020, Proceedings 46, 2020;pp. 519–530. Springer.

Publisher's Note