Finding Maximum Common Contractions Between Phylogenetic Networks

Bertrand Marchand
Department of Computer Science, University of Sherbrooke, Canada

Nadia Tahiri
Department of Computer Science, University of Sherbrooke, Canada

Olivier Tremblay-Savard
Department of Computer Science, University of Manitoba, Winnipeg, Canada

Manuel Lafond
Department of Computer Science, University of Sherbrooke, Canada

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.

2012 ACM Subject Classification Applied computing → Bioinformatics; Theory of computation → Problems, reductions and completeness

Keywords and phrases Phylogenetic networks, contractions, algorithms, weakly galled trees

Digital Object Identifier 10.4230/LIPIcs.WABI.2024.16

Related Version *Full Version*: `https://arxiv.org/abs/2405.16713`

1 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, 33] and hybridization [23], 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 [30]. 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 [6, 9, 13, 28]. However, given the same dataset, 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 [24, 44, 25]).

24th International Workshop on Algorithms in Bioinformatics (WABI 2024).
Editors: Solon P. Pissis and Wing-Kin Sung; Article No. 16; pp. 16:1–16:24
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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 datasets, and identify outliers or similarities among them. In the case of trees, one of the most established metrics is the Robinson-Foulds distance [42], due to the simplicity of its definition, its ease of computation, and the fact that it also yields a maximum *common structure* between the trees as a by-product. 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 [42] presented it as the minimum number of edge contractions and expansions required to go from one tree to the other.

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 [16]. Nonetheless, many 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* [30, 14], *softwired cluster* [30] and *tri-partition* [41, 14] distances. There exists however even binary networks that are distinct while exhibiting the same sets of clusters [30, Figure 6.28], and likewise for the tripartition distance [18]. In addition, in the case of the soft-wired 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 [30, Section 6.14.3]. Another group of proposals for metrics on phylogenetic networks includes the μ-distance [5, 17], as well as the *nodal distance* and *triplet distance* [15]. One could loosely describe them as comparing the "connectivity" induced on leaves by the networks topologies. While they are polynomial to compute, they are only valid on sub-classes of networks, namely *orchard networks* [17] and *time-consistent tree-child networks* [15]. A notable exception is the *subnetwork distance* [30, 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, [26]), *subtree prune and regraph* (SPR, [10]) or *cherry-picking operations* [35] on orchard networks. While cherry-picking operations have been successfully used to define and compute metrics on orchard networks [36], computing distances based on NNI+SPR operations is NP-hard even for trees [11, 21]. 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 [36] and SPR [32] allow to define such a maximum common structure. Another example of this philosophy is the computation of *maximum agreement sub-networks* [19], 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 leafsets 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 distance, and the latter a *semi-distance*, 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-distance. We prove its NP-hardness under different conditions: when restricted to bounded-degree networks and a common contraction of size ≤ 3, 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.

All omitted proofs may be found in the Appendix, or in the full online pre-print.

Related works outside of phylogenetics. Perhaps the closest related work in its philosophy is [45], 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 graph [12, 37, 31, 7, 40]. 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 [34, 3]. 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 unlabelled tree is the contraction of another unlabelled tree is NP-hard [40], whereas this is polynomial-time solvable when leaves are labeled, as this is equivalent to computing the Robinson-Foulds distance.

2 Contraction-Expansion distance

2.1 Preliminaries

This article uses standard directed graph terminology (nodes, edges, in-neighbors, out-neighbors, acyclic graphs, ...), as can for instance be found in [22]. 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 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 $V(\mathcal{N}), E(\mathcal{N}), \mathcal{I}(\mathcal{N})$, 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*,

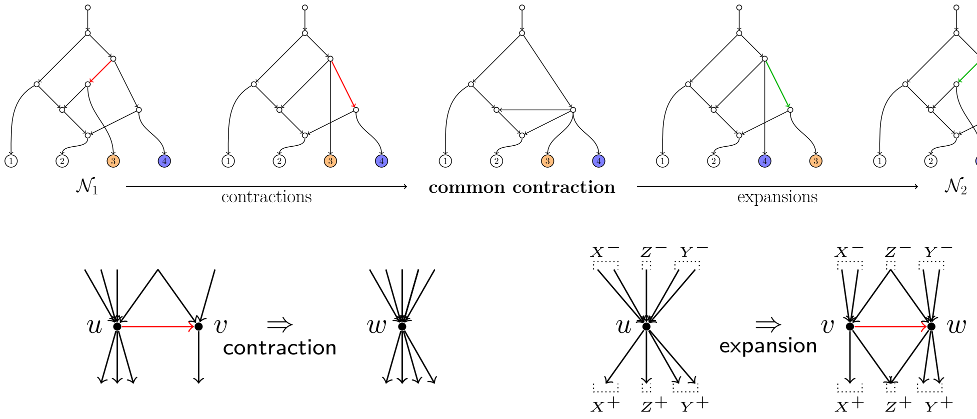

Figure 1 (top) The transformation of a network into another through contractions, followed by expansions. Midway, the intermediate network is a maximum common contraction, which can be achieved from $\mathcal{N}_2$ by reversing the expansions. (bottom) Illustration of edge contractions (left) and expansions (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.

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 [38], 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, [30, 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 [5] for the μ-distance.

2.2 Contractions and expansions

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

▶ **Definition 2** (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. $\Gamma^-_{\mathcal{N}'}(w) = \Gamma^-_{\mathcal{N}}(u) \cup \big(\Gamma^-_{\mathcal{N}}(v) \setminus \{u\}\big)$*; and*
3. $\Gamma^+_{\mathcal{N}'}(w) = \big(\Gamma^+_{\mathcal{N}}(u) \setminus \{v\}\big) \cup \Gamma^+_{\mathcal{N}}(v)$*.*

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

[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.

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. The following proposition characterizes the set of admissible contractions in a network. This result can also be found in [29, Lemma 3].

