A dissimilarity measure for semidirected networks

Michael Maxfield[1], Jingcheng Xu[1], Cécile Ané[1,2]

***Abstract*—Semidirected networks have received interest in evolutionary biology as the appropriate generalization of unrooted trees to networks, in which some but not all edges are directed. Yet these networks lack proper theoretical study. We define here a general class of semidirected phylogenetic networks, with a stable set of leaves, tree nodes and hybrid nodes. We prove that for these networks, if we locally choose the direction of one edge, then globally the set of directed paths starting at this edge is stable across all choices to root the network. We define an edge-based representation of semidirected phylogenetic networks and use it to define a dissimilarity between networks, which can be efficiently computed in near-quadratic time. Our dissimilarity extends the widely-used Robinson-Foulds distance on both rooted trees and unrooted trees. After generalizing the notion of tree-child networks to semidirected networks, we prove that our edge-based dissimilarity is in fact a distance on the space of tree-child semidirected phylogenetic networks.**

***Index Terms*—phylogenetic, admixture graph, Robinson-Foulds, tree-child, μ-representation, ancestral profile**

I. Introduction

HISTORICAL relationships between species, virus strains or languages are represented by phylogenies, which are rooted graphs in which the edge direction indicates the flow of time going forward. Semidirected phylogenetic networks are to rooted networks what undirected trees are to rooted trees. They appeared recently, following studies showing that the root and the direction of some edges in the network may not be identifiable from various data types [29, 3]. Consequently, several methods to infer phylogenies from data aim to estimate semidirected networks, rather than fully directed networks, such as `SNaQ` [29], `NANUQ` [1], `admixtools2` [24], `poolfstat` [11], `NetRAX` [23], and `PhyNEST` [19].

The theoretical identifiability of semidirected networks is receiving increased attention [12, 25, 33, 2] but graph theory is still at an early stage for this type of network [but see 22]. In particular, adapted distance and dissimilarity measures are lacking, as are tools to test whether two phylogenetic semidirected networks are isomorphic. These tools are urgently needed for applications. For example, when an inference method is evaluated using simulations, its performance is quantified by how often the inferred network matches the true network used to simulate data, or how similar the inferred network is to the true network. Tools for semidirected networks would also help summarize a posterior sample of networks output by Bayesian inference methods. Even for a basic summary such as the posterior probability of a given topology, we need to decide which semidirected networks are isomorphic in a potentially very large posterior sample of networks.

Unless additional structure is assumed, current methods appeal to a naive strategy that considers all possible ways to root semidirected networks, and then use methods designed for directed networks, e.g. to check if the candidate rooted networks are isomorphic or to minimize a dissimilarity between the two rooted networks across all possible root positions.

In this work, we first generalize the notion of semidirected phylogenetic networks in which edges are either of tree type or hybrid type, such that any edge can be directed or undirected. We relax the constraint of a single root (of unknown position). Multi-rooted phylogenetic networks were recently introduced, although for directed networks, to represent the history of closely related and admixed populations [30] or distant groups of species that exchanged genes nonetheless [14, 15]. Our general definition requires care to define a set of leaves consistent across all compatible directed phylogenetic networks.

For these semidirected networks, we define an edge-based "μ-representation" μ_E, extending the node-based μ-representation, denoted here as μ_V, by Cardona, Rosselló, and Valiente [6]. The "ancestral profile" of a rooted network contain the same information as the node-based representation μ_V, that is: the number of directed paths from each node to each leaf [5]. So our edge-based representation μ_E also extends the notion of ancestral profile to semidirected networks, in which ancestral relationships are unknown between some nodes because the root is unknown. Briefly, μ_E's information for an edge depends on whether the edge direction is known or implicitly constrained by the direction of other edges in the network. For example, if N is a semidirected tree and an edge is explicitly or implicitly directed, then μ_E associates the edge to the cluster of taxa below it. If instead the edge direction depends on the unknown placement of the root(s), then μ_E associates the edge to the bipartition of the taxa obtained by deleting the edge from the semidirected tree. If N has reticulations, μ_V uses μ-vectors to generalize the notion of clusters, storing the number of directed paths from a given node to each taxon. Our extension μ_E associates a directed edge to the μ-vector of its child node, and an undirected edge to two μ-vectors: one for each direction that the edge can take. To handle and distinguish semidirected networks with multiple roots, μ_E also associates each root to a μ-vector, well-defined even when the exact root position(s) are unknown. We then define a dissimilarity measure d_{μ_E} between semidirected networks.

Our network representation μ_E can be calculated in polynomial time, namely $\mathcal{O}(n|E|)$ where n is the number of leaves and $|E|$ is the number of edges. The associated network

[1] Department of Statistics, University of Wisconsin - Madison, USA.
[2] Department of Botany, University of Wisconsin - Madison, USA.

dissimilarity d_{μ_E} can then be calculated in $\mathcal{O}\big(|E|(n+\log|E|)\big)$ time. It provides the first dissimilarity measure adapted to semidirected networks (without iterative network re-rooting) that can be calculated in polynomial time. Linz and Wicke [22] also recently considered semidirected networks (with a single root of unknown position). They showed that "cut edge transfer" rearrangements, which transform one network into another, define a finite distance on the space of level-1 networks with a fixed number of hybrid nodes. This distance is NP-hard to compute, however, because it extends the subtree prune and regraft (SPR) distance on unrooted trees [13].

Finally, we generalize the notion of tree-child networks to semidirected networks, and prove that d_{μ_E} is a true distance on the subspace of tree-child semidirected networks, extending the result of [6] to semidirected networks using an edge-based representation. On trees, this dissimilarity equals the widely used Robinson-Foulds distance between *unrooted* trees [28]. We provide concrete algorithms to construct μ_E from a given semidirected network, and to reconstruct the network from μ_E.

As a proper distance, d_{μ_E} can decide in polynomial time if two tree-child semidirected networks are isomorphic. For rooted phylogenetic networks, the isomorphism problem is solvable in linear time when restricted to tree-child networks [18], but otherwise is polynomially equivalent to the general graph isomorphism problem even if restricted to tree-sibling time-consistent rooted networks [9]. As semidirected networks include single-rooted networks, the graph isomorphism problem for semidirected networks is necessarily more complex. The general graph isomorphism problem was shown to be of subexponential complexity [4] but is not known to be solvable in polynomial time. Therefore, there is little hope of finding a dissimilarity that can be computed in polynomial time, and that is a distance for general semidirected phylogenetic networks. Hence, dissimilarities like d_{μ_E} offer a balance between computation time and the extent of network space in which it can discriminate between distinct networks.

II. Basic definitions for semidirected graphs

For a graph G we denote its vertex set as $V(G)$ and its edge set as $E(G)$. The subgraph induced by a subset of vertices $V' \subseteq V(G)$ is denoted as $G[V']$, and the edge-induced subgraph is denoted as $G[E']$ for a subset $E' \subseteq E(G)$.

We use the following terminology for a directed graph $G = (V, E)$. For a node v, its *in-degree* is denoted as $\deg_{\mathrm{i}}(v, G)$ and *out-degree* as $\deg_{\mathrm{o}}(v, G)$. Its *total degree* $\deg(v, G)$ is $\deg_{\mathrm{i}}(v, G) + \deg_{\mathrm{o}}(v, G)$. We may omit G when no confusion is likely. For $u, v \in V$, we write $u > v$ if there exists a directed path $u \rightsquigarrow v$. A node v is a *leaf* if $\deg_{\mathrm{o}}(v) = 0$; v is an *internal* node otherwise. We denote the set of leaves as $V_L(G)$. A node v is a *root* if $\deg_{\mathrm{i}}(v) = 0$; a *tree node* if $\deg_{\mathrm{i}}(v) \leq 1$; and a *hybrid node* otherwise. We denote the set of tree nodes as V_T and the set of hybrid nodes as V_H. An edge $(u, v) \in E$ is a *tree edge* if v is a tree node; a *hybrid edge* otherwise. We denote the set of tree edges as E_T and the set of hybrid edges as E_H. A *descendant* of v is any node u such that $v > u$. A *tree path* is a directed path consisting only of tree edges. A *tree descendant leaf* of v is any leaf u such that there exists a tree path $v \rightsquigarrow u$. An *elementary path* in G is a directed path such that the first node has out-degree 1 in G and all intermediate nodes have in-degree and out-degree 1 in G. In the remainder, we use "path" to mean "directed path" for brevity, unless otherwise specified.

We now extend these notions to semidirected graphs.

Definition 1 (**semidirected graph**). A *semidirected graph* N is a tuple $N = (V, E)$, where V is the set of vertices, and $E = E_U \sqcup E_D$ is the set of edges, E_U being the set of undirected edges and E_D the set of directed edges.

Undirected edges in E_U are denoted as uv for some $u, v \in V$, instead of the standard notation of $\{u, v\}$ for brevity. Directed edges in E_D are denoted as (u, v) for some $u, v \in V$, implying the direction from u to v, with u referred to as the *parent* of the edge, and v as its *child*. A directed graph is a semidirected graph with no undirected edges: $E_U = \emptyset$.

For $v \in V(N)$, $\deg_{\mathrm{i}}(v, N)$ denotes the number of directed edges with v as their child, $\deg_{\mathrm{o}}(v, N)$ the number of directed edges with v as their parent, $\deg_{\mathrm{u}}(v, N)$ the number of undirected edges in N incident to v, and $\deg = \deg_{\mathrm{i}} + \deg_{\mathrm{o}} + \deg_{\mathrm{u}}$. We may omit N when no confusion is likely. Furthermore, $\mathrm{child}(v, N) = \{w \in V : (v, w) \in E_D\}$. N is *binary* if $\deg(v) = 1$ or 3 for all nodes. N is *bicombining* if $\deg_{\mathrm{i}} v = 2$ for all hybrid nodes.

A semidirected graph $N' = (V, E')$ is *compatible* with another semidirected graph $N = (V, E)$ if N' can be obtained from N by directing some undirected edges in N.

The *contraction* of N, denoted as $\mathrm{Cont}(N)$, is the directed graph obtained by contracting every undirected edge in N. It is well defined, as it can be viewed as the quotient graph of N under the partition that groups nodes connected by a series of undirected edges. For $v \in V(N)$, $\mathrm{Cont}(v, N)$ is defined as the node in $\mathrm{Cont}(N)$ which v gets contracted into.

In a semidirected graph N, a node $v \in V(N)$ is a *root* if it only has outgoing edges. It is a *tree node* if $\deg_{\mathrm{i}}(v, N) \leq 1$. Otherwise, v is called a *hybrid node*. The set of tree nodes is denoted as V_T or $V_T(N)$ and the set of hybrid nodes as V_H or $V_H(N)$. A *tree edge* is an undirected edge, or a directed edge whose child is a tree node. A *hybrid edge* is a directed edge whose child is a hybrid node. We denote the set of tree edges as E_T or $E_T(N)$ and the set of hybrid edges as E_H or $E_H(N)$.

A semidirected *cycle* is a semidirected graph if its undirected edges can be directed so that it becomes a directed cycle. A semidirected graph is *acyclic* if it does not contain a semidirected cycle. We refer to directed acyclic graphs as DAGs and to semidirected acyclic graphs as SDAGs.

Next, we define a more stringent notion of compatibility to maintain the classification of nodes and edges as being of tree versus hybrid type, illustrated in Figs. 1 and 2.

Definition 2 (**phylogenetically compatible, rooted partner, network**). An SDAG N' is *phylogenetically compatible* with another SDAG N if N' is compatible with N and $E_H(N') = E_H(N)$. A *rooted partner* of N is a DAG that is phylogenetically compatible with N. A *multi-root semidirected network*, or *network* for short, is an SDAG that admits a rooted partner.

