When Two Trees Go to War

Leo van Iersel[b,2,*], Steven Kelk[a,1,**]

[a] *Centrum voor Wiskunde en Informatica (CWI), P.O. Box 94079, 1090 GB Amsterdam, The Netherlands.*

[b] *University of Canterbury, Department of Mathematics and Statistics, Private Bag 4800, Christchurch, New Zealand.*

Abstract

Rooted phylogenetic networks are used to model non-treelike evolutionary histories. Such networks are often constructed by combining trees, clusters, triplets or characters into a single network that in some well-defined sense simultaneously represents them all. We review these four models and investigate how they are related. Motivated by the parsimony principle, one often aims to construct a network that contains as few *reticulations* (non-treelike evolutionary events) as possible. In general, the model chosen influences the minimum number of reticulation events required. However, when one obtains the input data from two binary (i.e. fully resolved) trees, we show

*Corresponding author

**Principal corresponding author, telephone +31 (0)20 5924265, fax +31 (0)20 5924199.

Email addresses: l.j.j.v.iersel@gmail.com (Leo van Iersel), s.m.kelk@cwi.nl (Steven Kelk)

[1]Steven Kelk was funded by a Computational Life Sciences grant of The Netherlands Organisation for Scientic Research (NWO).

[2]Leo van Iersel was funded by the Allan Wilson Centre for Molecular Ecology and Evolution.

that the minimum number of reticulations is independent of the model. The number of reticulations necessary to represent the trees, triplets, clusters (in the softwired sense) and characters (with unrestricted multiple crossover recombination) are all equal. Furthermore, we show that these results also hold when not the number of reticulations but the level of the constructed network is minimised. We use these unification results to settle several computational complexity questions that have been open in the field for some time. We also give explicit examples to show that already for data obtained from three binary trees the models begin to diverge.
Keywords: Reticulation, phylogenetic network, cluster, triplet, character.

1. Introduction

One of the main challenges in phylogenetics is to reconstruct evolutionary histories from biological data of currently living organisms. The traditional and most widely-used model for representing evolutionary histories is the phylogenetic tree. However, recent years have seen more and more interest in the generalisation of phylogenetic trees to phylogenetic networks, which can model non-treelike evolution. These phylogenetic networks contain special nodes, called *reticulations*, in which previously diverged lineages recombine. These nodes represent "reticulate" evolutionary phenomena such as hybridisation, recombination or lateral (horizontal) gene transfer. For a full overview of theory and methods concerning phylogenetic networks, see [1–3].

Motivated by the parsimony principle, a phylogenetic network with fewer reticulations is often preferred over a network with more reticulations, when

both networks represent the available data equally well. Alternatively, one can aim to minimise the "level" of the constructed network, i.e. the number of reticulations per tangled part of the network, see Figure 1. Thus, it is interesting to compute the minimum number of reticulations, or alternatively the minimum level, necessary to represent certain data by a phylogenetic network.

How these minima depend on the chosen model is still very poorly understood. Many algorithms and software packages (see [1–3] and the overview we give in Section 2) are available for many different models, but how these models are related, and whether they are essentially different, often remains undiscussed. This article illuminates the relation between several such models. The special case of an input consisting of two phylogenetic trees has been discussed repeatedly in different contexts [4–10]. We take a closer look at this special case and show that it is indeed very special: three fundamentally different models turn out to be, in this special case, equivalent.

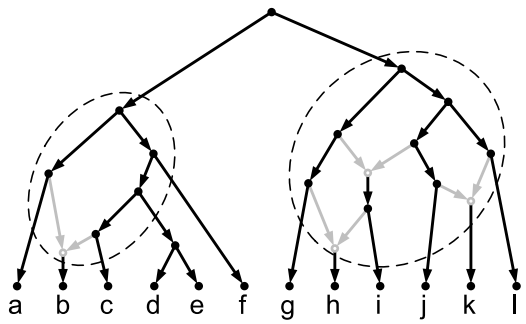

Figure 1: A phylogenetic network with four reticulations (grey, unfilled vertices). This is a level-3 network, because the tangled parts (encircled) contain at most three reticulations each.

We focus on four models for the construction of phylogenetic networks. Probably the most natural one is the "tree-model" which aims at combining several phylogenetic trees into a single phylogenetic network that precisely displays each of the trees; e.g., see [11]. This is especially interesting when certain parts of the genome (e.g. genes) are known to have evolved in a tree-like fashion. One can then generate a phylogenetic tree for each tree-like part of the genome separately, and combine them into a phylogenetic network that represents each of the trees.

Another model is to extract a set of *triplets* (phylogenetic trees with three taxa each) and to combine them into a phylogenetic network that represents each of the triplets; e.g., see [12]. Triplets can be constructed in two ways. Firstly, one can use any of the methods for constructing phylogenetic trees for some or all combinations of three taxa (using a fourth taxon as an outgroup in order to root the triplet). Alternatively, one can first construct one or more phylogenetic trees (on all taxa) and subsequently find the set of triplets that are contained in these trees. The main motivation for the latter approach is that representing all triplets might require fewer reticulations than representing the entire trees. In Section 3.3, we indeed give an explicit example of three trees for which the triplets in the trees can be represented with fewer reticulations than necessary to represent the trees themselves. On the other hand, this section also shows that, for two fully resolved trees, the numbers of reticulations needed to represent the trees or the triplets in the trees are always the same. Moreover, these results also hold when the level rather than the total number of reticulations is minimised.

A third model extracts a set of *clusters* and combines those into a phylogenetic network; e.g., see [8]. Clusters can be obtained from morphological data or from phylogenetic trees. The latter approach has a similar motivation as in the triplet-model. The clusters from the trees might be representable using fewer reticulations than that would be necessary to represent the trees themselves. In addition, the clusters described by a phylogenetic tree are biologically the most interesting features of the tree, because they describe putative monophyletic groups of species (also called clades). In Section 3.2, we show that clusters are in some sense 'between' triplets and trees. The number of reticulations required by the triplets is always less than or equal to the number of reticulations required by the clusters, and this latter number is in turn less than or equal to the number of reticulations required to represent the trees themselves. In Section 3.3, we give examples of sets of three trees for which these inequalities are strict. However, in this section we also show that, for two fully resolved trees, the number of reticulations needed to represent the clusters is always equal to the number of reticulations needed to represent the triplets or trees. We again show that all these results also hold when the level rather than the total number of reticulations is minimised.

The last model we consider in this article constructs phylogenetic networks from *binary characters*. This kind of data consists of a matrix of 0s and 1s and can for example be constructed from DNA, morphological data or phylogenetic trees. Binary characters have been well studied in the field of population genetics [13]. In Section 3.1, we clarify the relation between this model and the cluster model mentioned above, to put our main results

in the correct context.

The structure of the remainder of this article is as follows. The next section describes the mathematical models in detail, gives an overview of known results for each model, and summarises our results. In Section 3 we prove our unification results and in Section 4 we use these results to prove several computational complexity results. We end the article in Section 5 with some concluding remarks.

2. Mathematical Models and Summary of Results

2.1. Phylogenetic Networks

Consider a set of taxa $\mathcal{X}$. A *rooted phylogenetic network* on $\mathcal{X}$ is a directed acyclic graph with exactly one vertex with indegree-zero (the *root*) in which the outdegree-zero nodes (the *leaves*) are bijectively labelled by $\mathcal{X}$. It is common to identify a leaf with the taxon it is labelled by and it is usually assumed that there are no nodes with indegree and outdegree one; we adopt both conventions. Nodes with indegree at least two are called *reticulations*. The edges entering a reticulation are called *reticulation edges*. Nodes that are not reticulations are called *tree nodes*. A phylogenetic network is called *binary* (or fully resolved) if all reticulations have indegree two and outdegree one and all other nodes have outdegree zero or two. In this article we only consider rooted (as opposed to unrooted) phylogenetic networks and for this reason we henceforth omit the prefix "rooted".

As mentioned before, we are interested in minimizing either the number

of reticulation events or the level of the constructed network. The following subtlety has to be taken into account when reticulations with indegree higher than two are considered. When counting such reticulations, indegree-d reticulations are counted $d-1$ times, because such reticulations represent $d-1$ reticulate evolutionary events (of which the order is not specified). Hence, using $\delta^-(v)$ to denote the indegree of a node v, we formally define the *number of reticulations* in a phylogenetic network $N = (V, E)$ as

$$\sum_{v \in V : \delta^-(v) > 0} (\delta^-(v) - 1) = |E| - |V| + 1 \ .$$

Thus, we define the following fundamental problem MinRet. Given some data describing some taxa, find a phylogenetic network that "represents" the given data and contains a minimum number of reticulations over all phylogenetic networks that represent the given data. We consider three specific variants of this problem: MinRetTrees, MinRetTriplets and MinRetClusters, for data consisting of trees, triplets and clusters respectively.

Let us now formally define the level of a phylogenetic network. A *biconnected component* is a maximal subgraph that cannot be disconnected by removing a single node. A biconnected component is *trivial* if it is equal to a single edge and *nontrivial* otherwise. For $k \in \mathbb{N}$, a phylogenetic network is called a *level-k* network if each nontrivial biconnected component contains at most k reticulations. See Figure 1 for an example of a phylogenetic network with four reticulations. The grey, unfilled vertices are reticulations and the grey edges are reticulation-edges. This is a level-3 network, because the nontrivial biconnected components (encircled by dashed lines) contain at most

three reticulations each.

We are now ready to define the following MinLev variant of the fundamental problem. Given some data describing some taxa, find a level-k phylogenetic network that "represents" the given data such that k is as small as possible. There are again three versions: MinLevTrees, MinLevTriplets and MinLevClusters, for data consisting of trees, triplets and clusters respectively.

The following four subsections take a more detailed look at the four possible types of input data: trees, triplets, clusters and binary characters. Throughout the paper we assume a fixed set $\mathcal{X}$ of taxa.

