Algorithmica
DOI 10.1007/s00453-016-0241-9

Beyond Representing Orthology Relations by Trees

K. T. Huber[1] · G. E. Scholz[1]

Received: 18 February 2016 / Accepted: 2 November 2016

Abstract Reconstructing the evolutionary past of a family of genes is an important aspect of many genomic studies. To help with this, simple relations on a set of sequences called orthology relations may be employed. In addition to being interesting from a practical point of view they are also attractive from a theoretical perspective in that e. g. a characterization is known for when such a relation is representable by a certain type of phylogenetic tree. For an orthology relation inferred from real biological data it is however generally too much to hope for that it satisfies that characterization. Rather than trying to correct the data in some way or another which has its own drawbacks, as an alternative, we propose to represent an orthology relation δ in terms of a structure more general than a phylogenetic tree called a phylogenetic network. To compute such a network in the form of a level-1 representation for δ, we formalize an orthology relation in terms of the novel concept of a symbolic 3-dissimilarity which is motivated by the biological concept of a "cluster of orthologous groups", or COG for short. For such maps which assign symbols rather that real values to elements, we introduce the novel NETWORK- POPPING algorithm which has several attractive properties. In addition, we characterize an orthology relation δ on some set X that has a level-1 representation in terms of eight natural properties for δ as well as in terms of level-1 representations of orthology relations on certain subsets of X.

Keywords Orthology relation · Phylogenetic network · Level-1 network · Trinet · 3-Dissimilarity · COG

✉ K. T. Huber
k.huber@uea.ac.uk

G. E. Scholz
gllm.scholz@gmail.com

[1] School of Computing Sciences, University of East Anglia, Norwich, UK

Published online: 09 November 2016

1 Introduction

Unraveling the evolutionary past of a family $\mathscr{G}$ of genes is an important aspect for many genomic studies. For this, it is generally assumed that the genes in $\mathscr{G}$ are orthologs, that is, have arisen from a common ancestor through speciation. However it is known that shared ancestry of genes can also arise via, for example, whole genome duplication which gives rise to paralogs. This potentially obscures the signal used for reconstructing the evolutionary past of the genes in $\mathscr{G}$ in the form of a gene tree (essentially a rooted tree whose leaves are labelled by the elements of $\mathscr{G}$—we present precise definitions of the main concepts used in the next section) [28]. To tackle this problem, tree-based approaches have been proposed such as the ones underpinning the parsimony based NOTUNG [4] (see also [25]), ecceTERA [14] and RANGER- DTL [2] softwares, and the maximum likelihood and Bayesian-based approaches introduced in [8] and [20], respectively. Typically, these work by reconciling a gene tree with an assumed further tree called a species tree in terms of a map that operates on their vertex sets. For this, certain evolutionary events are postulated such as the ones mentioned above (see e. g. [21] for a recent review as well as [17] and the references therein).

Although undoubtedly highly attractive, one of the main drawbacks of tree-based approaches is the dependence of the resulting reconciliation on the quality of the employed trees which is not always guaranteed. Furthermore, these types of approaches can be computationally demanding for datasets generated by modern sequencing technology which might contain hundreds of thousands of sequences (see e. g. [23,29] for more on this). To overcome this problem, orthology relations (essentially maps whose range is a set of symbols representing evolutionary events of interest) have been proposed as an alternative. Relying on, for example, some notion of sequence similarity [26], gene order [16,19], or annotation of genes based on their function [5]—see e.g. [18] for more on this—these operate directly on the set of sequences from which a gene tree is built by applying some sort of clustering [1]. In addition to having attractive practical properties such as providing a way forward in cases where no species tree is available for a data set of interest, such relations are also interesting from a theoretical point of view due to their relationship with e. g. co-trees [9,10]. Furthermore, a characterization is known for when an orthology relation can be represented in terms of a certain type of phylogenetic tree [9].