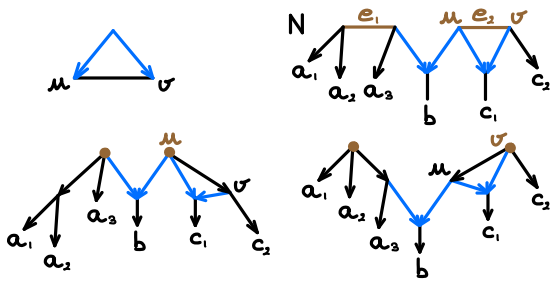

Fig. 1. Examples of SDAGs and rooted partners. Top left: SDAG that is not a network, because it has no rooted partner: directing uv would cause u or v to become a hybrid node and result in a non-phylogenetically compatible DAG. Top right: N is a network. Bottom: 2 of N's 4 rooted partners. Each rooted partner has 2 roots (brown dots), one from each brown edge in N. Of the nodes incident to e_2 for example, either u (bottom left) or v (bottom right) can serve as root.

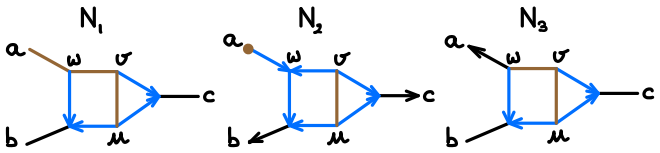

Fig. 2. Examples to illustrate definitions and notations. Directed edges (E_D) are shown with arrows; hybrid edges (E_H) in blue. Root components, whose nodes (V_R) can serve as roots, are shown in brown. N_2 is compatible but not phylogenetically compatible with N_1 (e.g. w is hybrid in N_2 and not in N_1). N_3 is phylogenetically compatible with N_1. Rooted leaves are b and c in N_1 and N_2; a is an ambiguous leaf in N_1. N_3 has no ambiguous leaves so it is an $\mathcal{L}$-network on $\mathcal{L} = V_{UL}(N_3) = V_{RL}(N_3) = \{a, b, c\}$. N_1 has 1 root component. Its rooted partner rooted at u is tree-child, but none of the others are (rooted at a, w or v), so it is weakly tree-child. N_2 has 2 root components $R_1 = \{a\}$ and $R_2 = \{u, v\}$, and 2 rooted partners: one from the root choice $\rho(R_1) = a$, $\rho(R_2) = u$; and the other from the root choice $\rho(R_1) = a$, $\rho(R_2) = v$. N_2 is not tree-child in either sense, as none of its rooted partners are tree-child. $N_2 = \mathcal{C}(N_2)$ is complete. N_3 is not complete: in $\mathcal{C}(N_3)$, edges incident to b and c are directed.

Note that if SDAG N' is phylogenetically compatible with SDAG N, then $V_T(N) = V_T(N')$ and $V_H(N) = V_H(N')$. Note also the rooted partner of a binary network may not be binary by the typical definition for rooted networks, since the root can have degree 3 or 1 instead of 2.

Not all SDAGs are networks (see Fig. 1 for an example). We are interested in networks rather than general SDAGs, because a network can represent evolutionary history up to "rerooting", as captured by its rooted partners. Fig. 1 gives an example network (N, top) and 2 of its 4 rooted partners (bottom).

Traditionally, phylogenetic trees and networks are connected and with a single root [31]. We consider here a broader class of graphs, allowing for multiple connected components and multiple roots per connected component. We also allow for non-simple graphs, that is, for multiple parallel edges between the same two nodes u and v, directed and in the same direction for the graph to be acyclic. With a slight abuse of notation, we keep referring to each parallel edge as (u, v). Self-loops are not allowed as they would cause the graph to be cyclic. The term "rooted partner" was introduced by Linz and Wicke [22] in the context of traditional semidirected phylogenetic networks in which the set of directed edges is precisely the set of hybrid edges, and for which a rooted partner has a single root.

We will use the following results frequently. The first one is trivial.

Proposition 1. *Let N be an SDAG phylogenetically compatible with SDAG N'. Then $E_D(N) \setminus E_D(N') \subseteq E_T(N)$.*

Proposition 2. *Let N be a network, and N' the semidirected graph obtained from N by undirecting some of the tree edges of N. Then N' is a network, and N is phylogenetically compatible with N'.*

Proof. Let A be the set of tree edges in N that are undirected to obtain N'. We first consider the case when A consists of a single edge (u, v). We shall establish the following claims:

1) N' is a network;
2) the edges of N' in $E(N) \setminus A$ retain their type (undirected, tree, or hybrid);
3) N is phylogenetically compatible with N'.

For claim 1, suppose for contradiction that N' is not an SDAG. Then there exists in N' a semidirected cycle C which contains uv. Let G be a rooted partner of N, and C^+ the subgraph made of the corresponding edges in C. Since C^+ cannot be a directed cycle, there exist two hybrid edges (a, h) and (b, h) in C^+. Both are also directed in N by phylogenetic compatibility. Because they are hybrid edges, they are distinct from (u, v), so they are directed in N' and C. This contradicts C being a semidirected cycle, showing that N' is an SDAG. Since it also admits G as a rooted partner, N' is a network.

Claim 2 follows from the observation that the types of nodes (tree vs hybrid) in N' stays the same. Claim 3 follows from claim 2.

If A contains multiple edges, then we iteratively undirect one edge at a time. By the previous argument, the resulting graph at each step is a network with which N is phylogenetically compatible, because phylogenetic compatibility is transitive. Hence N' is a network and N is phylogenetically compatible with it. □

The next definition is motivated by the fact that phylogenies are inferred from data collected at leaves, which are known entities with labels (individuals, populations, or species), whereas non-leaf nodes are inferred and unlabeled.

Definition 3 (**unrooted, rooted and ambiguous leaves**). A node v in a network N is a *rooted leaf* in N if v is a leaf in every rooted partner of N; and an *unrooted leaf* if it is a leaf in some rooted partner N. We denote the set of rooted leaves and unrooted leaves as $V_{RL}(N)$ and $V_{UL}(N)$ respectively. An *ambiguous leaf* is a node in $V_{UL}(N) \setminus V_{RL}(N)$.

Clearly, $V_{RL}(N) \subseteq V_{UL}(N)$. If N is directed, then $V_{RL}(N) = V_{UL}(N) = V_L(N)$. As we will see later in Lemma 12, an ambiguous leaf is of degree 1 in the undirected graph obtained by undirecting all edges in N, hence a leaf in the traditional sense. In Fig. 2 for example, a is an ambiguous leaf in N_1 but a rooted leaf in N_3.

In Section III we make no assumption regarding $V_{UL}(N)$ and $V_{RL}(N)$. But to extend μ-vectors to semidirected graphs in Section IV, we will need a stable set of leaves across rooted partners. For a vector of distinct labels $\mathcal{L}$, an $\mathcal{L}$-DAG is a

DAG whose leaves are bijectively labeled by $\mathcal{L}$. To extend this definition we impose $V_{UL}(N) = V_{RL}(N)$ below, but we will see (after Lemma 12) that this requirement is less stringent than it appears.

Definition 4 (**leaf-labeled network**). A network N is *labeled in* $\mathcal{L}$ and called an *$\mathcal{L}$-network*, if $V_{UL}(N) = V_{RL}(N)$ and $V_{UL}(N)$ is bijectively labeled by elements of $\mathcal{L}$. The *leaf set* of an $\mathcal{L}$-network N is defined as $V_L(N) = V_{UL}(N)$.

For example, in Fig. 2 N_3 is an $\mathcal{L}$-network with $\mathcal{L} = \{a, b, c\}$, while N_1 is not because $V_{UL}(N_1) \neq V_{RL}(N_1)$. N_2 is a $\{b, c\}$-network.

A rooted partner G of an $\mathcal{L}$-network N must be an $\mathcal{L}$-DAG: since $V_{RL}(N) \subseteq V_L(G) \subseteq V_{UL}(N)$ generally, we have that $V_L(G) = V_L(N)$ and $V_L(G)$ can inherit the labels in N.

Our main result concerns the class of tree-child graphs. If G is a DAG, it is *tree-child* if every internal node v of G has at least one child that is a tree node [16]. Equivalently, G is tree-child if every non-leaf node in G has a tree descendant leaf [6]. We extend this notion to semidirected networks, illustrated in Fig. 2.

Definition 5 (**semidirected tree-child**). A network is *weakly tree-child* if one of its rooted partners is tree-child. It is *strongly tree-child*, or simply *tree-child*, if all its rooted partners are tree-child.

Since a DAG G is a network with a single rooted partner, G is strongly and weakly tree-child if and only if it is tree-child as a DAG. In Fig. 2, N_3 is weakly but not strongly tree-child: of its 3 rooted partners, only one is tree-child. Later in Proposition 11, we provide a characterization that is easy to check without enumerating rooted partners.

III. Properties of semidirected acyclic graphs

Proposition 3. *A semidirected graph $N = (V, E)$ is acyclic if and only if the undirected graph induced by its undirected edges E_U consists of trees only (i.e. is a forest), and* $\mathrm{Cont}(N)$ *is acyclic.*

Proof. Let N be a semidirected graph. Suppose that $N[E_U]$ is not a forest. Then there exists a compatible directed graph of $N[E_U]$ which contains a cycle that exists in a compatible directed graph of N, therefore N is not acyclic. Next suppose that $\mathrm{Cont}(N)$ contains a cycle C'. Then there exists a compatible directed graph G of N which contains a cycle C such that C' is obtained from C after contracting edges in G that correspond to undirected edges in N, and so N is not acyclic.

Now suppose the graph induced by E_U is a forest and that $\mathrm{Cont}(N)$ is acyclic. If N is not acyclic, then by definition there exists a compatible directed graph G that contains a cycle C. Since $\mathrm{Cont}(N)$ is acyclic, C must contract into a single node in $\mathrm{Cont}(N)$. This implies that C contains undirected edges only, hence is contained in $N[E_U]$, a contradiction. Therefore N is acyclic. □

In a network N, a *semidirected path* from u_0 to u_n is a sequence of nodes $u_0u_1 \dots u_n$, such that for $i = 1, \dots n$, either $u_{i-1}u_i$ or (u_{i-1}, u_i) is an edge in N. On $V(N)$ we define $v \lesssim u$ if there is a semidirected path from u to v, and the associated equivalence relation: $u \sim v$ if $u \lesssim v$ and $v \lesssim u$. On equivalence classes, $\lesssim$ becomes a partial order. In Fig. 2 for example, $a \lesssim v$ in N_1 and N_3. Also $v \lesssim a$ so $a \sim v$ in N_1, but not in N_3. In N_3, $\{u, v, w\}$ and $\{a\}$ are two equivalence classes. There is a semidirected path from u to a (for example) so $\{a\} \lesssim \{u, v, w\}$ and $\{a\}$ is not maximal under $\lesssim$. Using these partially ordered classes, we can define the following.

Definition 6 (**undirected components, root components, directed part**). In a network N, an *undirected component* is the subgraph induced by an equivalence class under $\sim$. A *root component* of N is an undirected component that is maximal under $\lesssim$. A root component is *trivial* if it consists of a single node. The set of edges and nodes that are not in a root component is called the *directed part* of N. We denote the set of nodes in the directed part as $V_{DP}(N)$ and the set of edges $E_{DP}(N)$. We also denote the set of nodes in root components $V_R(N)$ and the set of edges $E_R(N)$.

The directed part of N is generally not a subgraph: it may contain an edge but not one of its incident nodes. In Fig. 2, edges in the directed part E_{DP} are those in black (tree edges) and blue (hybrid edges). Edges in root components E_R are in brown. In N_3, v is adjacent to the directed part, but is not in $V_{DP}(N_3)$. N_2 has a trivial root component: $\{a\}$. The following result justifies the name given to the equivalence classes.

Proposition 4. *For u, v nodes in a network N, $u \sim v$ if and only if there is an undirected path between u and v.*

Proof. Consider u and v connected by an undirected path. Since this path does not contain any directed edge, clearly, $u \lesssim v$ and $v \lesssim u$, hence $u \sim v$.