2.2. Trees

A *rooted (binary) phylogenetic tree* on $\mathcal{X}$ is a rooted (binary) phylogenetic network on $\mathcal{X}$ without reticulations. We only consider rooted trees and thus omit the prefix "rooted". A phylogenetic tree T is *displayed* by a phylogenetic network N if T can be obtained from some subtree of N by suppressing nodes with indegree one and outdegree one (i.e. if some subtree of N is a subdivision of T). See Figure 2 for an example.

For a set $\mathcal{T}$ of phylogenetic trees on $\mathcal{X}$, we define:

- $r_t(\mathcal{T})$ as the minimum number of reticulations in any phylogenetic network on $\mathcal{X}$ that displays each tree in $\mathcal{T}$ and
- $\ell_t(\mathcal{T})$ as the minimum k such that there exists a level-k phylogenetic

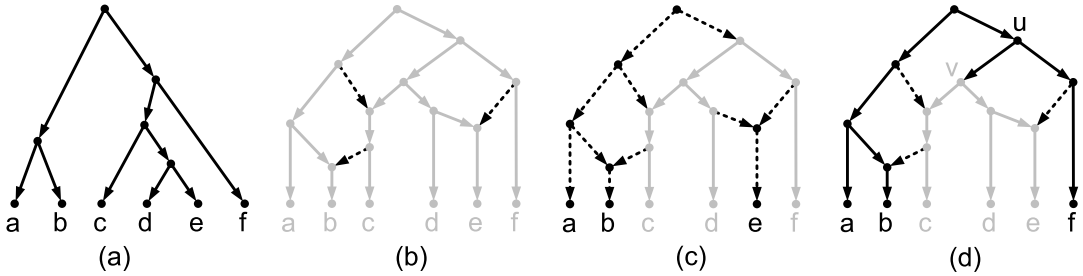

Figure 2: A phylogenetic tree T (a) and a phylogenetic network N (b,c,d); (b) illustrates in grey that N displays T (edges not in the subdivision are dashed); (c) illustrates that N is consistent with the triplet $cd|f$ from T (edges not in the embedding are again dashed); (d) illustrates that N represents cluster $\{c, d, e\}$ from T in the softwired sense (dashed reticulation edges are "switched off").

network on $\mathcal{X}$ that displays each tree in $\mathcal{T}$.

The computation of r_t has received much attention in the literature. For two binary trees on the same taxon set the problem is NP-hard and APX-hard [6] although on the positive side it is fixed-parameter tractable in r_t [4, 5]; [3] offers a good overview of these and related results. These algorithmic insights have been translated into the software HybridNumber [4] and its more advanced successor HybridInterleave [7]. These programs compute r_t exactly for two binary trees on the same taxon set. The program SPRDist [10] solves the same problem (using integer linear programming) and the program PIRN [14] can compute lower and upper bounds on r_t for any number of binary trees on the same taxon set. In [15] a polynomial-time algorithm is described that constructs a level-1 phylogenetic network that displays all trees and has a minimum number of reticulations, if such a network exists.

2.3. Triplets

A *(rooted) triplet* on $\mathcal{X}$ is a rooted binary phylogenetic tree on a size-3 subset of $\mathcal{X}$. As with networks and trees we drop the prefix "rooted", assuming that it is implicit. We use $xy|z$ to denote the triplet with taxa x, y on one side of the root and z on the other side of the root. For triplets, the notion of "represent" can be formalised by the notion of "display" introduced above. However, for triplets "consistent with" is often used instead of "displayed by". A triplet $xy|z$ is *consistent* with a phylogenetic network N (and N is *consistent* with $xy|z$) if $xy|z$ is displayed by N. See Figure 2 for an example. Given a phylogenetic tree T on $\mathcal{X}$, we let $Tr(T)$ denote the set of all triplets on $\mathcal{X}$ that are consistent with T. For a set of phylogenetic trees $\mathcal{T}$, we let $Tr(\mathcal{T})$ denote the set of all triplets that are consistent with some tree in $\mathcal{T}$, i.e. $Tr(\mathcal{T}) = \bigcup_{T \in \mathcal{T}} Tr(T)$.

For a set $\mathcal{R}$ of triplets on $\mathcal{X}$, we define:

- $r_{tr}(\mathcal{R})$ as the minimum number of reticulations in any phylogenetic network on $\mathcal{X}$ that is consistent with each triplet in $\mathcal{R}$ and
- $\ell_{tr}(\mathcal{R})$ as the minimum k such that there exists a level-k phylogenetic network on $\mathcal{X}$ that is consistent with each triplet in $\mathcal{R}$.

Throughout the article we will write $r_{tr}(\mathcal{T})$ and $\ell_{tr}(\mathcal{T})$ as abbreviations for $r_{tr}(Tr(\mathcal{T}))$ and $\ell_{tr}(Tr(\mathcal{T}))$ respectively.

A triplet set $\mathcal{R}$ on $\mathcal{X}$ is said to be *dense* when, for every three distinct taxa $x, y, z \in \mathcal{X}$, at least one of $xy|z, xz|y, yz|x$ is in $\mathcal{R}$ [16]. Given a dense

triplet set, [16, 17] describe a polynomial-time algorithm that constructs a level-1 network displaying all triplets, if such a network exists. The algorithm in [18] can be used to find such a network that also minimizes the number of reticulations, and this is available as the program MARLON [19]. These results have later been extended to level-2 [18, 20] (see also the program LEVEL2 [21]) and more recently to level-k, for all $k \in \mathbb{N}$ [22]. The program SIMPLISTIC [18, 23] can be used to construct (simple) networks of arbitrary level (again, assuming density).

2.4. Clusters

A *cluster* on $\mathcal{X}$ is a proper subset of $\mathcal{X}$. We use $Cl(T)$ to denote the set of clusters of a phylogenetic tree T, i.e. for each edge (u, v) of T, the set $Cl(T)$ contains a cluster consisting of precisely those taxa that are reachable by a directed path from v. For a set $\mathcal{T}$ of phylogenetic trees, we define $Cl(\mathcal{T}) = \bigcup_{T \in \mathcal{T}} Cl(T)$.

Similar to tree- and triplet methods, the general aim of cluster methods is to construct a phylogenetic network that "represents" some set of input clusters. There are two different notions of "representing" for clusters: the "hardwired" and the "softwired" sense. Given a cluster $C \subset \mathcal{X}$ and a phylogenetic network N on $\mathcal{X}$, we say that N *represents* C *in the hardwired sense* if there exists an edge (u, v) in N such that C is the set of taxa reachable from v by a directed path [24].

The definition of "representing" in the "softwired sense" is longer but biologically more relevant. We say that N *represents* C *in the softwired sense*

if there exists an edge (u, v) in N such that C is the set of taxa reachable from v by a directed path, when for each reticulation r exactly one of its incoming edges is "switched on" and all other edges entering r are "switched off" (see Figure 2). As a direct consequence, C is represented by N in the softwired sense if and only if there exists a phylogenetic tree T on $\mathcal{X}$ that is displayed by N and has $C \in Cl(T)$. In this article, we do not consider cluster representation in the hardwired sense and therefore often write "represents" as short for "represents in the softwired sense".

For a set of clusters $\mathcal{C}$ on $\mathcal{X}$, we define:

- $r_c(\mathcal{C})$ as the minimum number of reticulations in any phylogenetic network on $\mathcal{X}$ that represents all clusters in $\mathcal{C}$ in the softwired sense and
- $\ell_c(\mathcal{C})$ as the minimum k such that there exists a level-k phylogenetic network on $\mathcal{X}$ that represents all clusters in $\mathcal{C}$ in the softwired sense.

We write $r_c(\mathcal{T})$ as shorthand for $r_c(Cl(\mathcal{T}))$ and $\ell_c(\mathcal{T})$ as shorthand for $\ell_c(Cl(\mathcal{T}))$.

A network is a *galled network* if it contains no path between two reticulations that is contained in a single biconnected component. In [25] and [8] an algorithm is described for constructing a galled network representing $\mathcal{C}$ in the softwired sense. In [9] the algorithm Cass [26] is presented which aims at constructing a low-level network that represents $\mathcal{C}$. Cass always returns a network representing all input clusters and, when $\ell_c(\mathcal{C}) \leq 2$, it is guaranteed to compute ℓ_c exactly. Alongside the algorithms from [8, 24, 25], Cass is

available as part of the program DENDROSCOPE [27].

2.5. Binary character data

Within the field of population genomics the literature on phylogenetic networks has evolved along a slightly different route to the literature on trees, triplets and clusters. At the level of populations the principle reticulation event is *recombination*, and in this context phylogenetic networks are sometimes called *recombination networks*. To avoid repetition we refer to [28–30] for background and formal definitions; as in those articles we consider exclusively the "infinite sites" model where character data is assumed to be *binary* and where each character mutates at most *once*. We furthermore assume that the root sequence is the all-0 sequence i.e. we are dealing with the "root known" variant of the problem. The input is a binary $n \times m$ matrix M.

The basic definition given in [29] is for the *unrestricted multiple crossover* variant of the recombination network model. Stated informally this means that, at each reticulation, each character can freely "choose" from which of its parents it inherits its value. This is quite different to the *single crossover* variant which has received far more attention in the literature. In the single crossover variant the sequence at a reticulation is forced to obtain a prefix from one of its parents, and a suffix from the other, thus modelling chromosomal crossover. In both variants tree nodes behave the same: each character at a tree node v inherits its value from its parent, unless the character mutated along the edge entering v, in which case it takes the opposite value

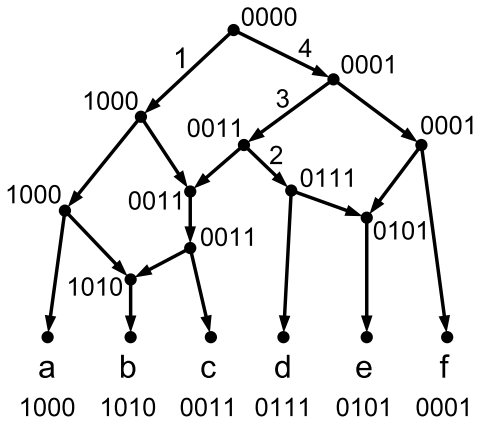

Figure 3: A recombination network that represents the binary character data given at the leaves under the unrestricted multiple crossover model. A label i on an edge indicates that character i mutated along that edge. The network does not represent the character data under the single crossover model, because 1010 can not be obtained by combining a prefix of 1000 with a suffix of 0011 or vice versa.

to its parent. (When the root is the all-0 sequence then this mutation will always be from 0 to 1).