▶ **Proposition 3.** *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 $u \to v$.*

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 4** (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 X^-, Y^-, Z^- (resp. X^+, Y^+, Z^+) is a partition of $\Gamma^-_{\mathcal{N}}(u)$ (resp. $\Gamma^+_{\mathcal{N}}(u)$) 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 [42], 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 5** (contraction-expansion distance). *Given two networks $\mathcal{N}_1$ and $\mathcal{N}_2$ over the same leafsets, the contraction-expansion distance $d_{CE}(\mathcal{N}_1, \mathcal{N}_2)$ is the minimum length of a sequence of admissible contractions and expansions transforming $\mathcal{N}_1$ into $\mathcal{N}_2' \sim \mathcal{N}_2$.*

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

▶ Remark 7. 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 $d_{\mathrm{CE}}(\mathcal{N}_1, \mathcal{N}_2) \leq \delta_{\mathrm{MCC}}(\mathcal{N}_1, \mathcal{N}_2)$.

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 $\delta_{\mathrm{MCC}}(\mathcal{N}_1, \mathcal{N}_2) \leq k'$, with $k' = |\mathcal{I}(\mathcal{N}_1)| + |\mathcal{I}(\mathcal{N}_2)| - 2k$. We next establish important properties of d_{CE} and δ_{MCC}.

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

Proof sketch. Let r be the root of $\mathcal{N}$. Because $\mathcal{N}$ is an acyclic digraph, r has a child u such that no other child of r reaches u (here, u would be next to r in a topological ordering). Thus, there is no other path from r to u, meaning that $r \to u$ is admissible by Proposition 3. We can thus keep doing contractions from the root until a star is achieved. ◀

The corollary below uses Proposition 8 and the star network to show that one can always transform a network into another.

▶ **Corollary 9.** *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 $\mathcal{N}_2' \sim \mathcal{N}_2$.*

▶ Remark 10. To finish, note that the length of the sequence described above is $|\mathcal{I}(\mathcal{N}_1)| + |\mathcal{I}(\mathcal{N}_2)| - 2$, which is an upper bound on the distance.

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 11.** *d_{CE} is a metric on the set of phylogenetic networks with the same leafsets.*

As for δ_{MCC}, it does verify identity ($\delta_{\mathrm{MCC}}(\mathcal{N}_1, \mathcal{N}_2) = 0$ if and only if their common contraction is themselves, i.e., they are isomorphic), positivity, and symmetry (by definition). However, Figure 2 (A) shows an example in which $\delta_{\mathrm{MCC}}(\mathcal{N}_1, \mathcal{N}_3) = 10$, while $\delta_{\mathrm{MCC}}(\mathcal{N}_1, \mathcal{N}_2) + \delta(\mathcal{N}_2, \mathcal{N}_3) = 4 + 2 = 6$, 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 Figure 2 (B) an example of two networks for which $d_{\mathrm{CE}}(\mathcal{N}_1, \mathcal{N}_2) = 2$, whereas $\delta_{\mathrm{MCC}}(\mathcal{N}_1, \mathcal{N}_2) = 6$.

▶ Remark 12. 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 unlabelled, undirected graphs with the same number of nodes has also been shown [45].

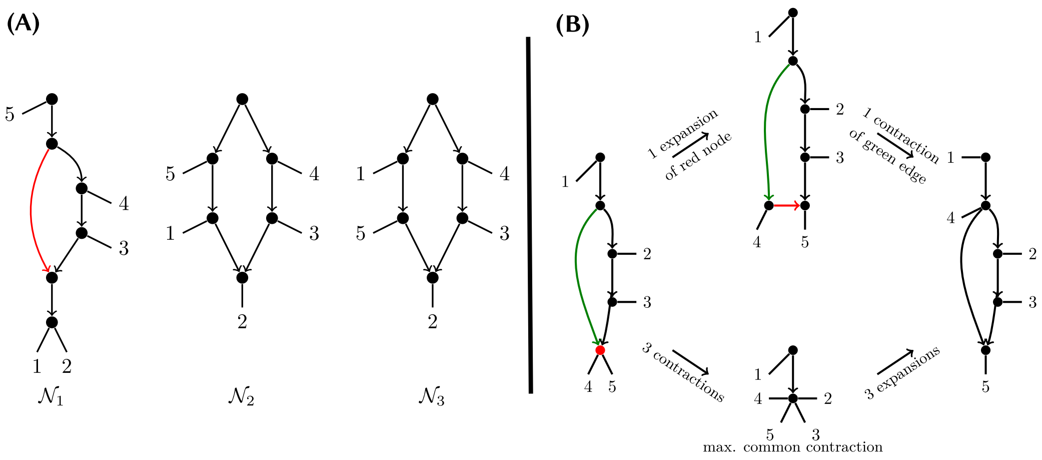

Figure 2 (A) Examples of 3 networks for which $\delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_3) > \delta_{\text{MCC}}(\mathcal{N}_1, \mathcal{N}_2) + \delta_{\text{MCC}}(\mathcal{N}_2, \mathcal{N}_3)$. Indeed, one can check that there is a common contraction of size 4 between $\mathcal{N}_1$ and $\mathcal{N}_2$, 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 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).

2.4 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 [7]) 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 set of nodes W_v of $\mathcal{N}$ that got contracted "together" to become v. Contractions impose W_v to be connected if we ignore edge directions (i.e., weakly connected), and the W_v's must partition $V(\mathcal{N})$.

▶ **Definition 13.** *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 $\mathcal{W} = \{W_u \mid u \in V(\mathcal{M})\}$ of the internal nodes of $\mathcal{N}$ such that:*

(1) *$\forall u \in V(\mathcal{M})$, W_u is a weakly connected sub-graph of $\mathcal{N}$.*
(2) *$\forall u, v \in V(\mathcal{M}) \times V(\mathcal{M})$, $(u, v) \in E(\mathcal{M})$ 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 $V(\mathcal{N})$ and $V(\mathcal{M})$ 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 8.