Now suppose $u \sim v$. By definition, there exists a semidirected path $p_{uv} = u_0u_1 \dots u_n$ from $u = u_0$ to $v = u_n$ and a semidirected path p_{vu} from v to u. If $p_{vu} = u_nu_{n-1} \dots u_0$ then all edges in these paths must be shared and undirected. This is because N is acyclic, in case p_{uv} and p_{vu} have distinct edges incident to u_i and u_{i+1}. Then, there is an undirected path between u and v as claimed. Otherwise, there exists $i_1 \geq 0$ and $i_2 > i_1 + 1$ such that p_{vu} is the concatenation of semidirected paths $u_nu_{n-1} \dots u_{i_2}$; $p_{u_{i_2}u_{i_1}}$ from u_{i_2} to u_{i_1} not containing any u_j for $i_1 < j < i_2$; and $u_{i_1}u_{i_1-1} \dots u_0$. Then, the concatenation of $p_{u_{i_2}u_{i_1}}$ with $u_{i_1}u_{i_1+1} \dots u_{i_2}$ forms a semidirected cycle. Since N is acyclic, this case cannot occur. □

We now characterize undirected components as the undirected trees in the forest induced by N's undirected edges.

Proposition 5. *Let N be a network and F the graph induced by the undirected edges of N. Then F is a forest where:*

1) *each tree corresponds to an undirected component of N, and has at most one node v with* $\deg_{\mathrm{i}}(v, N) \geq 1$*;*
2) *the root components of N are exactly the trees without such nodes, and they contain tree nodes only.*

Proof. By Proposition 3, F is a forest. By Proposition 4, each tree in F is an undirected component. If a tree T in F had more than one node v with $\deg_{\mathrm{i}}(v, N) \geq 1$, it would be impossible

to direct the edges in T without making one of them hybrid, contradicting the existence of a rooted partner of N.

For the second claim, let T be a tree in F. Note that $(u,v) \in E_D$ implies that $u \not\sim v$ and v's equivalence class is not maximal under $\lesssim$. So if T is maximal under $\lesssim$, then all nodes v in T have $\deg_{\text{i}}(v,N) = 0$, which implies that v is a tree node. Conversely, if T is not maximal, then there exist nodes v in T, $u \gtrsim v$ in a different tree of F, and a semidirected path from u to v containing a directed edge. Taking the last directed edge on this path gives a node v' in T with $\deg_{\text{i}}(v',N) \geq 1$. □

Lemma 6. *Let N be a network and G a rooted partner of N. Let $v \in V_R(N)$. If $v < u$ in G, then $u \in V_R(N)$.*

Proof. Let $v < u$ in G. Then $v \lesssim u$ in N. By Definition 6, $u \lesssim v$ in N as well. Therefore, u belongs to the same root component as v, hence $u \in V_R(N)$. □

Proposition 7. *Let N be a network. Let G_1 and G_2 be rooted partners of N. Then edges in $E_{DP}(N)$ have the same direction in G_1 and G_2.*

Proof. Let T be an undirected component of N that is not a root component. We need to show that T's edges have the same direction in G_1 and G_2. By Proposition 5, T is an undirected tree in N, and has exactly one node v_0 with an incoming edge in N. Then v_0 must be the root of T in both G_1 and G_2, for G_1 and G_2 to be phylogenetically compatible with N, which completes the proof. □

Definition 7 (**root choice function**). Let N be a network, and $\mathcal{R}$ be the set of root components of N. A *root choice function* of N is a function $\rho : \mathcal{R} \to V(N)$ such that for a root component $T \in \mathcal{R}$, $\rho(T) \in V(T)$. In other words, ρ chooses a node for each root component.

Conceptually, a semidirected network represents uncertainty about the root(s) position. Next, we show that to resolve uncertainty, exactly 1 node from each root component must be chosen as root, and this choice can be made independently across root components. In other words, root choice functions are in one-to-one correspondence with rooted partners. As a result, all rooted partners have the same number of roots: the number of root components. In Fig. 1, N has 2 root components each with 2 nodes, hence $2 \times 2 = 4$ rooted partners.

Proposition 8. *Given a root choice function ρ of a network N, there exists a unique rooted partner N^+_ρ of N such that the set of roots in N^+_ρ is the image of ρ. Conversely, given any rooted partner G of N, there exists a unique root choice function ρ such that $G = N^+_\rho$.*

Proof. Let G_0 be a rooted partner of N. For a root choice function ρ, let N^+_ρ be the graph compatible with N obtained by directing edges in $E_{DP}(N)$ as they are in G_0, and away from $\rho(T)$ in any root component T (which is possible by Proposition 5). N^+_ρ is a DAG because N is acyclic. To prove that N^+_ρ is phylogenetically compatible with N (and hence a rooted partner), we need to show that $E_H(N^+_\rho) = E_H(G_0)$. Since all edges in N incident to both $V_R = V_R(N)$ and $V_{DP} = V_{DP}(N)$ are directed from V_R to V_{DP}, they have the same direction in N^+_ρ and G_0. Therefore nodes in V_{DP} and edges in $E_{DP}(N)$ are of the same type in N^+_ρ and G_0 (and N). Furthermore, by construction and Proposition 5, all edges in $E_R(N)$ remain of tree type in both N^+_ρ and G_0. Hence N^+_ρ is a rooted partner of N. Finally, the root set of N^+_ρ is the image of ρ because a root component's root is a root of the full network (from $\deg_{\text{i}}(v,N) = 0$ for any $v \in V_R$) and because V_{DP} cannot contain any root of N^+_ρ (by Proposition 5 again).

To prove that N^+_ρ is unique, let G be a rooted partner of N whose set of roots is the image of ρ. By Proposition 7, edges in $E_{DP}(N)$ have the same direction in G and N^+. For a root component T, $G[V(T)] = N^+_\rho[V(T)]$ because T is an undirected tree in N, rooted by the same $\rho(T)$ in both G and N^+_ρ. Therefore $G = N^+_\rho$.

Let G be a rooted partner of N. By Proposition 5 and phylogenetic compatibility, G must have exactly one root in each root component T. Define ρ such that $\rho(T)$ is this root of G in T. By the arguments above, G cannot have any root in V_{DP}, and then $G = N^+_\rho$. □

We can define the following thanks to Proposition 7:

Definition 8 (**network completion**). The *completion* $\mathcal{C}(N)$ of a network N is the semidirected graph obtained from N by directing every undirected edge in its directed part, as it is in any rooted partner of N. More precisely, let G be a rooted partner of N. We direct $uv \in E_{DP}(N)$ as (u,v) in $\mathcal{C}(N)$ if $(u,v) \in E(G)$. A network N is *complete* if $\mathcal{C}(N) = N$.

Proposition 9. *For a network N, $\mathcal{C}(N)$ is a network and phylogenetically compatible with N.*

Proof. From Proposition 8, any rooted partner of N is of the form N^+_ρ. As seen in the proof of Proposition 8, $\mathcal{C}(N)$ and N^+_ρ differ in root components only: if $e \in E_R(N)$ then e is undirected in N and in $\mathcal{C}(N)$, directed in N^+_ρ, and is a tree edge in all. Therefore $\mathcal{C}(N)$ is phylogenetically compatible with N and is a network (admitting N^+_ρ as rooted partner). □

Remark 1. Propositions 5 and 8 yield practical algorithms. Finding the root components requires only traversing the network N, tracking the forest F of undirected components, and which nodes have nonzero in-degree. Computing $\mathcal{C}(N)$ then consists in directing the edges away from such a node in each tree of F, if one exists. In particular, in a single traversal of N we can construct a rooted partner G, record the roots of G, record the edges of G that were in root components of N, and for each such edge record the corresponding root. To do this, for each tree T in F that is a root component, we arbitrarily choose and record a node u as root, direct the edges in T away from u, record these edges as belonging to a root component of N, and for all these edges record u as the corresponding root. We then direct the rest of the undirected edges the same way as when computing the completion. We shall use this in Algorithm 1 later.

With as many directed edges as can be possibly implied by the directed edges in N, $\mathcal{C}(N)$ is the network that we are generally interested in. We define phylogenetic isomorphism between networks based on their completion.

Definition 9 (**network isomorphism**). $\mathcal{L}$-networks N and N' are *phylogenetically isomorphic*, denoted by $N \cong N'$, if $\mathcal{C}(N)$ and $\mathcal{C}(N')$ are isomorphic as semidirected graphs, with an isomorphism that preserves the leaf labels.

In Fig. 2 for example, N_2 is complete, but N_3 is not. N_1 and N_3 are not phylogenetically isomorphic, because the edge incident to a remains undirected in the completion $\mathcal{C}(N_1)$. N_2 is not phylogenetically compatible with N_1 or N_3, and not phylogenetically isomorphic to either.

Lemma 10. *Let N be a complete network. There exists a directed edge (u, v) in N if and only if $v \in V_{DP}(N)$.*

Proof. If $v \in V_{DP}(N)$, its undirected component U has no undirected edges by Definition 8. Then by Proposition 5, $U = \{v\}$ and $\deg_{\mathrm{i}}(v) \geq 1$, so v is the child of some directed edge. Conversely, if there exists $(u, v) \in E(N)$ then v's equivalence class is not maximal and $v \in V_{DP}(N)$. □

Using $\mathcal{C}(N)$, we can now decide if a network is weakly or strongly tree-child in a single traversal, thanks to the following.

Proposition 11. *Let N be a complete network, $\mathcal{R}$ its set of root components, and W_0 the set of nodes that form trivial root components. For $T \in \mathcal{R}$, let $W_1(T)$ be the set of nodes u in T adjacent to $V_{DP}(N)$ with $\deg_{\mathrm{u}}(u) = 1$ in N and without a tree child in N. N is weakly (resp. strongly) tree-child if and only if every non-leaf node in $V_{DP}(N) \cup W_0$ has a tree child in N; and for every $T \in \mathcal{R}$, $|W_1(T)| \leq 1$ (resp. $W_1(T) = \emptyset$).*

If N is a DAG, then $V_{DP}(N) \cup W_0$ is the full node set and we simply recover the tree-child definition. Recall that children are defined using directed edges only. For example, take N in Fig. 1. In $\mathcal{C}(N)$ the edges incident to b and c_1 are directed; e_i remains undirected and forms a root component T_i ($i = 1, 2$). $W_1(T_1) = \emptyset$ but $W_1(T_2) = \{u\}$, because u is incident to exactly 1 undirected edge and 2 outgoing hybrid edges. By Proposition 11, N is weakly tree-child. Indeed, its partners rooted at u are tree-child (e.g. Fig. 1 bottom left) but its partners rooted at v are not (e.g. bottom right). The proof below shows that, more generally, a weakly tree-child network has at most one 'problematic' node in each root component, and this node must serve as root for a rooted partner to be tree-child.

Proof of Proposition 11. We first characterize which rooted partners are tree-child. Suppose that N is complete, and that every non-leaf node in $V_{DP}(N) \cup W_0$ has a tree child in N. For a root choice function ρ, we claim that the rooted partner N^+_ρ is tree-child if and only if $W_1(T) \subseteq \{\rho(T)\}$ for every $T \in \mathcal{R}$. To prove this claim, consider a node u in N. If u is in $V_{DP}(N) \cup W_0$ then u has the same children in N as in any rooted partner, so its tree child in N is also its tree child in N^+_ρ. Otherwise, u is in some $T \in \mathcal{R}$, T is non-trivial and $\deg_{\mathrm{u}}(u) \geq 1$. If $\deg_{\mathrm{u}}(u) \geq 2$, then at least one of its neighbors in T is its tree child in N^+_ρ. If $\deg_{\mathrm{u}}(u) = 1$ and is not adjacent to $V_{DP}(N)$, then $\deg_{\mathrm{o}}(u) = 0$ and u is a leaf in N^+_ρ or its unique neighbor is its tree child in N^+_ρ. If u has a tree child w in N, then w is also its tree child in N^+_ρ. Otherwise, $u \in W_1(T)$. Let v be its unique neighbor in T. If $\rho(T) \neq u$ then v is the parent of u in N^+_ρ, u has the same children in N^+_ρ as in N, so u has no tree child in N^+_ρ (by definition of $W_1(T)$). If $\rho(T) = u$ then v is a child of u in N^+_ρ, and since v is a tree node (because $v \in T$) u has a tree child in N^+_ρ. Overall, we get that N^+_ρ is tree-child if and only if any node in any $W_1(T)$ is a root, which proves the claim.