See Figure 3 for an example recombination network that represents given binary character data under the unrestricted multiple crossover variant, but not under the single crossover variant.

For a binary matrix M, we define:

- $r_{sc}(M)$ as the minimum number of reticulations required by a recombination network that represents M, assuming the single crossover variant and an all-0 root, and
- $r_{uc}(M)$ as the minimum number of reticulations required by a recombi-

nation network that represents M, assuming the unrestrained multiple crossover variant and an all-0 root.

Given that the latter is a relaxation of the former, it is immediately clear that for any input M,

$$r_{uc}(M) \leq r_{sc}(M). \tag{1}$$

In [31] it was claimed that it is NP-hard to compute r_{uc}. However, [6] subsequently discovered that the proof in [31] was partially incorrect and modified it to prove that computation of r_{sc} is NP-hard.

There are some definitional subtleties when trying to map between the model of [29] and the other models summarised in this article. Some differences between the models are rather arbitrary and minor and thus easy to overcome, and we do not discuss them here. In this article we restrict ourself to a more fundamental comparison concerning (under an appropriate transformation) the values $r_{sc}(M)$, $r_{uc}(M)$ and $r_c(\mathcal{C})$.

The problem of computing r_{sc} (in defiance of its NP-hardness) has attracted much attention. Articles such as [13, 28–30, 32] give a good overview of the methods in use. Much energy has been invested in computing lower bounds for r_{sc} (e.g. the program HapBound [13]), and some lower bounding techniques also produce valid lower bounds for r_{uc} (e.g. [29]). Programs such as Shrub [13] produce upper bounds on r_{sc}, and Beagle [32] uses integer linear programming to compute r_{sc} exactly (for small instances). The programs HapBound-GC and Shrub-GC compute lower and upper bounds on a value that lies somewhere between r_{sc} and r_{uc} [33]. As in other ar-

eas of the phylogenetic network literature the problem of computing r_{sc} in a topologically constrained space of networks [34] has also been considered.

2.6. Summary of Results

In this article, we study how several methods for constructing phylogenetic networks are related. We begin by clarifying the relationship between phylogenetic networks that represent clusters in the softwired sense and recombination networks that represent binary character data. We explain that the two models are equivalent when unrestricted multiple crossover recombination is considered but fundamentally different when single crossover recombination is used. This clarification is necessary to place the main results from this article in the correct context.

We then turn to the problem of constructing phylogenetic networks from trees, triplets or clusters. In particular, we focus on triplets and clusters obtained from a set of trees on the same set of taxa. We show that the number of reticulations required to display the triplets is always less than or equal to the number of reticulations necessary to represent all clusters, and the latter number is in turn less than or equal to the number of reticulations necessary to display the trees themselves:

$$r_{tr}(\mathcal{T}) \leq r_c(\mathcal{T}) \leq r_t(\mathcal{T}) \ .$$

We give examples for which these inequalities are strict i.e. an example in which the triplets need strictly fewer reticulations than the clusters and

an example in which the clusters need strictly fewer reticulations than the trees.

However, the main result of this article shows that, when one considers a set $\mathcal{T}$ containing two binary trees on the same set of taxa, the numbers of reticulations required to represent the triplets, clusters or the trees themselves are all equal:

$$r_{tr}(\mathcal{T}) = r_c(\mathcal{T}) = r_t(\mathcal{T}) \ .$$

In addition, all the results above also hold for minimizing level. In particular:

$$\ell_{tr}(\mathcal{T}) = \ell_c(\mathcal{T}) = \ell_t(\mathcal{T}) \ .$$

These unification results turn out to have important consequences. We use the equalities above to settle several complexity questions that have been open for some time and to strengthen several existing complexity results. In particular, we show that computation of $\ell_t(\mathcal{T})$, $r_c(\mathcal{T})$, $\ell_c(\mathcal{T})$, $r_{tr}(\mathcal{T})$ and $\ell_{tr}(\mathcal{T})$ are all NP-hard and APX-hard even when $\mathcal{T}$ consists of two binary trees on the same set of taxa. Thus, problems MinRetTriplets, MinRetClusters, MinLevTrees, MinLevTriplets and MinLevClusters are all NP-hard and APX-hard (which was already known for MinRetTrees [6]).

3. Spot the difference

3.1. Clusters and binary character data

Let $\mathcal{C}$ be a set of clusters on $\mathcal{X}$. Let $\mathcal{X} = \{x_1, ..., x_n\}$ and $\mathcal{C} = \{c_1, ..., c_m\}$ i.e. impose an arbitrary ordering on $\mathcal{X}$ and $\mathcal{C}$. The *matrix encoding* of $\mathcal{C}$ is a binary matrix $Mat(\mathcal{C})$ with n rows and m columns. $Mat(\mathcal{C})_{i,j}$ has the value 1 if and only if c_j contains taxon x_i. It is also natural to define the "dual" encoding. Given an $n \times m$ binary matrix M, the *cluster encoding* of M is a cluster set $Clus(M)$ containing a set of m clusters $\{c_1, ..., c_m\}$ on taxon set $\{x_1, ..., x_n\}$ such that c_j contains x_i if and only $M_{i,j}$ has value 1. Clearly both encodings can be produced in polynomial time.

The following result was presented in [35] and is to some extent implicit in [36] (and thus should be attributed to these two groups of authors) although to the best of our knowledge has never been formally written down. It shows that in a very strong sense the construction of phylogenetic networks from clusters, and recombination networks from binary characters under the all-0 root, unrestricted multiple crossover variant, are equivalent.

Observation 1. *Given a cluster set $\mathcal{C}$, any phylogenetic network N that represents $\mathcal{C}$ can be relabelled (after possibly a trivial modification) to obtain a recombination network that represents $Mat(\mathcal{C})$ under the unrestricted multiple crossover variant with all-0 root. Given a binary matrix M, any recombination network that represents M under the unrestricted multiple crossover variant with all-0 root can be relabelled (after possibly a trivial modification) to obtain a phylogenetic network that represents $Clus(M)$.*

PROOF. The core idea is that the edges which represent clusters will become the edges upon which mutations from 0 to 1 will occur, and vice-versa. We will now formalise this.

Consider first a cluster set $\mathcal{C} = \{c_1, ..., c_m\}$ and a phylogenetic network N that represents it. If necessary we first modify N slightly to ensure that every reticulation has outdegree exactly 1. Now, for each cluster $c_j \in \mathcal{C}$ there exists some tree T_j on $\mathcal{X}$ that is displayed by N and which represents c_j. To obtain the recombination network for $Mat(\mathcal{C})$ we relabel as follows: the root of N receives the all-0 sequence and for each c_j $(1 \leq j \leq m)$ we locate the edge e_j in T_j that represents c_j, and fix some subdivision of T_j in N. The edge e_j will thus correspond to a directed path of edges in N; we arbitrarily choose one edge from this path as the edge at which character j mutates from 0 to 1. (We can assume without loss of generality that this is not a reticulation edge). For each node v in N we say that character j has value 1 if and only if v lies in the subdivision of T_j that we fixed and the node v' in T_j to which it corresponds is reachable in T_j from e_j by a directed path. In particular, each character at a reticulation v inherits its value from the node immediately preceding v in the subdivision.

Given an $n \times m$ binary matrix M and a recombination network N that represents it under the unrestricted multiple crossover variant with all-0 root, we first ensure that reticulations in N with outdegree 0 are modified to have outdegree exactly 1. We can thus assume without loss of generality that mutations do not occur on reticulation edges: the mutation can be moved if necessary to the edge leaving the reticulation. Now, we can relabel N

as follows. The leaf labelled with row i of M is mapped to taxon x_i of $\mathcal{X}$. Now, recall that the jth column of M corresponds to cluster $c_j \in Clus(M)$. Consider any such j. At every node v in N it is either (i) unambiguous from which parent of v the value of character j was inherited, or (ii) it is ambiguous, in which case we can arbitrarily choose any such parent, or (iii) character j mutates from a 0 to 1 on the edge feeding into v, in which case choose that edge. This induces a tree which will be a subdivision of some tree T_j on $\mathcal{X}$. Furthermore, T_j represents c_j, and we are done. □

Corollary 1. *Given a cluster set $\mathcal{C}$, $r_c(\mathcal{C}) = r_{uc}(Mat(\mathcal{C}))$. Given a binary matrix M, $r_{uc}(M) = r_c(Clus(M))$.*

It is natural to wonder whether the single crossover variant is genuinely more restrictive than the unrestrained multiple crossover variant. Could it be, for example, that the columns of an input matrix M can always be re-ordered to obtain a matrix M' such that $r_{sc}(M') = r_{uc}(M)$? This is not so, as the following simple example shows. We observe firstly that for a cluster set $\mathcal{C}$ on a set of taxa $\mathcal{X}$, $r_c(\mathcal{C}) \leq |\mathcal{X}| - 1$. This follows because we can use the construction depicted in Figure 4. Let, $n = |\mathcal{X}|$. For any $n \geq 5$, we let $\mathcal{C}_n$ be the set of all clusters that contain exactly $\lfloor n/2 + 1 \rfloor$ elements of $\mathcal{X}$. Let $M = Mat(\mathcal{C}_n)$. It follows by Observation 1 that $r_{uc}(M) = r_c(Clus(M)) = r_c(\mathcal{C}_n) \leq n - 1$.