▶ **Proposition 14.** *Let $\mathcal{M}$ and $\mathcal{N}$ two networks over the same set of leaves. Then $\mathcal{M}$ is a contraction of $\mathcal{N}$ if and only if there exists a $\mathcal{M}$-witness structure in $\mathcal{N}$.*

Proof sketch. In the ($\Rightarrow$) direction, if $v \in V(\mathcal{M})$ is a node of the contraction, reversing the contractions "re-expands" v to a witness set W_v, which must be weakly connected as each expansion adds an edge. In the ($\Leftarrow$) direction, in each witness set W_u, the first edge in a topological ordering is always admissible to contraction, and since it is connected, we may contract it to a single node. ◀

3 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 [27]).

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. 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 [20], 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) [4, Proposition 5.1]. As for the second one, it shows that MCNC remains hard even when one of the network 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.

3.1 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 Figure 3 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 $\{\ell_2, \ell'_2\}$ 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 Figure 3). 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 Figure 3), called $W^O[r_1]$, ensuring that $r_1 \rightsquigarrow u_S$,

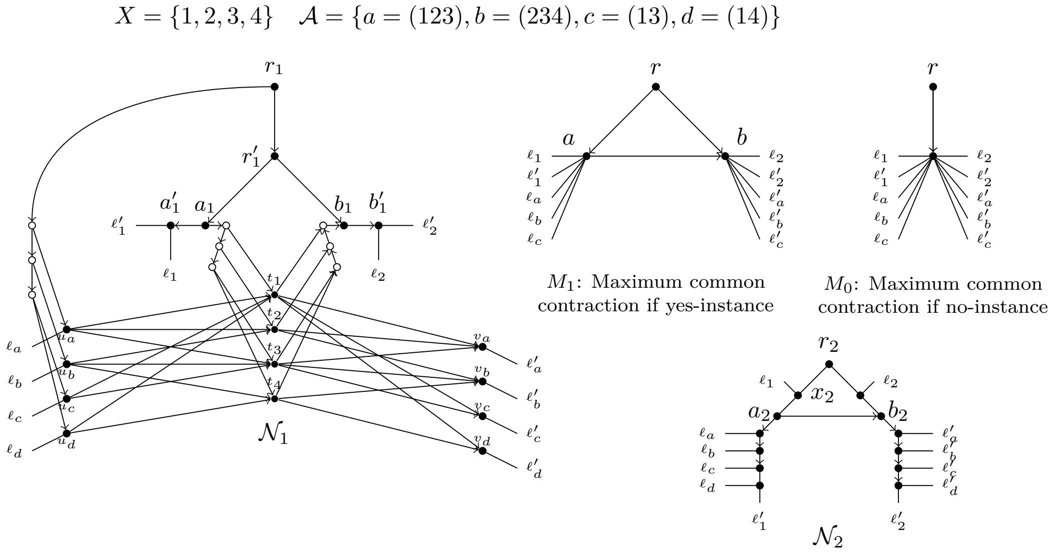

Figure 3 Illustration of our first reduction from SET SPLITTING to MAXIMUM COMMON NETWORK CONTRACTION on an example.

$\forall S \in \mathcal{A}$. 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 $\forall i \in [m-1]$, $p_i \to u_{S_i}$, $r_1 \to p_1$, and p_{m-1} is connected to S_m in addition to S_{m-1}. Likewise, $\forall x \in X$, we implement a directed path between a_1 and t_x, and between t_x and b_1, using intermediary nodes called $W^O[a_1]$ and $W^I[b_1]$, also arranged in a path of size $|X|-1$, with each node connected to one t_x (except the last one, connected to 2). Their purpose 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 $x \in S$.

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 has two 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 $\{\ell'_S \mid S \in \mathcal{A}\} \cup \{\ell_2, \ell'_2\}$.

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 Figure 3) if and only if $(X, \mathcal{A})$ is splittable.

▶ **Theorem 15.** *MAXIMUM COMMON NETWORK CONTRACTION is NP-hard, even when restricted to $k = 3$*

Proof sketch. Because ℓ_1, ℓ'_1 and ℓ_2, ℓ'_2 share a parent in $\mathcal{N}_1$, it follows that $\mathcal{N}_2$ must be contracted to M_1 to reach any common contraction with $\mathcal{N}_1$. The question is then whether M_1 is a possible contraction. We consider therefore the existence conditions of a M_1-witness structure. The essence of the proof is that to reach this witness structure, the u_S vertices must be in the same witness W_a set as a_1 and a'_1. That witness set must form a weakly connected subgraph, and the t_x must be used to connect each u_S with a_1, a'_1 (as r_1, r'_1 must belong to another witness set). A similar situation holds for the v_S vertices and b_1, b'_1, which must be in the same witness set W_b. Thus, t_x must be split between W_a and W_b, and this split may either yield a split for $(X, \mathcal{A})$, or be inferred from an existing 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 [20]), we get the following.

▶ **Theorem 16.** *Maximum Common Network Contraction is NP-hard, even with $k = 3$ and networks of degree ≤ 10.*

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 [4, Proposition 5.1]:

▶ **Theorem 17.** *Assuming the Exponential Time Hypothesis, Maximum Common Network Contraction cannot be solved in time $2^{o(|V(\mathcal{N}_1)|+|E(\mathcal{N}_1)|)}$.*

3.2 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 Figure S1 for an illustration). As for $\mathcal{N}_1$, it is composed of 4 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 $S \in \mathcal{A}$, 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 18.** *Maximum Common Network Contraction is NP-hard, even on networks with 5 leaves, one of them being a tree and $k = 4$.*

4 On weakly galled-trees, clades, and contraction sequences

When computing the classical Robinson-Foulds distance, the fact that trees are closed under contractions is quite useful, since after contracting an edge corresponding to a clade not in the other tree, the result is still a tree and we can repeat the procedure. Perhaps the simplest class beyond trees are *galled trees*, in which no two cycles share a node, but this network class is not closed under contractions. As a starting point for the investigation of polynomial-time algorithms to MCNC, 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 [43], the following properties were shown on weakly galled trees: (1) every reticulation of $\mathcal{N}$ has an in-degree of exactly 2, and is the reticulation of exactly one cycle; (2) 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 [2].