For the first direction of Proposition 11, suppose that N is complete and weakly tree-child. If u is a non-leaf node in $V_{DP}(N) \cup W_0$, then u has the same children in N as in any rooted partner, so u has a tree child in N. Also, we can apply our claim. Since N^+_ρ is tree-child for some ρ, by our claim we get $W_1(T) \subseteq \{\rho(T)\}$ which implies that $|W_1(T)| \leq 1$ for each $T \in \mathcal{R}$. Suppose further that N is strongly tree-child. If $W_1(T) \neq \emptyset$ for some $T \in \mathcal{R}$ then T is nontrivial, we can choose $\rho(T) = v \notin W_1(T)$ for which N^+_ρ is not tree-child, a contradiction. This proves the first direction.

For the second direction, assume that every non-leaf node in $V_{DP}(N) \cup W_0$ has a tree child in N and $|W_1(T)| \leq 1$ (resp. $W_1(T) = \emptyset$) for every $T \in \mathcal{R}$. Then N admits at least one tree-child rooted partner (setting ρ such that $\{\rho(T)\} = W_1(T)$ for any T with $W_1(T) \neq \emptyset$) and N is weakly tree-child. Further, if $W_1(T) = \emptyset$ for all T, then N^+_ρ is tree-child for any root choice function ρ, and N is strongly tree-child. □

Finally, we give a characterization of unambiguous leaves that leads to a fast algorithm.

Lemma 12. *In a network N, v is an ambiguous leaf if and only if $v \in V_R(N)$, $\deg_{\mathrm{u}}(v) = 1$, and $\deg_{\mathrm{o}}(v) = \deg_{\mathrm{i}}(v) = 0$.*

Proof. Let v be an ambiguous leaf. Then $\deg_{\mathrm{o}}(v) = 0$. By Proposition 7, v is in $V_R(N)$ so $\deg_{\mathrm{i}}(v) = 0$. Finally, if $\deg_{\mathrm{u}}(v) = 0$, then v is isolated, and a rooted leaf. If $\deg_{\mathrm{u}}(v) \geq 2$, then in any rooted partner one of the incident edges is directed away from v, and v is never a leaf. Therefore $\deg_{\mathrm{u}}(v) = 1$.

Conversely, let $v \in V_R(N)$ be incident to exactly one edge, uv. By Proposition 8, we can find a rooted partner where v is a non-leaf root as well as a rooted partner where u is a root and v is a leaf. Hence v is an ambiguous leaf. □

Remark 2. We argue that requiring $V_{UL}(N) = V_{RL}(N)$ for N to be an $\mathcal{L}$-network is reasonable in practice. By Lemma 12, an ambiguous leaf x is in a root component and incident to only one undirected edge. The ambiguity is whether x becomes a leaf or a root in a rooted partner. In practice, one knows which nodes are supposed to be leaves, with a label and data [10, 26]. Then one can, for each root component, either direct all incident edges to ambiguous leaves towards them, making them rooted leaves (as in Fig. 2, compare N_1 and N_3), or pick one ambiguous leaf to serve as root for that component (as in N_2, Fig. 2) and direct edges accordingly, turning the remaining ambiguous leaves into rooted leaves. This yields a network with $V_{UL} = V_{RL}$. By Proposition 5, this can be done in $\mathcal{O}(|E|)$ where $|E|$ is the number of edges.

IV. Vectors and Representations

Formally a multiset is a tuple (A, m), where A is a set and $m\colon A \to \mathbb{Z}^+$ gives the multiplicity. To simplify notation,

we use $\wr\int$ to denote a multiset by enumerating each element as many times as its multiplicity. For example, $A = \wr a, a, b\int$ contains a with multiplicity 2 and b with multiplicity 1. For brevity, we identify a multiset (A, m) with the set A if $m \equiv 1$, e.g. $\wr a, b, c\int = \{a, b, c\}$. We adopt the standard notion of sum and difference for multisets. The symmetric difference between multisets is defined as $A \triangle B = (A - B) + (B - A)$.

In what follows, we consider a vector of distinct labels $\mathcal{L}$, whose order is arbitrary but fixed, as it will determine the order of coordinates in all μ vectors and representations. For an $\mathcal{L}$-DAG G and for node $v \in V(G)$, the μ-vector of v is defined as the tuple $\mu(v, G) = (\mu_1(v), \ldots, \mu_n(v))$ where n is the number of labels in $\mathcal{L}$ and $\mu_i(v)$ is the number of paths in G from v to the leaf with the i^{th} label in $\mathcal{L}$. As in [6], the partial order $\geq$ between μ-vectors is the coordinatewise order. Namely, for $m = (m_1, \ldots, m_n)$ and $m' = (m'_1, \ldots, m'_n)$, $m \geq m'$ if $m_i \geq m'_i$ for all $i = 1, \ldots, n$. If $m \not\leq m'$ and $m \not\geq m'$ then m and m' are *incomparable*. The node-based μ-representation of G from [6], denoted as $\mu_V(G)$, is defined as the multiset $\wr \mu(v, G) : v \in V(G)\int$. Algorithm 1 in [6] computes $\mu_V(G)$ recursively in post-order thanks to the following property, which is a slight extension of Lemma 4 in [6] allowing for parallel edges by summing over child edges instead of child nodes.

Lemma 13. *Let G be a DAG and u a node in G. Then*

$$\mu(u, G) = \sum_{v \in \text{child}(u,G)} \sum_{(u,v) \in E(G)} \mu(v, G) \,.$$

We will make frequent use of the following result, whose original proof easily extends to DAGs with parallel edges thanks to Lemma 13. It is an extension of Lemma 5 of [6] stating the assumption used in the proof by [6], which is weaker than requiring a tree-child DAG.

Lemma 14. *Let G be an $\mathcal{L}$-DAG and u, v two nodes in G.*

1) *If there exists a path $u \rightsquigarrow v$, then $\mu(u, G) \geq \mu(v, G)$.*
2) *If $\mu(u, G) > \mu(v, G)$ and if v has a tree descendant leaf, then there exists a path $u \rightsquigarrow v$.*
3) *If $\mu(u, G) = \mu(v, G)$, v has a tree descendant leaf and $u \neq v$, then u, v are connected by an elementary path.*

Other results in [6] similarly hold when parallel edges are allowed, such as their Theorem 1 on tree-child networks (which must be non-binary if they have parallel edges).

The rest of the section is organized as follows. In part IV-A we define the edge-based μ-representation for $\mathcal{L}$-networks, with Algorithm 1 to compute it. Part IV-B presents properties of this μ-representation for tree-child networks, and part IV-C describes how to reconstruct a complete tree-child $\mathcal{L}$-network from its edge-based μ-representation. The networks in Fig. 3 are used as examples throughout.

A. Edge-based μ-representation

We first extend the notion of μ-vectors to nodes in the directed part of an $\mathcal{L}$-network.

Definition 10 (**μ-vector of a node in the directed part**). Let N be an $\mathcal{L}$-network and G any rooted partner of N. For $v \in V_{DP}(N)$, we define $\mu(v, N) = \mu(v, G)$.

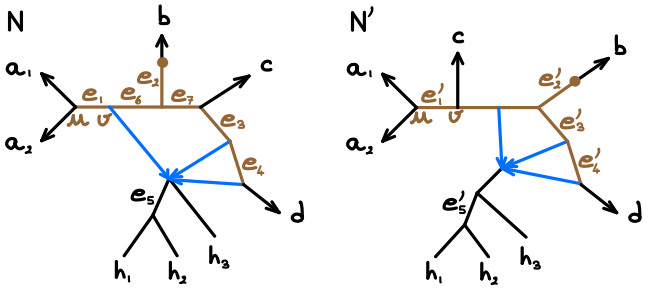

Fig. 3. Tree-child $\mathcal{L}$-networks on $\mathcal{L} = (a_1, a_2, b, c, d, h_1, h_2, h_3)$ with one root component. Directed edges are shown with arrows. Black: tree edges in the directed part, leading to A_1 in Algorithm 1. Blue: hybrid edges, leading to A_2 in Algorithm 1. Brown: edges in the root component (E_R), leading to A_4 in Algorithm 1. For $i \leq 5$, edge e_i in N (left) and e'_i in N' (right) share the same μ-vector set. The multisets $\mu_E(N)$ and $\mu_E(N')$ have 17 elements in common: 8 corresponding to edges incident to leaves, 5 from edges e_i and e'_i for $i \leq 5$, 3 from hybrid edges, and 1 root μ-vector set. $\mu_E(N)$ has 2 unique elements (from e_6 and e_7) and $\mu_E(N')$ has 3, so $d_{\mu_E}(N, N') = 5$. See the Appendix for details.

This is well-defined thanks to the next proposition.

Proposition 15. *Let N be an $\mathcal{L}$-network, and $v \in V_{DP}(N)$. Then the set of directed paths starting at v, and consequently $\mu(v, G)$, are the same for any rooted partner G of N.*

Proof. Let G be a rooted partner of N. Let $u_1 \ldots u_n$ ($n \geq 1$), where $u_1 = v$, be a directed path starting at v in G. We claim $u_i, i = 1, \ldots, n$ are all in $V_{DP}(N)$: Otherwise, we can find i such that $u_i \in V_{DP}(N)$ and $u_{i+1} \in V_R(N)$. Since $(u_i, u_{i+1}) \in E(G)$, either $u_i u_{i+1}$ or (u_i, u_{i+1}) is in $E(N)$. By Proposition 4, this implies either $u_i \in V_R(N)$ or $u_{i+1} \in V_{DP}(N)$, a contradiction. Therefore any directed paths from v in G lies entirely in $G[V_{DP}(N)]$. The conclusion then follows from Proposition 7. □

In Fig. 3 for example, e_5 is in the directed part of N. Applying Lemma 13 recursively on h_1, h_2 and their parent v in $\mathcal{C}(N)$, we get $\mu(v, N) = (0,0,0,0,0, 1,1,0)$ with leaf order given by $\mathcal{L} = (a_1, a_2, b, c, d, h_1, h_2, h_3)$. Then the hybrid node of N (parent of e_5) has μ-vector $(0,0,0,0,0, 1,1,1)$.

Now we turn to root components. Here the μ-vector for a node is not well-defined as it varies depending on the rooted partner. It turns out that if locally we choose the direction of an edge uv, say (u, v), then globally the set of directed paths from v across all rooted partners are the same, and consequently the μ-vector of v is fixed. To prove this, we first establish a lemma.

Lemma 16. *Let N be a network with uv an edge in some root component. Let N' be the semidirected graph obtained from N by directing uv as (u, v). Then N' is a network phylogenetically compatible with N.*

Proof. Let G be a rooted partner of N where u is a root. Let $A = E(G) \setminus E(N)$ be the set of edges that are directed in G but not in N. By phylogenetic compatibility, A consists of tree edges only. A also contains (u, v).

Since from G we get back N' if we undirect all the edges in $A \setminus \{(u, v)\}$, by Proposition 2, G is phylogenetically compatible with N'. Then $E_H(N') = E_H(G) = E_H(N)$ and N' is phylogenetically compatible with N. □

Proposition 17. *Let N be an $\mathcal{L}$-network with $uv \in E_R(N)$. Then the set of directed paths starting at v, and consequently $\mu(v, G)$, are the same for any rooted partner G of N where uv is directed as (u, v), and there always exists such a rooted partner.*

Proof. The existence of a rooted partner where uv is directed as (u, v) follows from Proposition 8.

Let G_1 and G_2 be rooted partners of N such that uv is directed as (u, v) in both. Let N' be the semidirected graph obtained from N by directing uv as (u, v). By Lemma 16, N' is a network phylogenetically compatible with N. Thus, G_1 and G_2 are rooted partners of N'. Since $\deg_{\text{i}}(v, N') \geq 1$, $v \in V_{DP}(N')$ by Proposition 5. The conclusion then follows from Proposition 15. □

Using Proposition 17, we can define the following.

Definition 11 (**directional μ-vector**). Let N be an $\mathcal{L}$-network, $uv \in E_R(N)$, and G any rooted partner of N where uv is directed as (u, v). We call $\mu(v, G)$ the *directional μ-vector of* (u, v), and write it as $\mu_d(u, v, N)$, or $\mu_d(u, v)$ if N is clear in the context.