Clearly M has $k = \binom{n}{\lfloor n/2+1 \rfloor}$ columns and k grows exponentially in n. Let M' be obtained from M by arbitrarily permuting its columns. We say that two clusters $C_1, C_2 \subset \mathcal{X}$ are *compatible* if either $C_1 \cap C_2 = \emptyset$ or $C_1 \subset C_2$

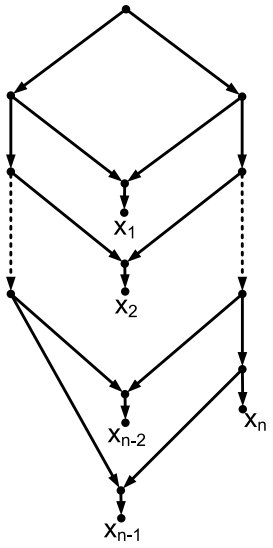

Figure 4: A network that is consistent with all $3\binom{n}{3}$ triplets and represents all $2^n - 1$ clusters on taxon set $\mathcal{X} = \{x_1, ..., x_n\}$.

or $C_2 \subset C_1$ and *incompatible* otherwise. Note that any adjacent pair of columns in M' fails the three-gamete test [29] (with respect to the all-0 root) because two distinct clusters containing $\lfloor n/2 + 1 \rfloor$ elements are necessarily incompatible. Hence, if we partition the columns of M' into $\lfloor k/2 \rfloor$ disjoint pairs of adjacent columns, and apply a composite haplotype bound (i.e. apply the haplotype bound independently to each disjoint pair of columns) [13][37], it follows that $r_{sc}(M') \geq \lfloor k/2 \rfloor$. This lower bound grows exponentially in n, independently of the exact column permutation applied, while the upper bound on $r_{uc}(M)$ grows only linearly. For $n \geq 5$, the gap between these bounds is already greater than zero.

We remark in passing that the "root unknown" version of the unrestrained multiple crossover variant (let us denote this by r^*_{uc}) has an interesting in-

terpretation when given $Mat(\mathcal{C})$ as input. In the "root unknown" version characters are allowed to start with value 1 at the root and mutate at most once to 0 (as opposed to always starting with value 0 at the root and mutating at most once to 1). It follows then that $r^*_{uc}(Mat(\mathcal{C}))$ is the minimum number of reticulations ranging over all networks that, for each cluster $c \in \mathcal{C}$, represents c *or the complementary cluster* $|\mathcal{X}|\backslash c$. It is easy to see that $r^*_{uc}(Mat(\mathcal{C}))$ can be significantly smaller than $r_{uc}(Mat(\mathcal{C}))$. For example, consider the set $\mathcal{C}$ of all size-2 clusters on a size-3 taxon set $\mathcal{X}$. These clusters are mutually incompatible, so $r_{uc}(Mat(\mathcal{C})) \geq 1$. However, the complement of each cluster is a singleton cluster, so (by choosing the all-1 root) $r^*_{uc}(Mat(\mathcal{C})) = 0$.

3.2. Clusters and triplets coming from trees

Let us take a closer look at sets of triplets or clusters that are obtained from a set $\mathcal{T}$ of (not necessarily binary) phylogenetic trees on the same set of taxa. We will show that any phylogenetic network that represents $Cl(\mathcal{T})$ is consistent with $Tr(\mathcal{T})$. It follows that representing all triplets requires at most as many reticulations as representing all clusters. Moreover, quite obviously, representing all clusters requires at most as many reticulations as representing the trees themselves. Thus,

$$r_{tr}(\mathcal{T}) \leq r_c(\mathcal{T}) \leq r_t(\mathcal{T}) \ . \tag{2}$$

Furthermore, this is true not only with respect to minimizing the number of reticulations, but with respect to minimizing any property of the networks,

e.g. level:

$$\ell_{tr}(\mathcal{T}) \leq \ell_c(\mathcal{T}) \leq \ell_t(\mathcal{T}) \ . \tag{3}$$

We will show that each of the inequalities in (2) and (3) is strict for some set of trees $\mathcal{T}$.

First, in order to prove (2) and (3), we show an important relation between $Tr(\mathcal{T})$ and $Cl(\mathcal{T})$.

Lemma 1. *For any three taxa $x, y, z \in \mathcal{X}$ holds that $xy|z \in Tr(T)$ if and only if there exists a cluster $C \in Cl(T)$ with $x, y \in C$ and $z \notin C$.*

Proof. First suppose that there is a cluster $C \in Cl(T)$ such that $x, y \in C$ and $z \notin C$. Then the triplet $xy|z$ is consistent with T and hence $xy|z \in Tr(T)$.

Now suppose that $xy|z \in Tr(T)$. Then the triplet $xy|z$ is displayed by T and hence there is a subtree T' of T such that $xy|z$ can be obtained from T' by suppressing nodes with indegree one and outdegree one. This subtree T' contains exactly one node with indegree one and outdegree two. Let C be the set of taxa reachable from this node. Then, $x, y \in C$, $z \notin C$ and $C \in Cl(T)$. □

It follows that, for any set $\mathcal{T}$ of trees on the same set $\mathcal{X}$ of taxa, $Cl(\mathcal{T})$ uniquely determines $Tr(\mathcal{T})$.

We will now prove the following proposition, from which correctness of (2) and (3) follows.

Proposition 1. *For any set $\mathcal{T}$ of trees on the same set $\mathcal{X}$ of taxa, any phylogenetic network on $\mathcal{X}$ representing $Cl(\mathcal{T})$ is consistent with $Tr(\mathcal{T})$.*

PROOF. Let N be a phylogenetic network on $\mathcal{X}$ representing $Cl(\mathcal{T})$. Consider a triplet $xy|z \in Tr(\mathcal{T})$. By Lemma 1, there is a cluster $C \in Cl(T)$ (for some $T \in \mathcal{T}$) with $x, y \in C$ and $z \notin C$. Cluster C is represented by N (in the softwired sense) and hence there exists a phylogenetic tree T_C on $\mathcal{X}$ that is displayed by N and has $C \in Cl(T_C)$. Because $x, y \in C$ and $z \notin C$, it follows that $xy|z$ is displayed by T_C. Since T_C is displayed by N, it follows that $xy|z$ is displayed by N. Hence, N is consistent with $xy|z$. □

Before proceeding further, the following two lemmas will be of use throughout the rest of the article.

Lemma 2. *Let N be a phylogenetic network on $\mathcal{X}$. Then we can transform N into a binary phylogenetic network N' such that N' has the same number of reticulations and the same level as N and any binary tree displayed by N is also displayed by N'.*

PROOF. Each reticulation v with outdegree 0, which is necessarily labelled by some taxon $x \in \mathcal{X}$, is transformed into a reticulation with outdegree 1 by introducing a new node v', adding an edge (v, v') and moving label x to

node v'. Next we deal with nodes v that have both indegree and outdegree greater than 1. Here we replace the node v by an edge (v_1, v_2) such that the edges incoming to v now enter v_1, and the edges outgoing from v now exit from v_2. Subsequently nodes with indegree at most 1, and outdegree $d \geq 3$, can be replaced by a chain of $(d-1)$ nodes of indegree at most 1 and outdegree 2. Nodes with indegree $d \geq 3$ and outdegree 1 can be replaced by a chain of $(d-1)$ nodes of indegree 2 and outdegree 1. This completes the transformation of N into N'. Note that this transformation, which clearly preserves the reticulation number of N, also preserves the level of N because it does not change the number of reticulations in any nontrivial biconnected component.

The critical observation is that if a binary tree T is displayed by N then there is a subdivision of T in N which is also binary. This means that for each node v in N the subdivision uses at most two outgoing edges of v and at most one incoming edge of v. Hence the subdivision can easily be extended to become a subdivision within N'. □

Lemma 3. *Let N be a phylogenetic network on $\mathcal{X}$ and $\mathcal{T}$ a set of binary trees on $\mathcal{X}$. Then there exists a binary phylogenetic network N' on $\mathcal{X}$ such that (a) N' has the same reticulation number and level as N, (b) if N displays all trees in $\mathcal{T}$ then so too does N', (c) if N is consistent with $Tr(\mathcal{T})$ then so too is N' and (d) if N represents $Cl(\mathcal{T})$ then so too does N'.*

Proof. (a) and (b) are immediate from Lemma 2. For (c) note that for each triplet $t \in Tr(\mathcal{T})$ there is some subdivision of t in N. A triplet t is

binary, and thus so too is any subdivision of t, so we can apply the same argument as used in Lemma 2. For (d), note that for each cluster $c \in Cl(\mathcal{T})$ there is some tree T on $\mathcal{X}$ which is displayed by N and which represents c. T is perhaps not binary, and thus a subdivision of it in N is perhaps also not binary, so after the transformation described in Lemma 2 this subdivision will have become the subdivision of some binary tree T'. However, T' is a refinement of T i.e. $Cl(T) \subseteq Cl(T')$ so c is also represented by N'. □

We will now show that each of the inequalities in (2) and (3) is strict for some set of trees. To do so for the first inequality in each formula, consider the set $\mathcal{T}$ of three trees, and the network N, shown in Figure 5. It is easy to check that N is consistent with all the triplets in $Tr(\mathcal{T})$. However, any network that represents $Cl(\mathcal{T})$ requires at least 3 reticulations, and will be level-3 or higher, as can be verified by a straightforward (but technical) case analysis or by using the program Cass [26]. Specifically: if a level-1 or level-2 network existed that represented $Cl(\mathcal{T})$ then Cass would find it [9], and it does not.