▶ **Proposition 19.** *If $\mathcal{M}$ is a contraction of a weakly galled tree $\mathcal{N}$, then $\mathcal{M}$ is also a weakly galled tree.*

One can verify that a contraction $c(u, v, w)$ cannot increase the number of reticulations. Therefore, by Proposition 19, 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.

Another difficulty of working on networks is that in trees, the sets $D(u)$ of clades in a contraction are contained in the set of clades of the original tree. In other words, a contraction in a tree T cannot create a clade that did not exist before. This is not true even in weakly galled trees, as new clades can be created by contraction (see Figure S2 in Appendix). For our algorithms, it is necessary to design an appropriate notion of clade that avoids this problem. We 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.*
- *(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 Figure 2: $\{1, 2, 3, 4\}$ is a 1-clade of $\mathcal{N}_1$, whose 1-clade vertex 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. 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 preserved under contractions.

▶ **Proposition 20** (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 $\Sigma_2(\mathcal{M}) \subseteq \Sigma_2(\mathcal{N})$.*

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. 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 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 21.** *Rule 1 and Rule 2 are safe.*

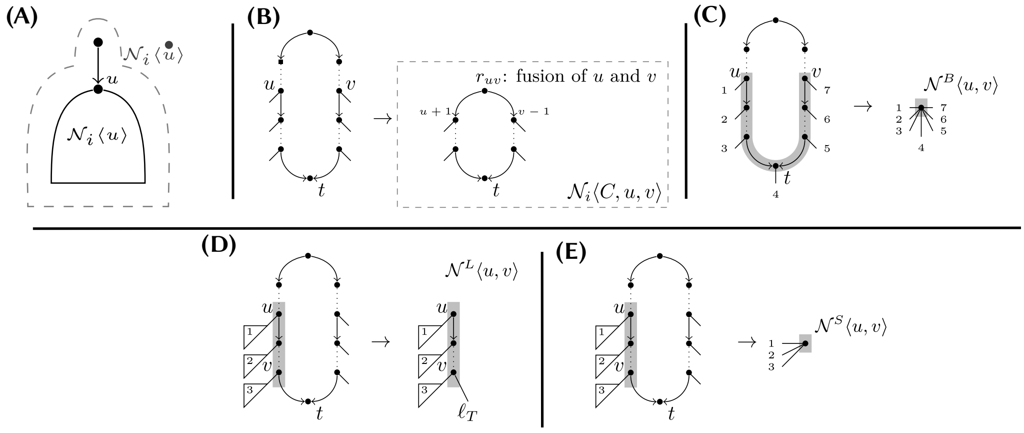
(A)
$\mathcal{N}_i\langle \overset{\bullet}{u}\rangle$
u
$\mathcal{N}_i\langle u\rangle$
(B)
u
v
t
r_{uv}: fusion of u and v
$u+1$
$v-1$
t
$\mathcal{N}_i\langle C, u, v\rangle$
(C)
u
v
1
2
3
4
5
6
7
t
$\mathcal{N}^B\langle u, v\rangle$
1
2
3
4
5
6
7
(D)
u
1
2
3
v
t
$\mathcal{N}^L\langle u, v\rangle$
u
1
2
3
v
ℓ_T
(E)
u
1
2
3
v
t
$\mathcal{N}^S\langle u, v\rangle$
1
2
3

Figure 4 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 $\mathcal{N}_i\langle C, u, v\rangle$.

We end this section by noting 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. An example is given in Figure S2 (Appendix). A common 2-clade may also need to be removed.

5 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 polynomial. 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 $u \leq_C w$ 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 $u \leq_C r$ and $r \leq_C u$. For $u \leq v$, we denote by $|v-u|$ the number of edges on the path from u to v, following the cyclic ordering. If $v < u$, 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 Figure 4.

- $\mathcal{N}\langle u\rangle$ and $\mathcal{N}\langle \overset{\bullet}{u}\rangle$ (Figure 4 (A)): 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.
- $\mathcal{N}\langle C, u, v\rangle$ (Figure 4 (B)): for C a cycle of $\mathcal{N}$ with reticulation t, and $u < t < v$, $\mathcal{N}\langle C, u, v\rangle$ 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 root multiple cycles). In this case, the cycle C is kept as is, but if u has children outside of C, they are removed.

- $\mathcal{N}^B\langle u, v\rangle$ (Figure 4 (C)): 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*.
- $\mathcal{N}^L\langle u, v\rangle$ (Figure 4 (D)): for u, v a pair of non-reticulation 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 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*.
- $\mathcal{N}^S\langle u, v\rangle$ (Figure 4 (E)): 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 leafsets. 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, alternatively, 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 22.** *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 $\mathcal{N}$.*

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 23.** *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 $i \in [p]$.*

A dynamic programming algorithm

We next describe our dynamic programming algorithm. Its overall workflow is sketched in Figure 5, relying on the concept of join subnetwork. The present section details the "decomposition" and "recursive equations" displayed in Figure 5.