In Fig. 3 for example, $e_1 = uv$ is in the root component of N, with directional μ-vectors: $\mu_d(v, u, N) = (1,1,0,0,0, 0,0,0)$ and $\mu_d(u, v, N) = (0,0,1,1,1, 3,3,3)$. In N', e'_1 has these same directional μ-vectors.

Next, we associate each root component (rather than a node or edge) to a μ-vector.

Definition 12 (**root μ-vector**). Let N be an $\mathcal{L}$-network and T a root component of N. The *root μ-vector* of T is defined as $\mu(\rho(T), N^+_\rho)$ where ρ is any root choice function of N. We write it as $\mu_r(T, N)$ or $\mu_r(T)$ if N is clear from context.

This is well-defined thanks to the following result.

Lemma 18. *Let N be an $\mathcal{L}$-network, and T a root component of N. Then the μ-vector $\mu(\rho(T), N^+_\rho)$ is independent of the root choice function ρ. Furthermore, if uv is an edge in T, this μ-vector is equal to $\mu_d(u, v) + \mu_d(v, u)$.*

Proof. The claims obviously hold when T is trivial. Now consider T non-trivial and distinct nodes $u \neq v$ in T. Let G_u (resp. G_v) be a rooted partner of N where u (resp. v) is a root. To prove the first claim, we show that $\mu(u, G_u) = \mu(v, G_v)$ by constructing a bijection f_u between $\mathcal{P}_u$ and $\mathcal{P}_v$, where $\mathcal{P}_z$ $(z = u, v)$ is the set of directed paths from z to x in G_z, for an arbitrary but fixed leaf x of N. Suppose $p_u = u \dots w \dots x \in \mathcal{P}_u$, where w is the last node such that $u \dots w$ lies in T. We can modify p_u to a new path p_v by changing the $u \dots w$ subpath to $v \dots w$, the unique tree path between v and w in T. By Lemma 6, the subpath $w \dots x$ only contain edges in $E_{DP}(N)$. Then by Proposition 7, $p_v \in \mathcal{P}_v$. Obviously, f_u is a bijection whose inverse is the map from $\mathcal{P}_v$ to $\mathcal{P}_u$ constructed by symmetry, proving the first claim.

For the second claim, let uv be an edge in T and x, G_u, G_v, $\mathcal{P}_u$ and $\mathcal{P}_v$ as before. Let $\mathcal{P}^v_u$ be the set of directed paths from v to x in G_u, such that $|\mathcal{P}^v_u|$ is the coordinate value for x in $\mu_d(u, v)$. Define $\mathcal{P}^u_v$ similarly. We can partition $\mathcal{P}_u = A \sqcup B$ where paths in A contain v, and paths in B do not. It is easily verified that $B = \mathcal{P}^u_v$, and that prepending u to a path gives a bijection $\mathcal{P}^v_u \to A$. Since x is arbitrary, we get $\mu_r(T) = \mu_d(u, v) + \mu_d(v, u)$. □

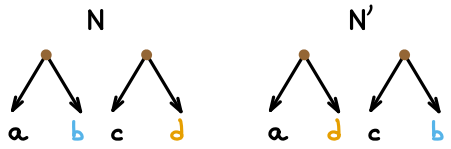

Fig. 4. Networks illustrating the need to include the μ-vector of trivial root components in Definition 13, to distinguish networks with multiple roots. N and N' are tree-child and non-isomorphic. They have the same multiset of edge μ-vector sets, but different root μ-vectors so $\mu_E(N) \neq \mu_E(N')$.

In Fig. 3 for example, N has one root component T (in brown), with $\mu_r(T, N) = (1,1,1,1,1, 3,3,3)$. The root component of N' has the same root μ-vector.

We are now ready to define the edge-based μ-representation of an $\mathcal{L}$-network, and an algorithm to compute it.

Definition 13 (**edge-based μ-representation**). Let N be a complete $\mathcal{L}$-network. To edge e of N we associate a set $\mu(e)$, called the *edge μ-vector set* of the edge e, as follows:

- For $e = (u, v)$, by Lemma 10 we have $v \in V_{DP}$, and we define $\mu(e) = \{(\mu(v, N), t)\}$ using Definition 10, where t is a tag taking value :**t** if e is a tree edge, and :**h** otherwise.
- For $e = uv \in E_R$, using Definition 11 we define $\mu(e) = \{(\mu_d(u, v), :\mathbf{t}), (\mu_d(v, u), :\mathbf{t})\}$.

Let $\mathcal{R}$ be the set of root components of N, then the *edge-based μ-representation* of N, denoted by $\mu_E(N)$, is defined as the multiset

$$\{\!\{\mu(e) : e \in E(N)\}\!\} + \{\!\{\{(\mu_r(T), :\mathbf{r})\} : T \in \mathcal{R}\}\!\}$$

with μ_r from Definition 12 and :**r** a tag value indicating a root μ-vector. For an $\mathcal{L}$-network N', $\mu_E(N')$ is defined as $\mu_E(\mathcal{C}(N'))$.

In Fig. 3, $\mu_E(N)$ contains 19 μ-vector sets: 9 in A_1, 3 in A_2, 6 in A_3 and 1 in A_4, using notations as in Algorithm 1 below. A_1 contains the unidirectional μ-vector sets from the 8 edges incident to the leaves, such as $\{((0,0,0,0,0, 1,0,0), :\mathbf{t})\}$ for the edge to h_1, and $\{((0,0,0,0,0, 1,1,0), :\mathbf{t})\}$ from e_5. A_2 is from the hybrid edges: $\{((0,0,0,0,0, 1,1,1), :\mathbf{h})\}$ with multiplicity 3. A_3 has only 1 element: $\{((1,1,1,1,1, 3,3,3), :\mathbf{r})\}$. Finally, A_4 contains the bidirectional μ-vector sets from the 6 edges in the root component(s). For example, e_1 contributes $\{((1,1,0,0,0, 0,0,0), :\mathbf{t}), ((0,0,1,1,1, 3,3,3), :\mathbf{t})\}$ See the Appendix for the other 5.

Lemma 18 together with Proposition 5 yields the following Algorithm 1 to compute the edge-based μ-representation of an $\mathcal{L}$-network $N = (V, E)$ with n leaves. As discussed in Remark 1, line 1 in Algorithm 1 takes a single traversal of N and $\mathcal{O}(|E|)$ time. Computing the node-based μ-representation by Algorithm 1 in [6] takes $\mathcal{O}(n|E|)$ time. The remaining steps iterate over edges and take $\mathcal{O}(n|E|)$ time, giving an overall complexity of $\mathcal{O}(n|E|)$.

Compared to the node-based representation μ_V, μ_E has two features. Unsurprisingly, each undirected edge (whose direction is not resolved by completion) is represented as

Algorithm 1 Given $\mathcal{L}$-network N, compute its edge-based μ-representation $A = \mu_E(N)$

1: compute a rooted partner G of N, and store:
   - R the set of roots in G
   - ρ: $V_R(N) \to R$ the function that maps a node in a root component T of N to the root of T in G
   - E_R^+ the set of edges in G that corresponds to $E_R(N)$
   - $E_{DP}^+ = E(G) \setminus E_R^+$
2: compute the node-based μ-representation of G, let $\mu = \mu_V(\cdot, G)$
3: $A_1 \leftarrow \wr\{(\mu(v), :\mathbf{t})\} : (u,v) \in E_{DP}^+ \cap E_T(G)\int$
4: $A_2 \leftarrow \wr\{(\mu(v), :\mathbf{h})\} : (u,v) \in E_{DP}^+ \cap E_H(G)\int$
5: $A_3 \leftarrow \wr\{(\mu(r), :\mathbf{r})\} : r \in R\int$
6: $A_4 \leftarrow \wr\{(\mu(v), :\mathbf{t}), (\mu(\rho(v)) - \mu(v), :\mathbf{t})\} : (u,v) \in E_R^+\int$
7: **return** $A = A_1 + A_2 + A_3 + A_4$

bidirectional using two μ-vectors. This is similar to the representation of edges in unrooted trees, as bipartitions of $\mathcal{L}$. The second feature is the inclusion of a μ-vector for each root component, which may seem surprising. For a non-trivial root component T, $\mu_r(T)$ is redundant with information from $\mu(e)$ for any e in T, by Lemma 18. The purpose of including the root μ-vectors in $\mu_E(N)$ is to keep information from trivial root components, for networks with multiple roots. Without this information, μ_E cannot discriminate simple networks with multiple roots when one or more root component is trivial, as illustrated in Fig. 4.

Networks with a unique and non-trivial root component correspond to standard phylogenetic rooted networks, with uncertainty about the root location. For these networks, we could use edge μ-vectors only: $\mu_E(N) = \wr\mu(e) : e \in E(N)\int$, that is, omit A_3 in Algorithm 1. Indeed, the root μ-vector of the unique root component T is redundant with $\mu(e)$ of any edge e in T. For this class of standard networks, then, our results below also hold using the simplified definition of μ_E.

B. Properties for tree-child networks

We will use the following results to reconstruct a tree-child network from its edge-based μ-representation. First we characterize when and how μ-vectors are comparable.

Proposition 19. *Let T_1 and T_2 be distinct nontrivial root components of a strongly tree-child $\mathcal{L}$-network N. Then directional μ-vectors from T_1 and from T_2 are incomparable to one another.*

Proof. Let $uu' \in E(T_1)$ and $vv' \in E(T_2)$. Suppose for contradiction that $\mu_d(u,u') \geq \mu_d(v,v')$. Let G be a rooted partner of N in which u and v are roots. Then $\mu_d(u,u') = \mu(u',G)$ and $\mu_d(v,v') = \mu(v',G)$. Since G is tree-child, there exists a path $u' \rightsquigarrow v'$ in G by Lemma 14 (possibly up to relabeling if $\mu_d(u,u') = \mu_d(v,v')$) Since (v,v') is a tree edge in G, by Lemma 1 in [6], $u' \rightsquigarrow v'$ contains or is contained in (v,v'). Both cases imply that $u' \in \{v,v'\}$ (using that v is a root for the first case), a contradiction. □

Proposition 20. *In a weakly tree-child $\mathcal{L}$-network, different root components have incomparable root μ-vectors.*

Proof. Let $T_1 \neq T_2$ be root components of a tree-child $\mathcal{L}$-network N. In a tree-child rooted partner of N, there is no directed path between the roots of T_1 and T_2. Therefore, by Lemmas 14 and 18, $\mu_r(T_1)$ and $\mu_r(T_2)$ are incomparable. □

Lemma 21. *Let N be a strongly tree-child $\mathcal{L}$-network. Suppose uv, st are two (not necessarily distinct) edges in root component T of N such that the undirected tree path from u to t in T contains v and s. Then:*

1) $\mu_d(u,v) \geq \mu_d(s,t)$,
2) $\mu_d(v,u)$ is incomparable to $\mu_d(s,t)$,
3) $\mu_d(u,v)$ is incomparable to $\mu_d(t,s)$.

Proof. Let G_u (resp. G_v) be a rooted partner of N with u (resp. v) as a root. For part 1, by Lemma 14, we have $\mu_d(u,v) = \mu(v,G_u) \geq \mu(t,G_u) = \mu_d(s,t)$.

For part 2, by symmetry, it suffices to show that $\mu_d(v,u) \not\leq \mu_d(s,t)$. Let $w_1, \ldots, w_k$ be the neighbors of u besides v. Then $\mu(w_i,G_u) = \mu(w_i,G_v)$ by Proposition 15 if $w_i \in V_{DP}(N)$, or Proposition 17 if $w_i \in V(T)$. Then by Lemma 13 we have $\mu_d(v,u) = \mu(u,G_v) = \sum_{i=1}^k \sum_{(u,w_i)\in E(G_v)} \mu(w_i,G_v) = \sum_{i=1}^k \sum_{(u,w_i)\in E(G_v)} \mu(w_i,G_u)$.