Due to e. g. errors, noise, or indeed true signal in an orthology relation, it is however in general too much to hope for that an orthology relation obtained from a real biological dataset satisfies that characterization. A natural strategy might therefore be to try and correct for this in some way. Yet even if an underlying tree-like evolutionary scenario is assumed for this, many natural formalizations of how this could be achieved lead to NP-complete problems [18]. Furthermore, true non-treelike evolutionary signal might be overlooked. As an alternative, we propose to represent orthology relations in terms of phylogenetic networks (as opposed to phylogenetic trees). These are essentially rooted, directed, acyclic graphs which generalize phylogenetic trees by permitting additional edges. To infer such a structure from an orthology relation δ, we introduce the novel NETWORK- POPPING algorithm which returns a representation of δ in the form of a level-1 representation, that is, a level-1 (phylogenetic) network some of whose interior vertices are labelled in terms of evolutionary eveents. Such net-

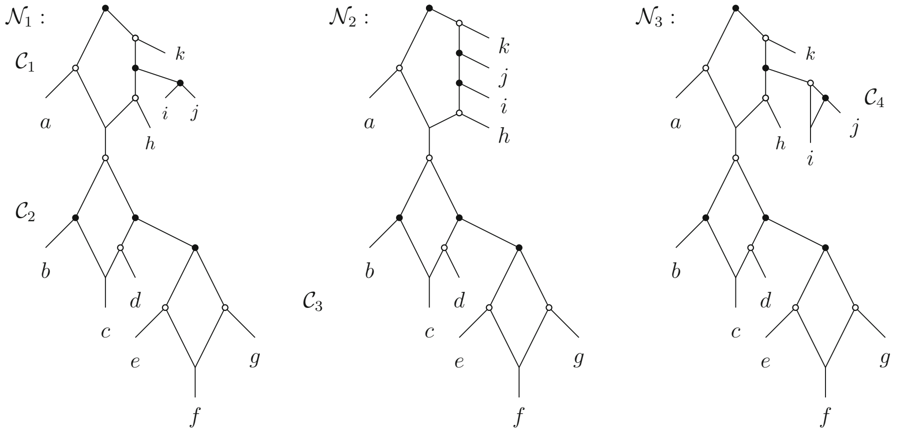

Fig. 1 Three distinct level-1 representations of the symbolic 3-dissimilarity $\delta_{\mathcal{N}_2}$ with leaf set $X = \{a, \ldots, k\}$ induced by $\mathcal{N}_2$. In all three cases the underlying phylogenetic network is a level-1 network. However, only $\mathcal{N}_1$ is returned by NETWORK- POPPING when given $\delta_{\mathcal{N}_2}$. Furthermore, $\mathcal{N}_2$, is not semi-discriminating but weakly labelled whereas $\mathcal{N}_3$ is semi-discriminating but not weakly labelled—see text for details

works have a set of organisms of interest as their leaf set and are characterized by the requirement that no two cycles (ignoring the directions for the moment!) share a vertex. This simplicity makes them obvious choices for orthology relation representation if the amount of non-treelike signals in a data set is relatively small. At the same time, their complexity renders them an ideal starting point for methodology development to account for further evolutionary signals (either in terms of spurious or true signal) in orthology relations (see e. g. [21] for more on this). To illustrate these concepts, we present in Fig. 1 three distinct level-1 representations of an orthology relation. In each representation, the interior vertices labelled in terms of $\bullet$ and $\circ$ represent two distinct evolutionary events such as speciation and whole genome duplication. The unlabelled interior vertices indicate non-treelike evolutionary signals such as the ones mentioned above.

Note that in cases where the underlying level-1 network N of a level-1 representation $\mathcal{N}$ is in fact a phylogenetic tree in the usual sense (see e. g. [24]), the orthology relation canonically induced on any two leaves by taking their lowest common ancestor is unique. In fact, if $\mathcal{N}$ is *discriminating*, that is, no two adjacent interior vertices in N have the same label, then $\mathcal{N}$ is uniquely determined by the orthology relation induced this way [3] and can be reconstructed using e. g. the BOTTOM- UP algorithm [9].

Intriguingly, the notion of a (unique) lowest common ancestor is also well-defined for general level-1 networks. This makes it tempting to speculate that similar kinds of ideas could also be made to work for such networks. As the example depicted in Fig. 2 illustrates, taking pairs of leaves as in the case of a phylogenetic tree can however be problematic from a reconstruction point of view as all three level-1 representations depicted in that figure represent the same orthology relation obtained that way.