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, whereas $f_C(\mathcal{N}_1', \mathcal{N}_2')$

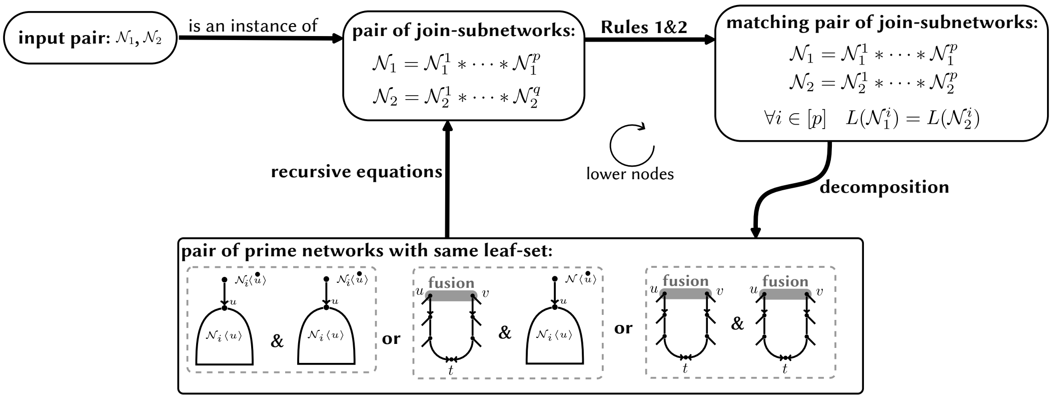
input pair: $\mathcal{N}_1, \mathcal{N}_2$
is an instance of
pair of join-subnetworks:
$\mathcal{N}_1 = \mathcal{N}_1^1 * \cdots * \mathcal{N}_1^p$
$\mathcal{N}_2 = \mathcal{N}_2^1 * \cdots * \mathcal{N}_2^q$
Rules 1&2
matching pair of join-subnetworks:
$\mathcal{N}_1 = \mathcal{N}_1^1 * \cdots * \mathcal{N}_1^p$
$\mathcal{N}_2 = \mathcal{N}_2^1 * \cdots * \mathcal{N}_2^p$
$\forall i \in [p] \quad L(\mathcal{N}_1^i) = L(\mathcal{N}_2^i)$
recursive equations
lower nodes
decomposition
pair of prime networks with same leaf-set:
$\mathcal{N}_i\langle\overset{\bullet}{u}\rangle$
u
$\mathcal{N}_i\langle u\rangle$
&
$\mathcal{N}_i\langle\overset{\bullet}{u}\rangle$
u
$\mathcal{N}_i\langle u\rangle$
or
fusion
u
v
t
&
$\mathcal{N}\langle\overset{\bullet}{u}\rangle$
u
$\mathcal{N}_i\langle u\rangle$
or
fusion
u
v
t
&
fusion
u
v
t

Figure 5 Overview of our algorithm for weakly galled trees.

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 $f_C(\mathcal{N}_1', \mathcal{N}_2') = f_P(\mathcal{N}_1', \mathcal{N}_2')$. 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 22, 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 23, 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: For $u \in V(\mathcal{N}_1)$, $v \in V(\mathcal{N}_2)$: $f_P(\mathcal{N}_1\langle\overset{\bullet}{u}\rangle, \mathcal{N}_2\langle\overset{\bullet}{v}\rangle) = f_C(\mathcal{N}_1\langle u\rangle, \mathcal{N}_2\langle v\rangle)$. This is because both networks start with a single edge $r_u \to u$ and $r_v \to v$ (respectively). 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: 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:

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

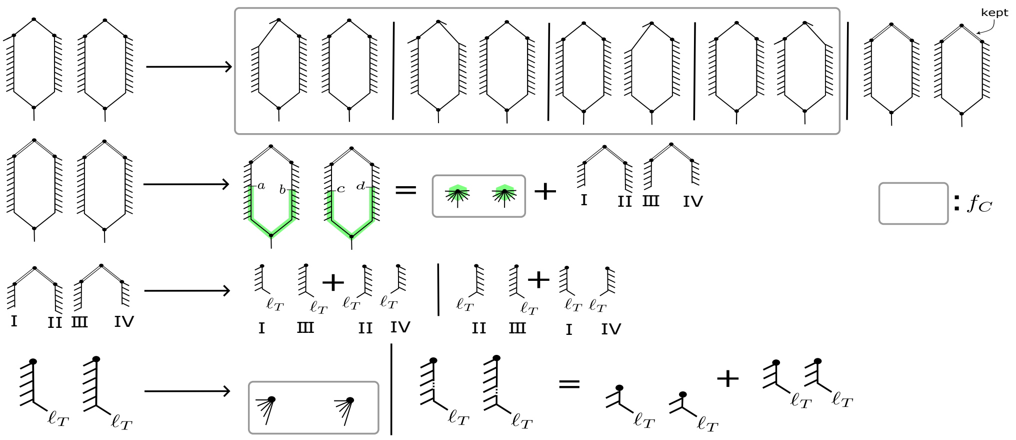

Figure 6 The handling of Case 4. The gray boxes represent calls to f_C entries, and vertical bars separate possible cases. First row: The left and right children of the first network are $u+1, v-1$, and the left and right children of the second are $w+1, x-1$. In the gray box, the four ways of contracting 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.

Here, $\mathcal{N}_1\langle\dot{u}\rangle$ 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, or (b) the edge is contracted.

Case 3: For a cycle C of $\mathcal{N}_1$, $u, v \in V(\mathcal{N}_1)$ and $w \in V(\mathcal{N}_2)$, the computation of $f_P(\mathcal{N}_1\langle C, u, v\rangle, \mathcal{N}_2\langle\dot{w}\rangle)$ is symmetric to the previous case.

Case 4: This is the most complicated case, which is illustrated in Figure 6. 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 $w, x \in C_2$, with respective cycle reticulations t_1 and 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:

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

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 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 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:

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

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 $\delta_{\text{MCC}}(\mathcal{N}_1,\mathcal{N}_2) = f_C(\mathcal{N}_1,\mathcal{N}_2)$. As detailed in the full online pre-print, where the proof of the Theorem below is given, a rough analysis of the worst-case complexity of this computation yields $O(n^9)$, with $n = \max(\mathcal{N}_1,\mathcal{N}_2)$.