First, suppose for contradiction that $\mu_d(v,u) < \mu_d(s,t) = \mu(t,G_u)$. Then for each i, $\mu(t,G_u) > \mu(w_i,G_u)$, and since G_u is tree-child there exists a path $t \rightsquigarrow w_i$ in G_u by Lemma 14. As u is a root in G_u and not contained in these paths, $w_1, \ldots, w_k$ are hybrid nodes. Then u does not have a tree child in G_v, a contradiction.

Now suppose instead $\mu_d(v,u) = \mu_d(s,t) = \mu(t,G_u)$, then $\mu(t,G_u) \geq \mu(w_i,G_u)$ for all i. If $\mu(t,G_u) = \mu(w_i,G_u)$ for some i, then $w_i = w_1$ is the only neighbor of u other than v. By Lemma 14, t and w_1 are connected by an elementary path in G_u, which is impossible as both have u as a parent. Therefore $\mu(t,G_u) > \mu(w_i,G_u)$ for all i, which leads to a contradiction by the argument above.

Part 3 follows from part 2 using that $\mu_d(a,b) + \mu_d(b,a) = \mu_r(T)$ for any $ab \in E(T)$ by Lemma 18. □

Next, we relate edges with identical directional μ-vectors.

Lemma 22. *Let N be a strongly tree-child $\mathcal{L}$-network, and x a fixed μ-vector. If we direct all edges $uv \in E_R(N)$ with $\mu_d(u,v) = x$ as (u,v), then these edges form a directed path. If the path is nonempty, we denote the first node as $h(x,N)$.*

Proof. Let $E_x = \{uv \in E_R(N) : \mu_d(u,v) = x\}$. Take uv and st in E_x. By Proposition 19, they are in the same root component T, so there is an undirected path p in T connecting uv and st. By applying Lemma 21 multiple times and permuting labels if necessary, we may assume p is of the form $uv \ldots st$ with $\mu_d(u,v) = \mu_d(s,t) = x$. For a rooted partner G of N with u a root, we have $\mu(v,G) = \mu(t,G) = x$, which implies by Lemma 14 there is an elementary path in G from v to t. By Lemma 6, this path lies in $E_R(N)$. But since $E_R(N)$ induces a forest, this elementary path must be the $v \ldots st$ part of the path p. Therefore all the intermediate nodes w in p have $\deg_{\mathrm{u}}(w,N) = 2$ and $\deg_{\mathrm{i}}(w,N) = \deg_{\mathrm{o}}(w,N) = 0$. Furthermore, if w_1 and w_2 are consecutive nodes in p, then $w_1w_2 \in E_x$ because $x = \mu(v,G) \geq \mu(w_2,G) = \mu_d(w_1,w_2) \geq \mu(t,G) = x$.

Therefore all edges in p are in E_x, and form a directed path when directed as in the statement.

Now take an undirected path p_0 of edges in E_x, of maximum length, and let e one of its edges. To show that p_0 contains all the edges in E_x, take $e' \in E_x$. By the previous argument, e and e' are connected by a tree path p_1. Also by the previous argument, all intermediate nodes in p_0 and in p_1 have $\deg_u = 2$. Since p_1 cannot extend p_0 by definition of p_0, p_1 must be contained in p_0, therefore e' is in p_0 as claimed. □

Finally, the next result was proved for orchard DAGs, which include tree-child DAGs [8, Proposition 10]. We restrict its statement to hybrid nodes here, because we allow networks to have in and out degree-1 nodes.

Lemma 23. *Let G be a tree-child $\mathcal{L}$-DAG. Let u, v be distinct hybrid nodes in G. Then $\mu(u, G) \neq \mu(v, G)$.*

C. Reconstructing a complete tree-child network

To reconstruct a complete tree-child network N from its edge-based μ-representation, Algorithm 2 will first construct $\mu_V(G)$ for a rooted partner G from $\mu_E(N)$. Then Algorithm 3 will use $\mu_E(N)$ to undirect some edges in G and recover N.

Algorithm 2 Given $A = \mu_E(N)$ from a tree-child $\mathcal{L}$-network N, compute $B = \mu_V(G)$ for some rooted partner G of N

Input: multiset A
Output: multiset B

1: $B_1 \leftarrow \{\!\{x\colon \{(x,:\mathbf{t})\} \in A\}\!\}$
2: $B_2 \leftarrow \{x\colon \{(x,:\mathbf{h})\} \in A\}$
3: $B_3 \leftarrow \{x\colon \{(x,:\mathbf{r})\} \in A\}$
4: $B_4 \leftarrow \{\!\{\}\!\}$
5: **for** z in B_3 **do**
6: $\quad M(z) \leftarrow \{\!\{x : \{(x,:\mathbf{t}), (z-x,:\mathbf{t})\} \in A\}\!\}$
7: $\quad$ **if** $M(z)$ is empty **then** skip to next iteration
8: $\quad r(z) \leftarrow$ some arbitrary element of $M(z)$
9: $\quad$ **for** $\{(x_1,:\mathbf{t}), (x_2,:\mathbf{t})\} \in A$ with $x_1 + x_2 = z$ **do**
10: $\quad\quad B_4 \leftarrow B_4 + \{\!\{y\}\!\}$ where $y = x_i$ if $x_i \leq r(z)$ else $y = z - x_i$ if $x_i > r(z)$ ($i = 1$ or 2)
11: $B \leftarrow B_1 + B_2 + B_3 + B_4$
12: **return** B

Continuing with N in Fig. 3 (left), $A = \mu_E(N)$ is given in the Appendix. Algorithm 2 starts with $B_1 = \{(1,0,0,0,0,\,0,0,0), \ldots, (0,0,0,0,0,\,0,0,1), (0,0,0,0,0,\,1,1,0)\}$ for edges incident to leaves and e_5 (in black in Fig. 3). $B_2 = \{(0,0,0,0,0,\,1,1,1)\}$, for the unique μ-vector shared by all 3 hybrid edges in N. B_3 has a single element $z = (1,1,1,1,1,\,3,3,3)$ because N has a single root component, so the loop on line 5 has a single iteration and all elements $\{(x_1,:\mathbf{t}), (x_2,:\mathbf{t})\}$ in A satisfy $x_1 + x_2 = z$ (from edges in brown in Fig. 3). On line 6, $M(z)$ has 12 elements (see the Appendix). We can arbitrarily pick $r(z) = (0,0,0,1,1,\,2,2,2) \in M(z)$, which corresponds to e_7 directed rightward. Then $B = \mu_V(G)$ for the partner G of N rooted at the node incident to e_6 and e_7 (see Fig. A7).

Proof of correctness for Algorithm 2. As N and $\mathcal{C}(N)$ have the same rooted partners and $\mu_E(N) = \mu_E(\mathcal{C}(N))$, we may assume N to be complete.

Let ρ be a root choice function such that $\rho(T) = h(r(\mu_r(T)), N)$, where r is the function on line 8 and h is defined in Lemma 22. By Lemma 18 and Proposition 20, $\rho(T) \in V(T)$ is well-defined. Let $G = N^+_\rho$. We shall show that Algorithm 2, with $A = \mu_E(N)$ as input, produces the output $B = \mu_V(G)$.

Consider partitioning $V(N) = V(G)$ into the following sets:

$V_1 = \{v$ is a tree node in the directed part of $N\}$,
$V_2 = \{v$ is a hybrid node$\}$,
$V_3 = \{v$ is a root in $G\}$,
$V_4 = \{v$ in a root component of N, but not a root in $G\}$.

We will establish $B_i = \{\!\{\mu(v, G) : v \in V_i\}\!\}$ for the multisets B_i in the algorithm ($i = 1, \ldots, 4$), to conclude the proof.

By Lemma 10, $(u, v) \mapsto v$ is a bijection between the directed tree edges and V_1. Then by Definition 13, $B_1 = \{\!\{\mu(v, N) : (u, v) \in E_T(N)\}\!\} = \{\!\{\mu(v, G) : v \in V_1\}\!\}$, which concludes case $i = 1$.

For $i = 2$, Lemma 23 implies that $\mu(u, G) \neq \mu(v, G)$ for distinct $u \neq v$ in V_2. Therefore $\{\!\{\mu(v, G) : v \in V_2\}\!\} = \{\mu(v, G) : v \in V_2\}$. By the definition of hybrid edges, $B_2 = \{x\colon \{(x,:\mathbf{h})\} \in A\} = \{\mu(v, G) : (u, v) \in E_H(N)\}$ is equal to $\{\mu(v, G) : v \in V_2\}$, which implies $B_2 = \{\!\{\mu(v, G) : v \in V_2\}\!\}$.

For $i = 3$, by Lemma 18 and Proposition 20, the μ-vectors of the roots in G are the same as the root μ-vectors, and are all distinct. Hence $B_3 = \{\!\{\mu(v, G) : v \in V_3\}\!\}$.

For $i = 4$, let E^+_R be the set of edges in G that corresponds to $E_R(N)$. Consider the map $V_4 \to E^+_R$ that associates v to its parent edge (u, v) in G. It is well-defined because V_4 excludes the roots of G, root components only contain tree nodes (Proposition 5) and $uv \in E_R(N)$ by Lemma 10. Furthermore, the map is a bijection. Therefore we have $\{\!\{\mu(v, G) : v \in V_4\}\!\} = \{\!\{\mu_d(u, v) : (u, v) \in E^+_R\}\!\}$.

B_4 is constructed in Algorithm 2 by taking a μ-vector from the pair $\mu_d(s, t)$ and $\mu_d(t, s)$, for each undirected edge st in each root component. Let T be the root component that contains st and let $z = \mu_r(T) \in B_3$. Then $\mu_d(s, t) + \mu_d(t, s)$ equals z but no other root μ-vector by Proposition 20, so $\mu(st)$ is considered at exactly one iteration of the loop on line 9. Next we need to show that on line 10, exactly one μ-vector gets chosen, and is $y = \mu_d(s, t)$ where $(s, t) \in E^+_R$.

From Lemma 22, let $u = h(r(z), N)$ be the root of T in G and v such that $r(z) = \mu_d(u, v)$. Since u is a root in G and $(s, t) \in E(G)$, the tree path p in T from u to t contains s. If p also contains v, then $\mu_d(s, t) \leq \mu_d(u, v)$ and $\mu_d(t, s) \not\leq \mu_d(u, v)$ by Lemma 21, so line 10 defines $y = \mu_d(s, t)$ as claimed. If p does not contain v, then the tree path from v to t contains u and s, so by Lemma 21 $\mu_d(s, t)$ is incomparable to $\mu_d(u, v)$ and $\mu_d(t, s) \geq \mu_d(u, v)$. Further, $\mu_d(t, s) > \mu_d(u, v)$ by the choice $u = h(r(z), N)$ and Lemma 22. Therefore line 10 defines $y = z - \mu_d(t, s) = \mu_d(s, t)$, which concludes the proof. □

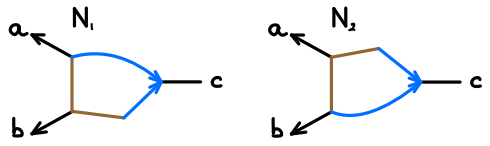

Fig. 5. Weakly tree-child (a, b, c)-networks for which Theorem 1 does not hold. For each network, the rooted partner rooted at the degree-2 node is tree-child. The other 2 rooted partners are not. N_1 and N_2 are not phylogenetically isomorphic yet $\mu_E(N_1) = \mu_E(N_2) = A_1 + \cdots + A_4$ with $A_1 = \wr\{((1,0,0),:\mathbf{t})\}, \{((0,1,0),:\mathbf{t})\}, \{((0,0,1),:\mathbf{t})\}\int$, $A_2 = \wr\{((0,0,1),:\mathbf{h})\}, \{((0,0,1),:\mathbf{h})\}\int$, $A_3 = \wr\{((1,1,2),:\mathbf{r})\}\int$ and $A_4 = \wr\{((1,0,1),:\mathbf{t}),((0,1,1),:\mathbf{t})\}, \{((1,1,1),:\mathbf{t}),((0,0,1),:\mathbf{t})\}\int$.