Figure 6 shows a set $\mathcal{T}$ of trees for which the second inequality in (2) and (3) is strict. A level-1 network with one reticulation is shown that represents all clusters from the three trees. However, a network with k reticulations can display at most 2^k distinct trees, so any network that displays all three trees will require at least two reticulations. It will also have level at least 2, because a level-1 network (which we may without loss of generality assume to be binary) displaying all three trees would have two nontrivial biconnected components, and thus all three trees would have a common non-singleton

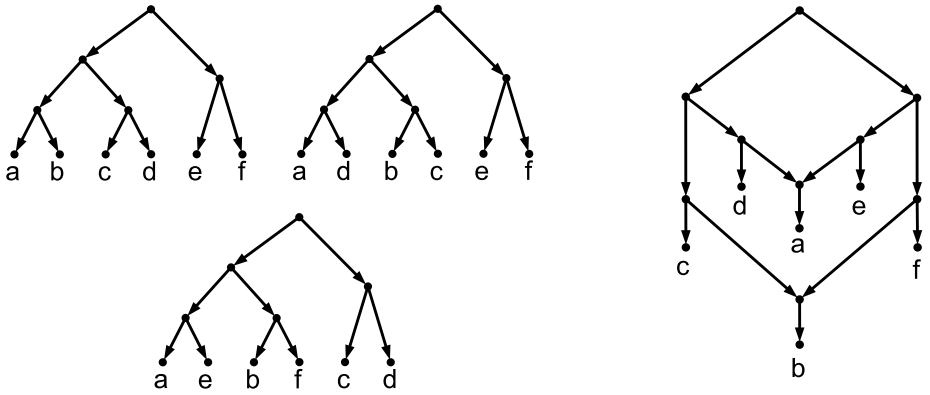

Figure 5: The triplets obtained from the three threes on the left are consistent with the level-2 network on the right containing two reticulations. However, any network representing all the clusters from these trees will have at least three reticulations and be level-3 or higher.

cluster, but this is not so.

Although we do not present a proof, empirical experiments furthermore suggest that it is possible to "boost" the example given in Figure 6 to create sets of three binary trees $\mathcal{T}$ such that the gap between $r_t(\mathcal{T})$ and $r_c(\mathcal{T})$ can be made arbitrarily large [38].

3.3. Clusters and triplets coming from two binary trees

This section presents the main results of this paper. We will show that the number of reticulations necessary to represent the clusters from two binary trees on the same taxa is equal to the number of reticulations necessary to represent the trees themselves. In addition, we will show that also the number of reticulations necessary to represent all triplets from the two trees is equal to the number of reticulations necessary to represent the trees them-

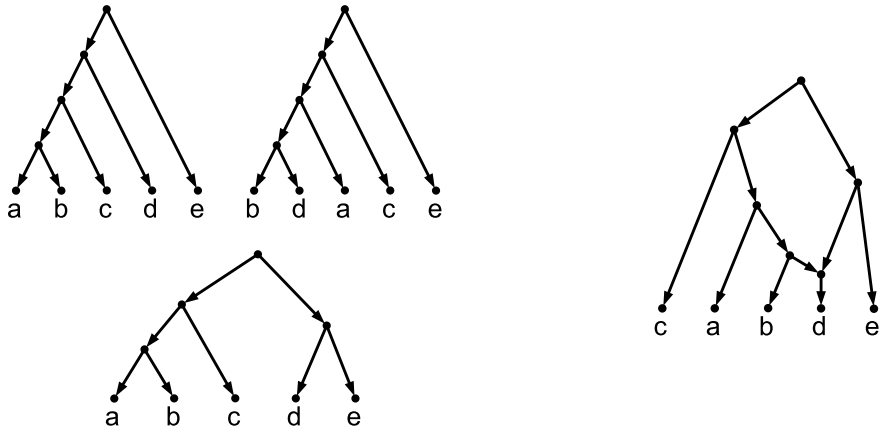

Figure 6: The level-1 network on the right with a single reticulation represents the union of the clusters (and triplets) obtained from the three trees on the left. However, any network that displays all three trees will have at least two reticulations and have level at least two.

selves. Moreover, we will show that the same is true when not the number of reticulations but the level of the networks is minimized. This means that for data coming from two binary trees on the same set of taxa, the tree-, cluster- and triplet problems all coincide.

Let $\mathcal{T}$ be a set containing two binary phylogenetic trees on the same set of taxa. Recall that $Cl(\mathcal{T})$ is the set of all clusters from both trees in $\mathcal{T}$ and $Tr(\mathcal{T})$ is the set of all triplets from both trees. We start by showing that the minimum number of reticulations in a network consistent with $Tr(\mathcal{T})$ is equal to the minimum number of reticulations in a network displaying both trees in $\mathcal{T}$. The fact that also the number of reticulations necessary to represent $Cl(\mathcal{T})$ is the same will be a corollary. After this corollary we will show that the results also hold for level-minimization.

First, however, some context is necessary. As mentioned earlier, [6] fixed

the partially correct result of [31] to prove that computation of r_{sc} is NP-hard. The correct part of the proof in [31], Claim 2, essentially showed that, for a set $\mathcal{T} = \{T_1, T_2\}$ of two binary trees on a set $\mathcal{X}$ of taxa, $r_t(\mathcal{T}) \leq r_{uc}(M^*)$ where M^* is the concatenation of $Mat(Clus(T_1))$ and $Mat(Clus(T_2))$ into a single matrix containing $4(n-1)$ columns (i.e. characters) and $|\mathcal{X}|$ rows. By (1) they thus also proved that that $r_t(\mathcal{T}) \leq r_{sc}(M^*)$ and this fact is used in [6][3]. Now, observe that $Clus(M^*)$ is equal to $Cl(\mathcal{T})$. Hence, by Observation 1, $r_t(\mathcal{T}) \leq r_{uc}(M^*) = r_c(\mathcal{T})$. It is clear that $r_c(\mathcal{T}) \leq r_t(\mathcal{T})$ and hence $r_t(\mathcal{T}) = r_c(\mathcal{T})$. In this sense the equivalence of $r_t(\mathcal{T})$ and $r_c(\mathcal{T})$ for pairs of binary trees was already implicitly present in the literature. However, given (a) the lack of clarity in the proof of [31], (b) the fact that Observation 1 has only been implicitly present in the literature up until now and (c) the desire to produce a unification result which also includes triplets, we have decided that it is useful to directly and explicitly prove this two-tree result and to explore its consequences.

Theorem 1. *If $\mathcal{T} = \{T_1, T_2\}$ consists of two binary phylogenetic trees on the same set of taxa, $r_{tr}(\mathcal{T}) = r_t(\mathcal{T})$.*

Proof. To increase the clarity of the proof we write $r_t(T_1, T_2)$ as shorthand for $r_t(\{T_1, T_2\})$ and $r_{tr}(T_1, T_2)$ as shorthand for $r_{tr}(\{T_1, T_2\})$.

[3]The specific column ordering in M^* - first the clusters from T_1 in arbitrary order, and then the clusters from T_2 in arbitrary order - is important for establishing that $r_t(\mathcal{T}) \leq r_{sc}(M^*)$. In particular, it is easy to construct instances $\{T_1, T_2\}$ such that a bad permutation of the columns of M^* causes $r_{sc}(M^*)$ to be arbitrarily larger than $r_t(\mathcal{T})$.

Clearly, $r_t(T_1, T_2) \geq r_{tr}(T_1, T_2)$, since any phylogenetic network displaying T_1 and T_2 is consistent with all triplets from T_1 and T_2. It remains to show $r_t(T_1, T_2) \leq r_{tr}(T_1, T_2)$.

Suppose this is not true. Let n be the number of leaves in a smallest counter example, i.e. n is the smallest number such that there exist two binary phylogenetic trees T_1 and T_2 on a set of taxa $\mathcal{X}$ with $|\mathcal{X}| = n$ such that $r_t(T_1, T_2) > r_{tr}(T_1, T_2)$. Clearly $n \geq 3$. Let N_t be a phylogenetic network on $\mathcal{X}$ with $r_t(T_1, T_2)$ reticulations that displays T_1 and T_2 and let N_{tr} be a phylogenetic network on $\mathcal{X}$ with $r_{tr}(T_1, T_2)$ reticulations that is consistent with all triplets in T_1 and T_2.

We may assume by Lemma 3 that N_{tr} and N_t are binary. We define a *reticulation leaf* as a leaf whose parent is a reticulation and a *cherry* as two leaves with a common parent.

We first prove that any binary phylogenetic network contains either a reticulation leaf or a cherry. Suppose that this is not true and let N be a smallest counter example, i.e. N has no reticulation leaves and no cherries and has a minimum number of leaves over all such networks. Take any leaf x of N and let p be its parent. It cannot be a reticulation, so p is either a node with indegree one and outdegree two, or the root. In both cases, we delete x and contract the remaining edge leaving p, giving a smaller counter example. We conclude that any binary phylogenetic network contains either a reticulation leaf or a cherry. Hence, this is also true for N_{tr}.

First suppose that N_{tr} contains a cherry. Let this cherry consist of

leaves a, b and their common parent v. Then $\{a, b\}$ is a cluster of T_1 and of T_2 i.e. they both contain an edge whose set of leaf descendants is exactly $\{a, b\}$. If this was not so, then at least one of T_1 and T_2 would be consistent with a triplet $ac|b$ or $bc|a$ for some $c \notin \{a, b\}$ and such a triplet is not consistent with N_{tr}. It follows that each of T_1 and T_2 contains a cherry with leaves a, b. Let T_1' and T_2' be the trees obtained from T_1, T_2 respectively by deleting leaves a and b and labeling their common parent by a new label ab. Now, Theorem 1 of Baroni et al. [39] states that, given a phylogenetic tree T and a cluster $C \in Cl(T)$, let $T|C$ denote the subtree of T on taxon set C and let $T^{C \to c}$ denote the phylogenetic tree obtained from T by replacing the subtree on C by a new leaf c. Then, $r_t(T_1, T_2) = r_t(T_1|C, T_2|C) + r_t(T_1^{C \to c}, T_2^{C \to c})$ whenever $C \in Cl(T_1) \cap Cl(T_2)$. Hence, if we take $C = \{a, b\}$ we have that $r_t(T_1', T_2') = r_t(T_1, T_2)$, because in this case $r_t(T_1|C, T_2|C) = 0$.