▶ **Theorem 24.** *The value of $\delta_{MCC}(\mathcal{N}_1,\mathcal{N}_2)$ between two weakly galled trees can be computed in polynomial time.*

6 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 measures 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* [39], their *scanwidth* [8] 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 [40]. 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. To finish, the *diameter* of a measure is often important in practice, i.e. obtaining the largest possible value between two elements, as it allows normalization and comparison between heterogeneous datasets. Establishing the diameters for the d_{CE} and δ_{MCC} is left as an open problem.

References

1 Sophie S Abby, Eric Tannier, Manolo Gouy, and Vincent Daubin. Lateral gene transfer as a support for the tree of life. *Proceedings of the National Academy of Sciences*, 109(13):4962–4967, 2012.

2 Akanksha Agrawal, Lawqueen Kanesh, Saket Saurabh, and Prafullkumar Tale. Paths to trees and cacti. *Theoretical Computer Science*, 860:98–116, 2021.

3 Tatsuya Akutsu, Avraham A Melkman, and Takeyuki Tamura. Improved hardness of maximum common subgraph problems on labeled graphs of bounded treewidth and bounded degree. *International Journal of Foundations of Computer Science*, 31(02):253–273, 2020.

4 Dhanyamol Antony, Yixin Cao, Sagartanu Pal, and RB Sandeep. Switching classes: Characterization and computation. *arXiv preprint arXiv:2403.04263*, 2024.

5 Allan Bai, Péter L Erdős, Charles Semple, and Mike Steel. Defining phylogenetic networks using ancestral profiles. *Mathematical Biosciences*, 332:108537, 2021.

6 Hans-Jurgen Bandelt, Peter Forster, and Arne Röhl. Median-joining networks for inferring intraspecific phylogenies. *Molecular biology and evolution*, 16(1):37–48, 1999.

7 Rémy Belmonte, Petr A Golovach, Pim van't Hof, and Daniël Paulusma. Parameterized complexity of three edge contraction problems with degree constraints. *Acta Informatica*, 51(7):473–497, 2014.

8 Vincent Berry, Celine Scornavacca, and Mathias Weller. 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*, pages 519–530. Springer, 2020.

9 Alix Boc, Alpha Boubacar Diallo, and Vladimir Makarenkov. T-rex: a web server for inferring, validating and visualizing phylogenetic trees and networks. *Nucleic acids research*, 40(W1):W573–W579, 2012.

10 Magnus Bordewich, Simone Linz, and Charles Semple. Lost in space? generalising subtree prune and regraft to spaces of phylogenetic networks. *Journal of theoretical biology*, 423:1–12, 2017.

11 Magnus Bordewich and Charles Semple. On the computational complexity of the rooted subtree prune and regraft distance. *Annals of combinatorics*, 8:409–423, 2005.

12 Andries Evert Brouwer and Henk Jan Veldman. Contractibility and np-completeness. *Journal of Graph Theory*, 11(1):71–79, 1987.

13 Pablo G Cámara, Arnold J Levine, and Raul Rabadan. Inference of ancestral recombination graphs through topological data analysis. *PLoS computational biology*, 12(8):e1005071, 2016.

14 Gabriel Cardona, Mercè Llabrés, Francesc Rosselló, and Gabriel Valiente. Metrics for phylogenetic networks i: Generalizations of the robinson-foulds metric. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1):46–61, 2008.

15 Gabriel Cardona, Mercè Llabrés, Francesc Rosselló, and Gabriel Valiente. Metrics for phylogenetic networks ii: Nodal and triplets metrics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(3):454–469, 2008.

16 Gabriel Cardona, Mercè Llabrés, Francesc Rosselló, Gabriel Valiente, et al. The comparison of tree-sibling time consistent phylogenetic networks is graph isomorphism-complete. *The Scientific World Journal*, 2014, 2014.

17 Gabriel Cardona, Joan Carles Pons, Gerard Ribas, and Tomas Martınez Coronado. Comparison of orchard networks using their extended μ-representation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–8, 2024. doi:10.1109/TCBB.2024.3361390.

18 Gabriel Cardona, Francesc Rosselló, and Gabriel Valiente. Tripartitions do not always discriminate phylogenetic networks. *Mathematical Biosciences*, 211(2):356–370, 2008.

19 Charles Choy, Jesper Jansson, Kunihiko Sadakane, and Wing-Kin Sung. Computing the maximum agreement of phylogenetic networks. *Theoretical Computer Science*, 335(1):93–107, 2005.

20 Andreas Darmann and Janosch Döcker. On a simple hard variant of not-all-equal 3-sat. *Theoretical Computer Science*, 815:147–152, 2020.

21 Bhaskar DasGupta, Xin He, Tao Jiang, Ming Li, John Tromp, and Louxin Zhang. On computing the nearest neighbor interchange distance. *Computing*, 23(22):21–26, 1998.

22 Reinhard Diestel. *Graph Theory*. Springer, 2016.

23 Norman C Ellstrand and Kristina A Schierenbeck. Hybridization as a stimulus for the evolution of invasiveness in plants? *Proceedings of the National Academy of Sciences*, 97(13):7043–7050, 2000.

24 Peter Forster, Lucy Forster, Colin Renfrew, and Michael Forster. Phylogenetic network analysis of sars-cov-2 genomes. *Proceedings of the National Academy of Sciences*, 117(17):9241–9243, 2020.

25 Peter Forster, Lucy Forster, Colin Renfrew, and Michael Forster. Reply to sanchez-pacheco et al., chookajorn, and mavian et al.: explaining phylogenetic network analysis of sars-cov-2 genomes. *Proceedings of the National Academy of Sciences of the United States of America*, 117(23):12524, 2020.

26 Philippe Gambette, Leo Van Iersel, Mark Jones, Manuel Lafond, Fabio Pardi, and Celine Scornavacca. Rearrangement moves on rooted phylogenetic networks. *PLoS computational biology*, 13(8):e1005611, 2017.

27 Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

28 Dan Gusfield, Satish Eddhu, and Charles Langley. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. *Journal of bioinformatics and computational biology*, 2(01):173–213, 2004.