Algorithm 3 Given $A = \mu_E(N)$ from a tree-child $\mathcal{L}$-network N, and a rooted partner G of N, modify G to obtain $\mathcal{C}(N)$

1: $B \leftarrow \mu_E(G)$
2: $F \leftarrow \wr x : \{(x,:\mathbf{t})\} \in B - A\int$
3: **for** $x \in$ Unique(F) **do**
4: $\quad m(x) \leftarrow$ multiplicity of x in F
5: $\quad p(x) \leftarrow$ the directed path in G formed by $\{(u,v) \in E(G) : \mu_V(v,G) = x\}$
6: $\quad$ undirect the first $m(x)$ edges in $p(x)$
7: **return** G

Given $\mu_E(N)$ from N in Fig. 3 and G from Algorithm 2 (Fig. A7), F contains 6 μ-vectors (see the Appendix) so 6 edges in G are undirected to obtain N. One of them $x = (0,0,1,0,0,0,0,0)$ has multiplicity 1 in F but corresponds to an elementary path of 2 edges in G adjacent to b. On this path, only e_2 is undirected by Algorithm 3.

Proof of correctness of Algorithm 3. Note that line 5 uses Lemma 14 to claim that $p(x)$ is a directed path, and so line 6 can be applied.

Let E_R^+ denote the set of edges in G that corresponds to edges in $E_R(N)$. It suffices to show that line 6 undirects all edges in E_R^+ and no other. Obviously, line 2 defines $F = \wr\mu(v,G) : (u,v) \in E_R^+\int$. Suppose F consists of elements $x_1, \ldots, x_k$ with multiplicities $m_1, \ldots, m_k$. We know that E_R^+ exactly consists of m_i edges whose children have μ-vector x_i, for $i = 1, \ldots, k$. Thus we only need to show that for each x_i, the first m_i edges in $p(x_i)$ are in E_R^+. By Lemma 6, if an edge e is not in E_R^+, then all edges below it are also not in E_R^+. Therefore along the path $p(x_i)$, edges in E_R^+ must come first before any edge not in E_R^+, which finishes the proof. □

The following theorem derives directly from Theorem 1 in [6] to reconstruct G from $\mu_V(G)$, and the application of Algorithms 2 and 3.

Theorem 1. *Let N_1 and N_2 be strongly tree-child $\mathcal{L}$-networks. Then $\mu_E(N_1) = \mu_E(N_2)$ if and only if N_1 and N_2 are phylogenetically isomorphic.*

Theorem 1 does not generally hold for weakly tree-child networks, as seen in a counter-example in Fig. 5.

V. The edge-based μ-distance

Definition 14 (**edge-based μ-distance**). Let N_1 and N_2 be $\mathcal{L}$-networks. The *edge-based* μ-dissimilarity between N_1 and N_2 is defined as

$$d_{\mu_E}(N_1, N_2) = |\mu_E(N_1) \triangle \mu_E(N_2)| \,.$$

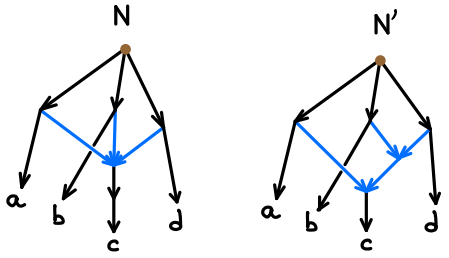

Fig. 6. Example rooted networks for which the node-based and edge-based μ distances differ, on leaves (a, b, c, d): Left: N is tree-child, non-bicombining. Right: N' is not tree-child, but bicombining. Both have 3 nodes (including leaf c) with μ-vector (0,0,1,0), and $d_{\mu_V}(N, N') = 0$. N and N' both have 5 edges with μ-vector (0,0,1,0), of which 3 (resp. 4) are of hybrid type in N (resp. N'), and $d_{\mu_E}(N, N') = 2$.

For example, $d_{\mu_E}(N, N') = 2$ in Fig. 6. For the networks in Fig. 3, $d_{\mu_E}(N, N') = 5$ due to non-matching μ-vector sets for edges e_6 and e_7 in N and the 3 unlabelled tree edges in N'. (see Fig. 3 and the Appendix for details).

We are now ready to state our main theorem, which justifies why we may refer to d_{μ_E} as a distance.

Theorem 2. *For a vector of leaf labels $\mathcal{L}$, d_{μ_E} is a distance on the class of (complete) strongly tree-child $\mathcal{L}$-networks.*

Proof. From the properties of the symmetric difference, d_{μ_E} is a dissimilarity in the sense that is it symmetric, non-negative, and satisfies the triangle inequality. It remains to show that d_{μ_E} satisfies the separation property. Let N_1 and N_2 be tree-child $\mathcal{L}$-networks. If $d_{\mu_E}(N_1, N_2) = 0$ then $\mu_E(N_1) = \mu_E(N_2)$ and by Theorem 1, $N_1 \cong N_2$. □

For unrooted trees, the μ-vector of each undirected edge encodes the bipartition on $\mathcal{L}$ associated with the edge, hence d_{μ_E} agrees with the Robinson-Foulds distance on unrooted trees.

On rooted trees, d_{μ_E} agrees with d_{μ_V}. Indeed, if T is a directed tree or forest on $\mathcal{L}$, then each non-root node v has a unique parent edge e with element $\{(\mu_V(v),:\mathbf{t})\}$ in $\mu_E(T)$; and each root u forms to a trivial root component with element $\mu_E(u) = \{(\mu_V(u),:\mathbf{r})\}$ in $\mu_E(T)$.

However, d_{μ_E} does not generally extend d_{μ_V}. For example, consider the rooted networks in Fig. 6. They have the same μ_V representation, hence $d_{\mu_V}(N, N') = 0$. However, their μ_E representations differ, due to edges with the same μ vector (1 path to c only) but different tags (tree edge in N versus hybrid edge in N'). Hence d_{μ_E} can distinguish these networks: $d_{\mu_E}(N, N') > 0$.

We can compute d_{μ_E} using a variant of Algorithm 3 in [6]. Specifically, we first group the elements of $\mu_E(N_i)$ $(i = 1, 2)$ by their type: of the form $\{(x,:\mathbf{r})\}$, $\{(x,:\mathbf{t})\}$, $\{(x,:\mathbf{h})\}$, or $\{(x,:\mathbf{t}),(y,:\mathbf{t})\}$. Then it suffices to equip a total order and apply Algorithm 3 in [6] to each group, then add the distances obtained from the 4 groups. For the first 3 types we can simply use the lexical order on the μ-vector x. For the last type, we

may compare two elements by comparing the lexically smaller μ-vector first and then the larger one, to obtain a total order within the group.

As in [6], with $\mu_E(N_i)$ computed and sorted, the above takes $\mathcal{O}(n|E|)$ time where $|E| = \max(E_1, E_2)$. Taking into account computing and sorting $\mu_E(N_i)$, computing d_{μ_E} takes $O\big(|E|(n + \log|E|)\big)$ time.

This complexity can also be expressed in terms of the number of leaves and root components, thanks to the following straightforward generalization of Proposition 1 in [6], allowing for multiple root components.

Proposition 24. *Let N be a tree-child $\mathcal{L}$-network with n leaves and t root components. Then $|V_H| \leq n - t$.*
A node v is called elementary *if it is a tree node and either $\deg_o(v) = \deg(v) = 1$, or $\deg_o(v) < \deg(v) = 2$. If N has no elementary nodes then*

$$|V| \leq 2n - t + \sum_{v \in V_H} \deg_i(v) \leq (m+2)(n-t) + t$$

where $m = \max_{v \in V_H}\{\deg_i(v)\}$, and $|E| \leq (2m+1)(n-t)$.

Proof. Since N is an $\mathcal{L}$-network, it has no ambiguous leaves, and the elementary nodes in N are the tree nodes of out-degree 1 in any rooted partner. By considering a rooted partner, we may assume that N is a DAG with t roots, and follow the proof of Proposition 1 in [6]. Their arguments remain valid for the bounds on $|V_H|$ and $|V|$ when N has $t \geq 1$ roots, and when the removal of all but 1 parent hybrid edges at each hybrid node gives a forest instead of a tree. To bound $|E|$ we enumerate the parent edges of each node: $|E| \leq (|V_T| - t) + m|V_H| = |V| - t + (m-1)|V_H|$ then use the previous bounds. □

Therefore, as long as m and t are bounded and there are no elementary nodes, for example in binary tree-child networks with a single root component, then $|E| = \mathcal{O}(n)$. Consequently, computing μ_E on one such network or computing d_{μ_E} on two such networks takes $\mathcal{O}(n^2)$ time.

VI. Conclusion and extensions

For rooted networks, the node-based representation μ_V, or equivalently the ancestral profile, is known to provide a distance between networks beyond the class of tree-child networks, such as the class of semibinary tree-sibling time-consistent networks [7] and stack-free orchard binary networks [5], a class that includes binary tree-child networks. Orchard networks can be characterized as rooted trees with additional "horizontal arcs" [17]. They were first defined as cherry-picking networks: networks that can be reduced to a single edge by iteratively reducing a cherry or a reticulated cherry [18]. A *cherry* is a pair of leaves (x, y) with a common parent. A *reticulated cherry* is a pair of leaves (x, y) such that the parent u of y is a tree node and the parent v of x is a hybrid node with $e = (u, v)$ as a parent hybrid edge. Reducing the pair $C = (x, y)$ means removing taxon x if C is a cherry or removing hybrid edge e if C is a reticulated cherry, and subsequently suppressing u and v if they are of degree 2. Cherries and reticulated cherries are both well-defined on the class of semidirected networks considered here, because leaves are well-defined (stable across rooted partners), each leaf is incident to a single tree edge, and hybrid nodes / edges are well-defined. A *stack* is a pair of hybrid nodes connected by a hybrid edge, and a rooted network is *stack-free* if it has no stack. As hybrid edges are well-defined on our general class of networks, the concepts of stacks and stack-free networks also generalize directly. Therefore, we conjecture that for semidirected networks, our edge-based representation μ_E and the associated dissimilarity d_{μ_E} also separate distinct networks well beyond the tree-child class, possibly to stack-free orchard semidirected networks.

To discriminate distinct orchard networks with possible stacks, Cardona et al. [8] introduced an "extended" node-based μ-representation of rooted phylogenetic networks. In this representation, the μ-vector for each node v is extended by one more coordinate, $\mu_0(v)$, counting the number of paths from v to a hybrid node (any hybrid node). On rooted networks, adding this extension allows μ_V to distinguish between any two orchard networks, even if they contain stacks (but assumed binary, without parallel edges and without outdegree-1 tree nodes in [8]). For semidirected networks, we conjecture that the edge-based representation μ_E can also be extended in the same way, and that this extension may provide a proper distance on the space of semidirected orchard networks.

Phylogenetic networks are most often used as metric networks with edge lengths and inheritance probabilities. Dissimilarities are needed to compare metric networks using both their topologies and edge parameters. For trees, extensions of the RF distance, which d_{μ_E} extends, are widely used. They can be expressed using edge-based μ-vectors as

$$d(T_1, T_2) = \sum_{m \in \mu_E(T_1) \cup \mu_E(T_2)} |\ell(m, T_1) - \ell(m, T_2)|^p \quad (1)$$

where $\ell(m, T_i)$ is the length in tree T_i of the edge corresponding to the μ-vector m, considered to be 0 if m is absent from $\mu_E(T_i)$. The weighted RF distance uses $p = 1$ [27] and the branch score distance uses $p = 2$ [21]. If all weights $\ell(m, T)$ are 1 for $m \in \mu_E(T)$, then (1) boils down to the RF distance when restricted to trees (either rooted or unrooted), and to our d_{μ_E} dissimilarity on semidirected phylogenetic networks more generally. For networks with edge lengths, (1) could be used to extend d_{μ_E}, where $\ell(\mu_E(e), N)$ is defined as the length of edge e in N as it is for trees. A root μ-vector could be assigned weight $\ell(\mu_r(T), N) = 0$, because in standard cases, such as for networks with a single root component, the root μ-vector(s) carry redundant information.