Furthermore, $r_{tr}(T_1', T_2') \leq r_{tr}(T_1, T_2)$ because deleting a and b from N_{tr} and labelling v by ab leads to a phylogenetic network with $r_{tr}(T_1, T_2)$ reticulations that is consistent with all triplets in T_1' and T_2'. We conclude that

$$r_t(T_1', T_2') = r_t(T_1, T_2) > r_{tr}(T_1, T_2) \geq r_{tr}(T_1', T_2') \ .$$

Hence, we have constructed a smaller counter example, which shows a contradiction.

Now suppose that N_{tr} contains a reticulation leaf. Let x be such a leaf and r its parent. Let $N_{tr} \backslash_x$ be the result of removing x and r from N_{tr}. Let $N_t \backslash_x$ be the result of removing x from N_t and removing the former parent of x as well if it is a reticulation. Let $T_1 \backslash_x$ and $T_2 \backslash_x$ be the trees obtained from T_1 and T_2 respectively by removing x and contracting the

remaining edge leaving the former parent of x. That is, do the following for $i \in \{1, 2\}$. Let p_i be the former parent of x. If p_i is not the root, there is one edge (u_i^x, p_i) entering p_i and one edge (p_i, v_i^x) leaving p_i. Remove p_i and replace the edges (u_i^x, p_i),(p_i, v_i^x) by a single edge (u_i^x, v_i^x). We will use the edges (u_i^x, v_i^x) later on. If p_i is the root, we remove x and p_i and leave (u_i^x, v_i^x) undefined.

First observe that $N_{tr}\backslash_x$ is consistent with all triplets of $T_1\backslash_x$ and $T_2\backslash_x$. Moreover, since $N_{tr}\backslash_x$ contains one reticulation fewer than N_{tr},

$$r_{tr}(T_1\backslash_x, T_2\backslash_x) < r_{tr}(T_1, T_2) < r_t(T_1, T_2) \tag{4}$$

and hence

$$r_{tr}(T_1\backslash_x, T_2\backslash_x) \leq r_t(T_1, T_2) - 2 \ .$$

Now observe that $N_t\backslash_x$ displays $T_1\backslash_x$ and $T_2\backslash_x$. We will show that

$$r_t(T_1\backslash_x, T_2\backslash_x) \geq r_t(T_1, T_2) - 1 \ . \tag{5}$$

Together, (4) and (5) imply that

$$r_{tr}(T_1\backslash_x, T_2\backslash_x) \leq r_t(T_1, T_2) - 2 \leq r_t(T_1\backslash_x, T_2\backslash_x) - 1$$

and hence that we have obtained a smaller counter example, which is a contradiction.

It remains to prove (5). Let N' be a phylogenetic network on $\mathcal{X} \setminus \{x\}$ with $r_t(T_1\backslash_x, T_2\backslash_x)$ reticulations that displays $T_1\backslash_x$ and $T_2\backslash_x$. Since $T_1\backslash_x$ is displayed by N', there exists a subgraph E_1 of N' that is a subdivision of $T_1\backslash_x$ (an embedding of $T_1\backslash_x$ into N'). Similarly, let E_2 be a subgraph

of N' that is a subdivision of $T_2\backslash_x$. We will now use the edges (u_1^x, v_1^x) and (u_2^x, v_2^x) that we introduced when defining $T_1\backslash_x$ and $T_2\backslash_x$. For $i \in \{1, 2\}$, if the edge (u_i^x, v_i^x) has been defined, we define the edge e_i as follows. The edge (u_i^x, v_i^x) corresponds to a directed path in E_i. Let e_i be any edge of this path. Notice that e_i is an edge of N'.

Let N^+ be the network obtained by subdividing e_1 and e_2 and making x a reticulation leaf below the new nodes. To be precise, for $i \in \{1, 2\}$, if $e_i = (u_i, v_i)$ has been defined, replace e_i by $(u_i, n_i), (n_i, v_i)$ with n_i a new node. If (u_i, v_i) has not been defined, add a new root n_i and an edge from n_i to the old root. Finally, add a leaf labelled x, a new reticulation r and edges $(n_1, r), (n_2, r)$ and (r, x).

Observe that N^+ displays T_1 and T_2, because we can simply extend each of the embeddings E_1 and E_2 by the new edges leading to the leaf x. Moreover, N^+ contains exactly one reticulation more than N'. Thus, $r_t(T_1, T_2) \leq r_t(T_1\backslash_x, T_2\backslash_x) + 1$, which remained to be shown. □

Corollary 2. *If $\mathcal{T}$ consists of two binary phylogenetic trees on the same set of taxa,*

$$r_{tr}(\mathcal{T}) = r_c(\mathcal{T}) = r_t(\mathcal{T}) \ .$$

Proof. Follows from combining Theorem 1 with (2). □

Given this result it is natural to ask whether every network that represents all the clusters (or triplets) from two binary trees T_1 and T_2 on the same taxon

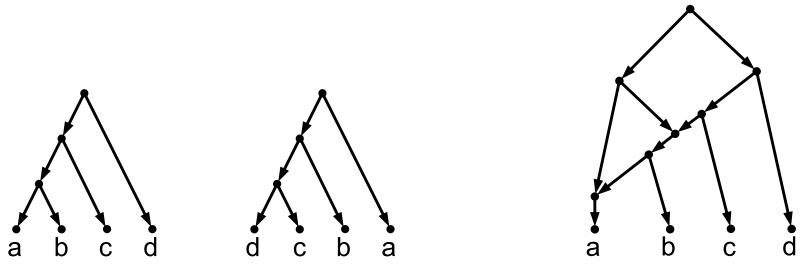

Figure 7: The network on the right represents the union of the clusters (and triplets) obtained from the two trees on the left, but it does not display both trees.

set, and having a minimum number of reticulations, also displays T_1 and T_2. This is not so. Consider the two trees in Figure 7. It is easy to check that two reticulations are necessary and sufficient to display both these trees. The network in this figure contains two reticulations and represents the union of the clusters (and triplets) from both trees, but it does not display both trees.

We note that Theorem 1 and Corollary 2 do not hold for sets of three or more trees, as demonstrated in Section 3.2 by Figure 6. In addition, they also do not hold for two possibly non-binary trees, as demonstrated by Figure 8[4].

We say that an edge of a network N is a *cut-edge* if its removal disconnects N. A cut-edge (u, v) is *trivial* if v is a leaf. N is said to be *simple* if it does not contain any nontrivial cut-edges.

Theorem 2. *If $\mathcal{T}$ consists of two binary phylogenetic trees on the same set*

[4]In some articles a non-binary tree is defined to be displayed by a network if some *binary refinement* of the tree is displayed by it [40]. The definition of r_t is then adjusted accordingly. We defer this issue to a future publication.

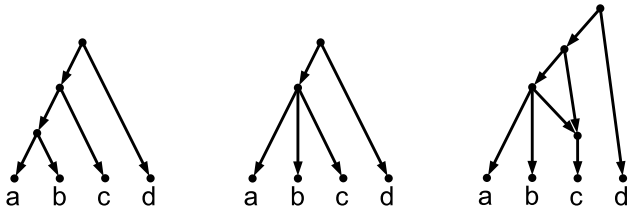

Figure 8: The network on the right displays the two trees on the left: at least one reticulation is necessary. However, the tree on the left is sufficient to represent the union of the clusters (or triplets) obtained from both trees.

of taxa,

$$\ell_{tr}(\mathcal{T}) = \ell_c(\mathcal{T}) = \ell_t(\mathcal{T}) \ .$$

PROOF. By (3), it suffices to show $\ell_t(\mathcal{T}) \leq \ell_{tr}(\mathcal{T})$. We do so by induction on $|\mathcal{X}|$. The base case for $|\mathcal{X}| \leq 2$ is clear. Now consider a set of two binary trees $\mathcal{T}$ on $\mathcal{X}$ with $|\mathcal{X}| = n$. Let N_t be a network that displays both trees in $\mathcal{T}$ and has optimal level $\ell_t(\mathcal{T})$. Similarly, let N_{tr} be a network consistent with $Tr(\mathcal{T})$ that has optimal level $\ell_{tr}(\mathcal{T})$. By Lemma 3 we may assume that N_t and N_{tr} are both binary. We distinguish three cases.

First suppose that neither N_t nor N_{tr} contains nontrivial cut-edges, i.e. that N_t is a simple level-$\ell_t(\mathcal{T})$ network and N_{tr} is a simple level-$\ell_{tr}(\mathcal{T})$ network. In that case, the number of reticulations in N_t is equal to $\ell_t(\mathcal{T})$ (because N_t only contains a single nontrivial biconnected component). So, $r_t(\mathcal{T}) \leq \ell_t(\mathcal{T})$. At the same time, $r_t(\mathcal{T}) \geq \ell_t(\mathcal{T})$, since the number of reticulations in any network is at least equal to its level. Thus, $r_t(\mathcal{T}) = \ell_t(\mathcal{T})$. Similarly, $r_{tr}(\mathcal{T}) = \ell_{tr}(\mathcal{T})$. Moreover, by Theorem 1, $r_{tr}(\mathcal{T}) = r_t(\mathcal{T})$ and we

can conclude that $\ell_{tr}(\mathcal{T}) = r_{tr}(\mathcal{T}) = r_t(\mathcal{T}) = \ell_t(\mathcal{T})$.