29 Marc Hellmuth, David Schaller, and Peter F Stadler. Clustering systems of phylogenetic networks. *Theory in Biosciences*, 142(4):301–358, 2023.

30 Daniel H Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic networks: concepts, algorithms and applications*. Cambridge University Press, 2010.

31 Marcin Kamiński, Daniël Paulusma, and Dimitrios M Thilikos. 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*, pages 122–133. Springer, 2010.

32 Jonathan Klawitter. *Spaces of phylogenetic networks*. PhD thesis, ResearchSpace@ Auckland, 2020.

33 Eugene V Koonin, Kira S Makarova, and L Aravind. Horizontal gene transfer in prokaryotes: quantification and classification. *Annual Reviews in Microbiology*, 55(1):709–742, 2001.

34 Nils Kriege, Florian Kurpicz, and Petra Mutzel. On maximum common subgraph problems in series–parallel graphs. *European Journal of Combinatorics*, 68:79–95, 2018.

35 Kaari Landry, Aivee Teodocio, Manuel Lafond, and Olivier Tremblay-Savard. Defining phylogenetic network distances using cherry operations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022.

36 Kaari Landry, Olivier Tremblay-Savard, and Manuel Lafond. A fixed-parameter tractable algorithm for finding agreement cherry-reduced subnetworks in level-1 orchard networks. *Journal of Computational Biology*, 2023.

37 Asaf Levin, Daniël Paulusma, and Gerhard J Woeginger. The computational complexity of graph contractions i: Polynomially solvable and np-complete cases. *Networks: An International Journal*, 51(3):178–189, 2008.

38 Eugene M Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of computer and system sciences*, 25(1):42–65, 1982.

39 Bertrand Marchand, Yann Ponty, and Laurent Bulteau. Tree diet: reducing the treewidth to unlock fpt algorithms in rna bioinformatics. *Algorithms for Molecular Biology*, 17(1):8, 2022.

40 Jiří Matoušek and Robin Thomas. On the complexity of finding iso-and other morphisms for partial k-trees. *Discrete Mathematics*, 108(1-3):343–364, 1992.

41 Bernard ME Moret, Luay Nakhleh, Tandy Warnow, C Randal Linder, Anna Tholse, Anneke Padolina, Jerry Sun, and Ruth Timme. Phylogenetic networks: modeling, reconstructibility, and accuracy. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):13–23, 2004.

42 David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. *Mathematical biosciences*, 53(1-2):131–147, 1981.

43 Francesc Rosselló and Gabriel Valiente. All that glisters is not galled. *Mathematical biosciences*, 221(1):54–59, 2009.

44 Santiago J Sánchez-Pacheco, Sungsik Kong, Paola Pulido-Santacruz, Robert W Murphy, and Laura Kubatko. Median-joining network analysis of sars-cov-2 genomes is neither phylogenetic nor evolutionary. *Proceedings of the National Academy of Sciences*, 117(23):12518–12519, 2020.

45 Bohdan Zelinka. Contraction distance between isomorphism classes of graphs. *Časopis pro pěstování matematiky*, 115(2):211–216, 1990.

A Proofs for Section 2 (Contraction-Expansion distance)

▶ **Proposition 8.** *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 (recall that a topological ordering is a permutation $\sigma = (u_1, \dots, u_n)$ of $\mathcal{N}$ such that $(u_i, u_j) \in E(\mathcal{N})$ implies $i < j$, and that such an ordering always exists in a directed acyclic graph). Let σ be any topological ordering of $V(\mathcal{N})$, and let u be the first non-leaf out-neighbor of r that occurs in σ. 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 3, $r \to u$ is admissible. It follows that $\mathcal{N}$ can then be contracted into the star network by picking such an element u as long as there are non-root internal nodes. ◀

▶ **Proposition 14.** *Let $\mathcal{M}$ and $\mathcal{N}$ two networks over the same set of leaves. Then $\mathcal{M}$ is a contraction of $\mathcal{N}$ if and 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 be $C = c_1 \dots c_p$ the contraction sequence yielding $\mathcal{M}$ from $\mathcal{N}$. 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 3, 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 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. ◀

B Proofs for Section 3 (NP-hardness of finding maximum common contractions)

▶ **Theorem 18.** *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 Figure S1, $\mathcal{N}_2$ 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 $\Rightarrow$ $\mathcal{N}_2$ 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 3). Then, since $\forall S \in A$, $S \cap X_1 \neq \emptyset$, there is now 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 $\mathcal{N}_2$.

$\mathcal{N}_2$ contraction of $\mathcal{N}_1$ $\Rightarrow$ $(X, \mathcal{A})$ splittable. Per Proposition 14, $\mathcal{N}_2$ 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) the parent

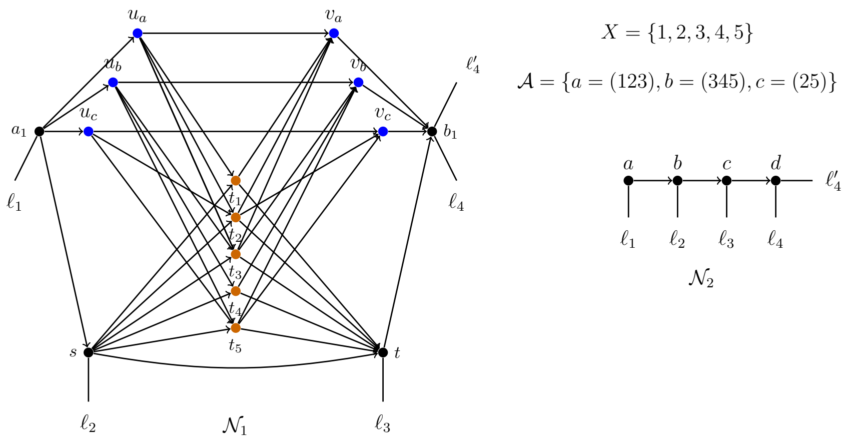

Figure S1 Illustration of our second reduction, also from SET SPLITTING. The blue nodes represent the sets of $\mathcal{A}$, and the orange nodes the elements of X. We use this construction in Theorem 18, to prove that even when restricted to networks on 5 leaves, MAXIMUM COMMON NETWORK CONTRACTION is NP-hard.