Alternatively, using inheritance probabilities could be useful to capture the similarity between a network having a hybrid edge with inheritance very close to 0 and a network lacking this edge. To this end, we could modify μ-vectors. Recall that [6] defined $\mu(v, N) = (\mu_1, \ldots, \mu_n)$ with μ_i equal to the number m_i of paths from v to taxon i in a directed network N. We could generalize μ_i to be a function of these m_i paths, possibly reflecting inheritance probabilities. For example, we could use the weight of a path p, defined as $\gamma(p) = \prod_{e \in p} \gamma(e)$. These weights sum to 1 over up-down paths between v and i [33], although not over the m_i directed paths from v to

i. The weights of the m_i paths could then be normalized before calculating their entropy $H_i = -\sum_{p:v\rightsquigarrow i} \gamma(p)\log\gamma(p)$ and then define $\mu_i = e^{H_i}$. The original definition $\mu_i = m_i$ corresponds to giving all paths $v \rightsquigarrow i$ equal weight $1/m_i$. This extension carries over from directed to semidirected networks because we proved here that the set of directed paths from an edge $e = (u, v)$ to i is independent of the root choice, given a fixed admissible direction assigned to e, as shown in Propositions 15 and 17. With this extension, μ-vectors are in the continuous space $\mathbb{R}^n_{\geq 0}$ instead of $\mathbb{Z}^n_{\geq 0}$. To use them in a dissimilarity between networks N and N', we could use non-trivial distance between μ-vectors (such as the L^1 or L^2 norm) then get the score of an optimal matching between μ-vectors in $\mu_E(N)$ and $\mu_E(N')$. Searching for an optimal matching would increase the computational complexity of the dissimilarity, but would remain polynomial using the Hungarian algorithm [20].

To reduce the dependence of d_{μ_E} on the number of taxa n in the two networks, d_{μ_E} should be normalized by a factor depending on n only. This is particularly useful to compare networks with different leaf labels, by taking the dissimilarity between the subnetworks on their shared leaves. Ideally, the normalization factor is the diameter of the network space, that is, the maximum distance $d_{\mu_E}(N, N')$ over all networks N and N' in a subspace of interest. For the subspace of unrooted trees on n leaves, this is $2(n-3)$ [31]. Future work could study the diameter of other semidirected network spaces, such as level-1 or tree-child semidirected networks (which have $n-t$ or fewer hybrid nodes where t is the number of root components, by Proposition 24) or orchard semidirected networks (whose number of hybrids is unbounded).

To compare semidirected networks N_1 on leaf set $\mathcal{L}_1$ and N_2 on leaf set $\mathcal{L}_2$ with a non-zero dissimilarity if $\mathcal{L}_1 \neq \mathcal{L}_2$, one idea is to consider the subnetworks $\tilde{N}_1$ and $\tilde{N}_2$ on their common leaf set $\mathcal{L} = \mathcal{L}_1 \cap \mathcal{L}_2$ then use a penalized dissimilarity:

$$d_{\mu_E}(\tilde{N}_1, \tilde{N}_2) + \lambda d_{\mathrm{Symm}}(\mathcal{L}_1, \mathcal{L}_2)$$

for some constant $\lambda \geq 0$. This dissimilarity may not satisfy the triangle inequality, which might be acceptable in some contexts. For example, consider as input a set of semidirected networks $N_1, \ldots, N_n$ with N_i on leaf set $\mathcal{L}_i$, and consider the full leaf set $\mathcal{L} = \cup_i^n \mathcal{L}_i$. We may then seek an $\mathcal{L}$-network N that minimizes some criterion, such as

$$\sum_{i=1}^{n} d(N, N_i)\ . \tag{2}$$

When N is constrained to be an unrooted tree, input networks N_i are unrooted trees and when d is the RF distance using N pruned to $\mathcal{L}_i$, this is the well-studied RF supertree problem [32]. When the input trees N_i are further restricted to be on 4 taxa, (2) is the criterion used by ASTRAL [34]. The very wide use of ASTRAL and its high accuracy points to the impact of distances that are fast to calculate, such as our proposed d_{μ_E}.

Acknowledgements

This work was supported in part by the National Science Foundation (DMS 2023239) and by a H. I. Romnes faculty fellowship to C.A. provided by the University of Wisconsin-Madison Office of the Vice Chancellor for Research with funding from the Wisconsin Alumni Research Foundation.

References

[1] Elizabeth S. Allman, Hector Baños, and John A. Rhodes. "NANUQ: a method for inferring species networks from gene trees under the coalescent model". In: *Algorithms for Molecular Biology* 14.1 (2019). DOI: 10.1186/s13015-019-0159-2.

[2] Cécile Ané et al. "Anomalous networks under the multispecies coalescent: theory and prevalence". In: *Journal of Mathematical Biology* 88 (2024), p. 29. DOI: 10.1007/s00285-024-02050-7.

[3] Hector Baños. "Identifying Species Network Features from Gene Tree Quartets Under the Coalescent Model". In: *Bulletin of Mathematical Biology* 81.2 (2019), pp. 494–534. DOI: 10.1007/s11538-018-0485-4.

[4] László Babai. "Graph Isomorphism in Quasipolynomial Time". In: *arXiv* (2016). DOI: 10.48550/arXiv.1512.03547.

[5] Allan Bai et al. "Defining phylogenetic networks using ancestral profiles". In: *Mathematical Biosciences* 332 (2021), p. 108537. DOI: 10.1016/j.mbs.2021.108537.

[6] Gabriel Cardona, Francesc Rosselló, and Gabriel Valiente. "Comparison of Tree-Child Phylogenetic Networks". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 6.4 (2009), pp. 552–569. DOI: 10.1109/tcbb.2007.70270.

[7] Gabriel Cardona et al. "A distance metric for a class of tree-sibling phylogenetic networks". In: *Bioinformatics* 24.13 (2008), pp. 1481–1488. DOI: 10.1093/bioinformatics/btn231.

[8] Gabriel Cardona et al. "Comparison of orchard networks using their extended μ-representation". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 21.3 (2024), pp. 501–507. DOI: 10.1109/TCBB.2024.3361390.

[9] Gabriel Cardona et al. "The Comparison of Tree-Sibling Time Consistent Phylogenetic Networks Is Graph Isomorphism-Complete". In: *The Scientific World Journal* 2014 (2014), p. 254279. DOI: 10.1155/2014/254279.

[10] Joseph Felsenstein. *Inferring Phylogenies*. Sunderland, Massachusetts: Sinauer Associates, 2004. ISBN: 0878931775. DOI: 10.1086/383584.

[11] Mathieu Gautier et al. "f-statistics estimation and admixture graph construction with Pool-Seq or allele count data using the R package poolfstat". In: *Molecular Ecology Resources* 22.4 (2022), pp. 1394–1416. DOI: 10.1111/1755-0998.13557.

[12] Elizabeth Gross et al. "Distinguishing level-1 phylogenetic networks on the basis of data generated by Markov processes". In: *Journal of Mathematical Biology* 83.3 (2021), p. 32. DOI: 10.1007/s00285-021-01653-8.

[13] Glenn Hickey et al. "SPR Distance Computation for Unrooted Trees". In: *Evolutionary Bioinformatics* 4 (2008), EBO.S419. DOI: 10.4137/EBO.S419.

[14] Katharina T. Huber, Vincent Moulton, and Guillaume E. Scholz. "Forest-Based Networks". In: *Bulletin of Mathematical Biology* 84 (2022), p. 119. DOI: 10.1007/s11538-022-01081-9.

[15] Katharina T. Huber et al. "Is this network proper forest-based?" In: *Information Processing Letters* 187 (2025), p. 106500. DOI: 10.1016/j.ipl.2024.106500.

[16] Daniel H. Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic Networks: Concepts, Algorithms and Applications*. Cambridge: Cambridge University Press, 2010. DOI: 10.1017/CBO9780511974076.

[17] Leo van Iersel et al. "Orchard Networks are Trees with Additional Horizontal Arcs". In: *Bulletin of Mathematical Biology* 84 (2022), p. 76. DOI: 10.1007/s11538-022-01037-z.

[18] Remie Janssen and Yukihiro Murakami. "On cherry-picking and network containment". In: *Theoretical Computer Science* 856 (2021), pp. 121–150. DOI: 10.1016/j.tcs.2020.12.031.

[19] Sungsik Kong, David L. Swofford, and Laura S. Kubatko. "Inference of Phylogenetic Networks from Sequence Data using Composite Likelihood". In: *bioRxiv* (2022). DOI: 10.1101/2022.11.14.516468.

[20] Harold W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97. DOI: 10.1002/nav.3800020109.

[21] M K Kuhner and J Felsenstein. "A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates". In: *Molecular Biology and Evolution* 11.3 (1994), pp. 459–468. DOI: 10.1093/oxfordjournals.molbev.a040126.

[22] Simone Linz and Kristina Wicke. "Exploring spaces of semi-directed level-1 networks". In: *Journal of Mathematical Biology* 87.70 (2023). DOI: 10.1007/s00285-023-02004-5.

[23] Sarah Lutteropp et al. "NetRAX: accurate and fast maximum likelihood phylogenetic network inference". In: *Bioinformatics* 38.15 (2022), pp. 3725–3733. DOI: 10.1093/bioinformatics/btac396.

[24] Robert Maier et al. "On the limits of fitting complex models of population history to f-statistics". In: *eLife* 12 (2023), e85492. DOI: 10.7554/eLife.85492.

[25] Samuel Martin, Vincent Moulton, and Richard M. Leggett. "Algebraic Invariants for Inferring 4-leaf Semi-directed Phylogenetic networks". In: *bioRxiv* (2023). DOI: 10.1101/2023.09.11.557152.

[26] Nico Neureiter et al. "Detecting contact in language trees: a Bayesian phylogenetic model with horizontal transfer". In: *Humanities and Social Sciences Communications* 9.1 (2022), p. 205. DOI: 10.1057/s41599-022-01211-7.

[27] D. F. Robinson and L. R. Foulds. "Comparison of weighted labelled trees". In: *Combinatorial Mathematics VI*. Ed. by A. F. Horadam and W. D. Wallis. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979, pp. 119–126. ISBN: 978-3-540-34857-3.

[28] D.F. Robinson and L.R. Foulds. "Comparison of phylogenetic trees". In: *Mathematical Biosciences* 53.1 (1981), pp. 131–147. DOI: doi.org/10.1016/0025-5564(81)90043-2.

[29] Claudia Solís-Lemus and Cécile Ané. "Inferring Phylogenetic Networks with Maximum Pseudolikelihood under Incomplete Lineage Sorting". In: *PLOS Genetics* 12.3 (2016), e1005896. DOI: 10.1371/journal.pgen.1005896.

[30] Samuele Soraggi and Carsten Wiuf. "General theory for stochastic admixture graphs and F-statistics". In: *Theoretical Population Biology* 125 (2019), pp. 56–66. DOI: 10.1016/j.tpb.2018.12.002.

[31] Mike Steel. *Phylogeny: Discrete and Random Processes in Evolution*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2016, p. 302. DOI: 10.1137/1.9781611974485.

[32] Pranjal Vachaspati and Tandy Warnow. "FastRFS: fast and accurate Robinson-Foulds Supertrees using constrained exact optimization". In: *Bioinformatics* 33.5 (2017), pp. 631–639. DOI: 10.1093/bioinformatics/btw600.

[33] Jingcheng Xu and Cécile Ané. "Identifiability of local and global features of phylogenetic networks from average distances". In: *Journal of Mathematical Biology* 86.1 (2023), p. 12. DOI: 10.1007/s00285-022-01847-8.

[34] Chao Zhang et al. "ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees". In: *BMC Bioinformatics* 19.Suppl 6 (2018), p. 153. DOI: 10.1186/s12859-018-2129-y.

Michael Maxfield received his Bachelor of Science degrees in Computer Science and Mathematics from the University of Wisconsin - Madison in 2024, and is currently working towards his Masters degree in Mathematics. His interests are in various areas of mathematics and logic.

Jingcheng Xu received his Ph.D. degree in Statistics from the University of Wisconsin-Madison in 2024, focusing on distance-based methods for phylogenetic networks. He currently works in a research role in the finance sector.

Cécile Ané is currently Professor at the University of Wisconsin - Madison. Her research interests are in the development of statistical and computational methods for the study of molecular and trait evolution.