Now suppose that N_t contains at least one nontrivial cut-edge and let e be such an edge. Let C be the set of taxa reachable from e by a directed path. Let $\mathcal{T}|C$ be the set of trees obtained by restricting each of the trees in $\mathcal{T}$ to the taxa in C and let $\mathcal{T}^{C\to c}$ denote the set of trees obtained by collapsing, in each tree in $\mathcal{T}$, the subtree on C to a single leaf labelled c. We claim that

$$\begin{aligned}\ell_t(\mathcal{T}) &\leq \max\{\ell_t(\mathcal{T}|C), \ell_t(\mathcal{T}^{C\to c})\}\\ &= \max\{\ell_{tr}(\mathcal{T}|C), \ell_{tr}(\mathcal{T}^{C\to c})\}\\ &\leq \ell_{tr}(\mathcal{T})\ .\end{aligned}$$

To see that $\ell_t(\mathcal{T}) \leq \max\{\ell_t(\mathcal{T}|C), \ell_t(\mathcal{T}^{C\to c})\}$, notice that any network displaying $\mathcal{T}^{C\to c}$ can be combined with any network displaying $\mathcal{T}|C$ in order to obtain a network displaying $\mathcal{T}$. This can be done by replacing the leaf c of the network displaying $\mathcal{T}^{C\to c}$ by the network displaying $\mathcal{T}|C$. The network obtained in this way displays $\mathcal{T}$ and its level is equal to the maximum of the levels of the networks displaying $\mathcal{T}^{C\to c}$ and $\mathcal{T}|C$. So, $\ell_t(\mathcal{T}) \leq \max\{\ell_t(\mathcal{T}|C), \ell_t(\mathcal{T}^{C\to c})\}$. Then we use that $\ell_t(\mathcal{T}|C) = \ell_{tr}(\mathcal{T}|C)$ and $\ell_t(\mathcal{T}^{C\to c}) = \ell_{tr}(\mathcal{T}^{C\to c})$, which itself follows by combining the induction hypothesis with the fact that $\ell_t(\mathcal{T}|C) \geq \ell_{tr}(\mathcal{T}|C)$ and $\ell_t(\mathcal{T}^{C\to c}) \geq \ell_{tr}(\mathcal{T}^{C\to c})$. To prove the last inequality, observe that $\ell_{tr}(\mathcal{T}|C) \leq \ell_{tr}(\mathcal{T})$ because removing leaves can not increase the level. In addition, $\ell_{tr}(\mathcal{T}^{C\to c}) \leq \ell_{tr}(\mathcal{T})$ because $\mathcal{T}^{C\to c}$ can be constructed by removing all leaves in C except for one, which is relabeled c, and removing or relabeling leaves can not increase the level.

The final case is that N_{tr} contains a nontrivial cut-edge e. Let C be the set of taxa that can be reached from e by a directed path in N_{tr}. Clearly, for $x, y \in C$ and $z \notin C$, $xy|z \in Tr(\mathcal{T})$. Thus, C is a cluster of each of the trees of $\mathcal{T}$. Therefore, we can argue in the same way as in the previous case that $\ell_t(\mathcal{T}) \leq \ell_{tr}(\mathcal{T})$. □

4. Complexity Consequences

Theorem 1 and Corollary 2 allow us to elegantly settle several complexity questions in the phylogenetic network literature that have been open for some time, and to significantly strengthen some already existing hardness results.

Corollary 3. *Computing $r_c(\mathcal{T})$ and computing $r_{tr}(\mathcal{T})$ are both NP-hard and APX-hard, even for sets $\mathcal{T}$ consisting of two binary trees on the same set of taxa.*

Proof. Follows from Corollary 2 and the fact that computing $r_t(\mathcal{T})$, for sets $\mathcal{T}$ consisting of two binary trees on the same set of taxa, is NP-hard and APX-hard [6]. □

It follows directly that the following two problems are NP-hard and APX-hard.

MinRetClusters

Instance: A set $\mathcal{X}$ of taxa and a set $\mathcal{C}$ of clusters on $\mathcal{X}$.

Objective: Construct a phylogenetic network on $\mathcal{X}$ that represents each cluster in $\mathcal{C}$ and has a minimum number of reticulations over all such networks.

MinRetTriplets

Instance: A set $\mathcal{X}$ of taxa and a set $\mathcal{R}$ of triplets on $\mathcal{X}$.

Objective: Construct a phylogenetic network on $\mathcal{X}$ that is consistent with each triplet in $\mathcal{C}$ and has a minimum number of reticulations over all such networks.

Moreover, the latter problem is even NP-hard and APX-hard for dense sets of triplets. This strengthens a result by Jansson et al. [16], who showed that MinRetTriplets and MinLevTriplets are NP-hard, by constructing a non-dense set of triplets such that positive instances of the NP-complete problem Set Splitting corresponded to a level-1 network with exactly one reticulation. Corollary 3 extends this result by showing that MinRetTriplets is even NP-hard for dense sets of triplets and that it is hard to approximate (APX-hard).

We now turn our attention to the problems that minimize level.

Theorem 3. *Computing $\ell_t(\mathcal{T})$ is NP-hard and APX-hard, even for sets $\mathcal{T}$ consisting of two binary trees on the same set of taxa.*

Proof. We again reduce from the problem of computing $r_t(\mathcal{T})$, for sets $\mathcal{T}$

consisting of two binary trees on the same set of taxa. We first reduce this problem to the restriction to pairs of trees T_1, T_2 that do not have a common non-singleton cluster. Call this restricted problem ResMinRetTrees.

Consider a set $\mathcal{T}$ consisting of two binary phylogenetic trees T_1, T_2 on a set $\mathcal{X}$ of taxa. Recall Theorem 1 of Baroni et al. [39] and the application of it described in the proof of Theorem 1 in this article. To summarise, $r_t(T_1, T_2) = r_t(T_1|C, T_2|C) + r_t(T_1^{C\to c}, T_2^{C\to c})$ whenever $C \in Cl(T_1) \cap Cl(T_2)$. Thus, repeatedly applying the Baroni theorem, we obtain in polynomial time a collection of at most polynomially-many instances of ResMinRetTrees such that the minimum reticulation number of the original instance is equal to the sum of the minimum reticulation numbers of the obtained instances of ResMinRetTrees. Thus, we can solve the original instance by solving each instance of ResMinRetTrees. This completes the reduction.

We continue by reducing ResMinRetTrees to the problem of computing $\ell_t(\mathcal{T})$. Consider an instance $(\mathcal{X}, T_1, T_2)$ of ResMinRetTrees. Let $\mathcal{T} = \{T_1, T_2\}$. We will prove that $\ell_t(\mathcal{T}) = r_t(\mathcal{T})$ and this will complete the reduction. Clearly $\ell_t(\mathcal{T}) \leq r_t(\mathcal{T})$. Suppose then for the sake of contradiction that $\ell_t(\mathcal{T}) < r_t(\mathcal{T})$. If that is the case, then any level-$\ell_t(\mathcal{T})$ network that displays T_1 and T_2 contains at least two nontrivial biconnected components. By Lemma 3, there exists a binary such phylogenetic network N. Since this network contains at least two nontrivial biconnected components, it contains a cut-edge $e = (u, v)$ such that at least two taxa are reachable from v (by a directed path) and at least one taxon is not. Define cluster E to contain all taxa that are reachable from v in N. Thus, $|E| \geq 2$. T_1 and T_2 are both

displayed by N so, for $i \in \{1, 2\}$, there is a subdivision of T_i in N. Fix any such subdivision. So, each edge of T_i maps to a directed path of one or more edges in N. Both subdivisions must pass through (u, v) and it thus follows that E is a non-singleton cluster of both T_1 and T_2, giving us a contradiction. This completes the NP-hardness proof.

To see that computing $\ell_t(\mathcal{T})$ is not only NP-hard but also APX-hard, observe that ResMinRetTrees is APX-hard because (as shown above) $r_t(\mathcal{T})$ can be computed by simply adding up the optima of polynomially-many instances of ResMinRetTrees. This additivity means that an ϵ-approximation to ResMinRetTrees yields an ϵ-approximation for the problem of computing $r_t(\mathcal{T})$. Combining this with the optimality-preserving reduction from ResMinRetTrees to the problem of computing $\ell_t(\mathcal{T})$ described above gives the desired result. □

It follows directly that the following problem is NP-hard and APX-hard.

MinLevTrees

Instance: A set $\mathcal{X}$ of taxa and a set $\mathcal{T}$ of phylogenetic trees on $\mathcal{X}$.

Objective: Construct a level-k phylogenetic network on $\mathcal{X}$ that displays each tree in $\mathcal{T}$ and such that k is as small as possible.

Corollary 4. *Computing $\ell_c(\mathcal{T})$ and computing $\ell_{tr}(\mathcal{T})$ are both NP-hard and APX-hard, even for sets $\mathcal{T}$ consisting of two binary trees on the same set of taxa.*

Proof. Follows from Theorem 2 and Theorem 3. □

Thus, also the following two problems are NP-hard and APX-hard.

MinLevClusters

Instance: A set $\mathcal{X}$ of taxa and a set $\mathcal{C}$ of clusters on $\mathcal{X}$.

Objective: Construct a level-k phylogenetic network on $\mathcal{X}$ that represents each cluster in $\mathcal{C}$ and such that k is as small as possible.

MinLevTriplets

Instance: A set $\mathcal{X}$ of taxa and a set $\mathcal{R}$ of triplets on $\mathcal{X}$.

Objective: Construct a level-k phylogenetic network on $\mathcal{X}$ that is consistent with each triplet in $\mathcal{R}$ and such that k is as small as possible.

Moreover, the latter problem is even NP-hard and APX-hard for dense sets of triplets.

5. Concluding Remarks

In this article, we have proven an important unification result that shows that when computing the minimum number of reticulations (or minimum level) required to represent data obtained from two binary trees on the same taxon set, it does not matter whether one calculates this using trees, triplets or clusters. In the process of proving this, we have clarified a number of confusing issues in the literature.

The unification result has the interesting practical consequence that the two-tree case thus forms an interesting benchmark for comparing the performance of different phylogenetic network software. It was already empirically

observed in [9], for example, that for a specific two-tree data set the independently developed programs Cass (which takes clusters as input, and attempts to minimise level), PIRN (which takes trees as input, and attempts to minimise the reticulation number) and HybridInterleave (which takes two binary trees as input, and minimises the reticulation number) all returned the same optimum. The intriguing possibility thus exists of creating hybrid software for the two-tree problem by combining the best parts of several existing software packages. It should be noted, however, that the *networks* achieving these optima are not always transferrable. For example, a network obtaining the minimum number of reticulations under the cluster model does not automatically display both the trees.