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) and (3) 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. 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, whereas it must be 3. Therefore $V_a = \{a_1\}$ 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), $\{u_S\}_{S \in \mathcal{A}} \subseteq V_b$ 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 $(X, \mathcal{A})$. ◀

C Proofs for Section 4 (On weakly galled-trees, clades, and contraction sequences)

▶ **Proposition 20** (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. 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). Let S be a 1-clade of $\mathcal{M}$. We have $S = D_M(x)$ 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, 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 $D_{\mathcal{N}}(x) = D_M(x) = S$, as the contraction does not alter the leaves that x reaches.

If $x = w$, then $S = D_M(x) = D_M(w) = D_{\mathcal{N}}(u)$. Either u was a 1-clade node of $\mathcal{N}$, and S is indeed a 1-clade of $\mathcal{N}$, or u was internal to a cycle in $\mathcal{N}$. Since w is a 1-clade node in $\mathcal{M}$, the only possibility is that u was the only internal node in its cycle, and v was the reticulation of the cycle on which u stood. In this case u, v was a 2-clade pair of $\mathcal{N}$, with $D_{\mathcal{N}}(u) \cup D_{\mathcal{N}}(v) = S$. Therefore, $S \in \Sigma_2(\mathcal{N})$.

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). If $w \notin \{x, y\}$, then we also have that neither x or y are the nodes u and v, and therefore also nodes of $\mathcal{N}$. Since they stand on distinct paths in C in $\mathcal{M}$, this is also the case in $\mathcal{N}$, and x and y are also a 2-clade node pair in $\mathcal{N}$. To finish on this case, we need $D_{\mathcal{N}}(x) \cup D_{\mathcal{N}}(y) = D_{\mathcal{M}}(x) \cup D_{\mathcal{N}}(y)$. We have $D_{\mathcal{N}}(x) \cup D_{\mathcal{N}}(y) \subseteq D_{\mathcal{M}}(x) \cup D_{\mathcal{N}}(y)$ as a path $\mathcal{N}$ from x or y to a leaf still exists (although potentially slightly modified) after the contraction of (u, v). If $D_{\mathcal{N}}(x) \cup D_{\mathcal{N}}(y) \neq D_{\mathcal{M}}(x) \cup D_{\mathcal{N}}(y)$, then the only possibility is that x or y could reach v but not u, and that the contraction of (u, v) allowed to reach new leaves from $D_{\mathcal{N}}(u)$. But if this is the case, then v is a reticulation. Whether it is the reticulation of C or not, x or y can reach u in $\mathcal{N}$, which is a contradiction. Therefore S is 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 u was also internal in C in $\mathcal{N}$. Therefore u, y is a 2-clade node pair, and $S = D_M(w) \cup D_M(y) = D_{\mathcal{N}}(u) \cup D_{\mathcal{N}}(y)$ 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 $\mathcal{N}$. 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 $S = D_{\mathcal{N}}(u) \cup D_{\mathcal{N}}(y) \in \Sigma_2(\mathcal{N})$. ◀

▶ **Lemma 21.** *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 $\mathcal{N}_2$. 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 20, $D_{\mathcal{N}_1}(u)$ 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 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, the 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 20, this is also a 1-clade or 2-clade of $\mathcal{N}_2$, which contradicts the conditions of the rule. ◀

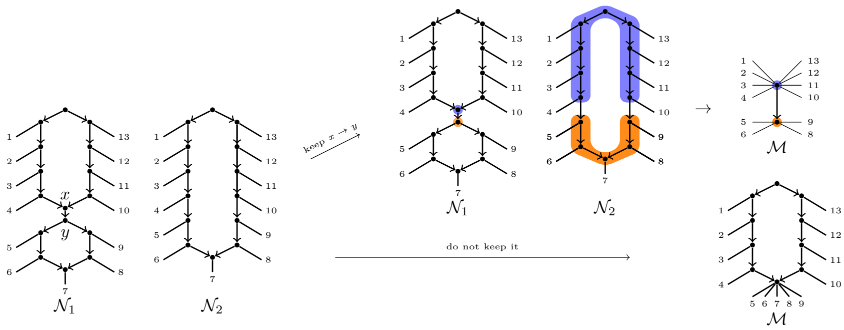

Figure S2 Example of an instance where keeping a common 1-clade is not optimal. Indeed, if the edge incoming to the root of the second cycle in $\mathcal{N}_1$ is kept (corresponding to the 1-clade $\{5, 6, 7, 8, 9\}$), then its two ends belong to two different sets W_u and W_v of an $\mathcal{M}$-witness 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.

D Proofs for Section 5 (A Dynamic Programming Algorithm for Weakly Galled Trees)

▶ **Lemma 22.** *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 $w \in V(\mathcal{N})$. 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 $\mathcal{N}\langle w \rangle = \mathcal{N}_u^1 * \ldots * \mathcal{N}_u^q$. 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. 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. ◀

▶ **Lemma 23.** *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 leafsets 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 both to some 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 $S = D(u)$, 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 states that $\forall i \in [p]$, $\exists j$ 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 $B \setminus A \neq \emptyset$ or $A \setminus B \neq \emptyset$. 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, or rooted at a cycle.

Consider the case where the root r_2 of $\mathcal{N}_2^j$ has a single child v. Let $x \in A \cap B$ and $y \in B \setminus A$. 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 goes through r_1, by 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 $\mathcal{N}_2^j$ is rooted at a cycle C. Let z be the reticulation of C and let x be a leaf that z 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 should 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. ◀