It is also interesting to view our results next to other two-tree findings in the literature. Phillips and Warnow [41] showed that, given a set of clusters coming from two trees, it is polynomial-time solvable to find a phylogenetic tree consistent with a maximum number of clusters, while this problem is NP-hard for three or more trees. Another interesting two-tree result was discovered by Bordewich, Semple and Spillner [42]. They found a polynomial-time algorithm for finding an optimal set of taxa that maximizes the weighted sum of the phylogenetic diversity across two phylogenetic trees, while also this problem is NP-hard for three or more trees. It would be interesting to try and identify general families of objective functions (i.e. optimization criteria) for which the two-tree case is special.

On the other hand, we have shown that the tree, triplet and cluster models already start to diverge for three binary trees on the same set of taxa. A

natural follow-up question is thus: can we predict under what circumstances the models significantly differ, and what does it say about our choice of model if sometimes one model requires significantly more reticulations, or higher level, than another? The "triplet $\leq$ cluster $\leq$ trees" inequality from Section 3.2 suggests that in appropriate combinations existing software for triplets, clusters and trees could be used to develop lower and upper bounds for each other, but under what circumstances are these bounds strong?

References

[1] D. H. Huson, R. Rupp, C. Scornavacca, Phylogenetic Networks, Cambridge University Press, 2011, to appear.

[2] L. Nakhleh, The Problem Solving Handbook for Computational Biology and Bioinformatics, Springer, 2009, Ch. Evolutionary phylogenetic networks: models and issues.

[3] C. Semple, Reconstructing Evolution - New Mathematical and Computational Advances, Oxford University Press, 2007, Ch. Hybridization Networks, pp. 277–314.

[4] M. Bordewich, S. Linz, K. S. John, C. Semple, A reduction algorithm for computing the hybridization number of two trees, Evolutionary Bioinformatics 3 (2007) 86–98.

[5] M. Bordewich, C. Semple, Computing the hybridization number of two phylogenetic trees is fixed-parameter tractable, IEEE/ACM Trans. Comput. Biol. Bioinform. 4 (3) (2007) 458–466.

[6] M. Bordewich, C. Semple, Computing the minimum number of hybridization events for a consistent evolutionary history, Discrete Appl. Math. 155 (8) (2007) 914–928.

[7] J. Collins, S. Linz, C. Semple, Quantifying hybridization in realistic time, to appear in J. Comput. Biol. (2010).

[8] D. H. Huson, R. Rupp, V. Berry, P. Gambette, C. Paul, Computing galled networks from real data, Bioinformatics 25 (12) (2009) i85–i93.

[9] L. J. J. van Iersel, S. M. Kelk, R. Rupp, D. H. Huson, Phylogenetic networks do not need to be complex: Using fewer reticulations to represent conflicting clusters, Bioinformatics 26 (2010) i124–i131, special issue: Proceedings of Intelligent Systems for Molecular Biology 2010 (ISMB2010), 10th-13th September 2010, Boston USA.

[10] Y. Wu, W. Jiayin, Fast computation of the exact hybridization number of two phylogenetic trees, in: Bioinformatics Research and Applications (ISBRA), Vol. 6053, 2010, pp. 203–214.

[11] M. Baroni, S. Grünewald, V. Moulton, C. Semple, Bounding the number of hybridisation events for a consistent evolutionary history, J. Math. Biol. 51 (2005) 171–182.

[12] L. J. J. van Iersel, J. C. M. Keijsper, S. M. Kelk, L. Stougie, F. Hagen, T. Boekhout, Constructing level-2 phylogenetic networks from triplets, in: Research in Computational Molecular Biology (RECOMB), Vol. 4955 of Lecture Notes in Bioinformatics, 2008, pp. 464–476.

[13] Y. Song, Y. Wu, D. Gusfield, Efficient computation of close lower and upper bounds on the minimum number of recombinations in biological sequence evolution, Bioinformatics 21 (Suppl. 1) (2005) i413 – i422.

[14] Y. Wu, Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees, Bioinformatics 26 (2010) i140–i148, special issue: Proceedings of Intelligent Systems for Molecular Biology 2010 (ISMB2010), 10th-13th September 2010, Boston USA.

[15] T. N. D. Huynh, J. Jansson, N. Nguyen, W.-K. Sung, Constructing a smallest refining galled phylogenetic network, in: Research in Computational Molecular Biology (RECOMB), Vol. 3500 of Lecture Notes in Bioinformatics, 2005, pp. 265–280.

[16] J. Jansson, N. B. Nguyen, W.-K. Sung, Algorithms for combining rooted triplets into a galled phylogenetic network, SIAM J. Comput. 35 (5) (2006) 1098–1121.

[17] J. Jansson, W.-K. Sung, Inferring a level-1 phylogenetic network from a dense set of rooted triplets, Theor. Comput. Sci. 363 (1) (2006) 60–68.

[18] L. J. J. van Iersel, S. M. Kelk, Constructing the simplest possible phylogenetic network from triplets, to appear in Algorithmica (2009).

[19] L. J. J. van Iersel, S. M. Kelk, MARLON: Constructing level one phylogenetic networks with a minimum amount of reticulation, `http://homepages.cwi.nl/~kelk/marlon.html` (2008).

[20] L. J. J. van Iersel, J. C. M. Keijsper, S. M. Kelk, L. Stougie, F. Hagen,

T. Boekhout, Constructing level-2 phylogenetic networks from triplets, IEEE/ACM Trans. Comput. Biol. Bioinform. 6 (4) (2009) 667–681.

[21] L. J. J. van Iersel, J. C. M. Keijsper, S. M. Kelk, L. Stougie, LEVEL2: A fast method for constructing level-2 phylogenetic networks from dense sets of rooted triplets, `http://homepages.cwi.nl/~kelk/level2triplets.html` (2007).

[22] T.-H. To, M. Habib, Level-k phylogenetic networks are constructable from a dense triplet set in polynomial time, in: Combinatorial Pattern Matching (CPM), Vol. 5577 of Lecture Notes in Computer Science, 2009, pp. 275–288.

[23] L. J. J. van Iersel, S. M. Kelk, SIMPLISTIC: Simple Network Heuristic, `http://homepages.cwi.nl/~kelk/simplistic.html` (2008).

[24] D. H. Huson, R. Rupp, Summarizing multiple gene trees using cluster networks, in: Workshop on Algorithms in Bioinformatics (WABI), Vol. 5251 of Lecture Notes in Computer Science, 2008, pp. 296–305.

[25] D. H. Huson, T. H. Klöpper, Beyond galled trees - decomposition and computation of galled networks, in: Research in Computational Molecular Biology (RECOMB), Vol. 4453 of Lecture Notes in Computer Science, 2007, pp. 211–225.

[26] L. J. J. van Iersel, S. M. Kelk, R. Rupp, D. H. Huson, CASS: Combining phylogenetic trees into a phylogenetic network, `http://www.win.tue.nl/~liersel/cass.html` (2009).

[27] D. H. Huson, D. C. Richter, C. Rausch, M. Franz, R. Rupp, Dendroscope: An interactive viewer for large phylogenetic trees, BMC Bioinformatics 8 (1) (2007) 460.

[28] D. Gusfield, V. Bansal, V. Bafna, Y. Song, A decomposition theory for phylogenetic networks and incompatible characters, J. Comput. Biol. 14 (10) (2007) 1247–1272.

[29] D. Gusfield, D. Hickerson, S. Eddhu, An efficiently computed lower bound on the number of recombinations in phylognetic networks: Theory and empirical study, Discrete Appl. Math. 155 (6-7) (2007) 806–830.

[30] Y. Wu, D. Gusfield, A new recombination lower bound and the minimum perfect phylogenetic forest problem., J. Comb. Optim. 16 (3) (2008) 229–247.

[31] L. Wang, K. Zhang, L. Zhang, Perfect phylogenetic networks with recombination, J. Comput. Biol. 8 (1) (2001) 69–78.

[32] R. B. Lyngsø, Y. S. Song, J. Hein, Minimum recombination histories by branch and bound, in: Workshop on Algorithms in Bioinformatics (WABI), Vol. 3692 of Lecture Notes in Computer Science, 2005, pp. 239–250.

[33] Y. S. Song, Z. Ding, D. Gusfield, C. H. Langley, Y. Wu, Algorithms to distinguish the role of gene-conversion from single-crossover recombination in the derivation of snp sequences in populations, J. Comput. Biol. 14 (10) (2007) 1273–1286.

[34] D. Gusfield, S. Eddhu, C. Langley, Optimal, efficient reconstruction of phylogenetic networks with constrained recombination, J. Bioinform. Comput. Biol. 2 (2004) 173–213.

[35] D. Gusfield, Different models for phylogenetic networks: how do they relate?, presentation at the Phylogenetics programme at the Isaac Newton Institute (Cambridge, UK) (2007).

[36] I. A. Kanj, L. Nakhleh, C. Than, G. Xia, Seeing the trees and their branches in the network is hard, Theor. Comput. Sci. 401 (1-3) (2008) 153–164.

[37] S. R. Myers, R. C. Griffiths, Bounds on the minimum number of recombination events in a sample history, Genetics 163 (2003) 375–394.

[38] L. J. J. van Iersel, S. M. Kelk, A short experiment to demonstrate a progressively larger tree-cluster gap, `http://homepages.cwi.nl/~kelk/clusters/treeclus3treegap/` (2010).

[39] M. Baroni, C. Semple, M. Steel, Hybrids in real time, Syst. Biol. 55 (2006) 46–56.

[40] S. Linz, C. Semple, Hybridization in non-binary trees, IEEE/ACM Trans. Comput. Biol. Bioinform. 6 (1) (2009) 30–45.

[41] C. Phillips, T. J. Warnow, The asymmetric median tree—a new model for building consensus trees, Discrete. Appl. Math. 71 (1-3) (1996) 311–335.

[42] M. Bordewich, C. Semple, A. Spillner, Optimizing phylogenetic diversity across two trees, Appl. Math. Lett. 22 (5) (2009) 638 – 641.