As it turns out, the key to overcoming the resulting uniqueness problem is held by a point made in [6, Chapter 12] and [1]. Namely, that estimates on l-subsets, $l \geq 3$,

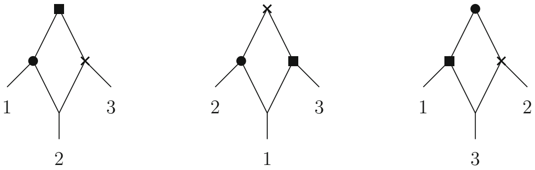

Fig. 2 Three distinct level-1 representations of the 2-dissimilarity $\delta : \binom{\{1,2,3\}}{2} \to M = \{\bullet, \times, \blacksquare\}$ defined by taking lowest common ancestors of pairs of leaves

of a set X are potentially more accurate than mere distances on X as they capture more information. Combined with ideas from, for example, [27] relating to *clusters of orthologous groups (COGs)* on how such estimates could be obtained, we formalize an orthology relation in terms of the novel concept of a symbolic 3-dissimilarity on X. Contrary to symbolic 2-dissimilarities used in [9] which operate on subsets of X of size at most two, such maps assign a symbolic value to any subset of X of size at most three.

As we shall see, Network- Popping takes as input a 3-dissimilarity on some set X and is guaranteed to find, in $\mathscr{O}(|X|^6)$-time, a level-1 representation for it if such a representation exists. For this, it relies on the three further algorithms below which we also introduce. It works by first finding for a symbolic 3-dissimilarity δ on X all pairs of subsets of X that support a cycle in a potential level-1 representation for δ using algorithm Find-Cycles. Subsequent to this, it employs algorithm Build-Cycles to construct from each such pair (H, R') a structurally very simple level-1 representation for the symbolic 3-dissimilarity induced on $H \cup R'$ where H and R' are as in the statement of algorithm Find-Cycles. Combined with algorithm Vertex-Growing which constructs a symbolic discriminating representation for a symbolic 2-dissimilarity (*i. e.* a symbolic distance), Network- Popping then recursively grows the level-1 representation for δ by repeatedly applying algorithms Build-Cycles and Vertex-Growing in concert. For the convenience of the reader, we illustrate all four algorithms by means of the level-1 representations depicted in Fig. 1. As part of our analysis of algorithm Network- Popping, we characterize level-1 representable symbolic 3-dissimilarities δ on X in terms of eight natural properties (P1)—(P8) enjoyed by δ (Theorem 2). Furthermore, we characterize such dissimilarities in terms of level-1 representable symbolic 3-dissimilarities on subsets of X of size $|X| - 1$ (Theorem 4). Within a divide-and-conquer framework the resulting speed-up of algorithm Network- Popping might allow it to also be applicable to large datasets.

The paper is organized as follows. In the next section, we present basic definitions and results. Subsequent to this, we introduce in Sect. 3 the crucial concept of a δ-trinet associated to a symbolic 3-dissimilarity and state Property (P1). In Sect. 4, we present algorithm Find- Cycles as well as Properties (P2) and (P3). In Sect. 5, we introduce and analyze algorithm Build- Cycles. Furthermore, we state Properties (P4)–(P6). In Sect. 6, we present algorithms Vertex- Growing and Network- Popping. As suggested by the example in Fig. 1, algorithm Network- Popping need not return the level-1 representation of a symbolic 3-dissimilarity that induced it. Employing a further algorithm called TRANSFORM, we address in Sect. 7 the associated uniqueness question (Corollary 2). As part of this we establish Theorem 2 which includes stating Properties (P7) and (P8). In Sect. 8, we establish Theorem 4. We conclude with Sect. 9 where we present research directions that might be worth pursuing.

2 Basic Definitions and Results

In this section, we collect relevant basic terminology and results concerning phylogenetic networks and symbolic 2- and 3-dissimilarities. From now on and unless stated otherwise, X denotes a finite set of size $n \geq 3$, M denotes a finite set of symbols of size at least two and $\odot$ denotes a symbol not already contained in M. Also, all directed/undirected graphs have no loops or multiple directed/undirected edges.

2.1 Directed Acyclic Graphs

Suppose G is a rooted directed acyclic graph (DAG), that is, a DAG with a unique vertex with indegree zero. We call that vertex the *root* of G, denoted by ρ_G. Also, we call the graph $U(G)$ obtained from G by ignoring the directions of its edges the *underlying graph* of G. By abuse of terminology, we call an induced subgraph H of G a *cycle* of G if the induced subgraph $U(H)$ of $U(G)$ is a cycle of $U(G)$. We call a vertex v of G an *interior vertex* of G if v is not a leaf of G where we say that a vertex v is a *leaf* if the indegree of v is one and its outdegree is zero. We denote the set of interior vertices of G by $V(G)_{int}$ and the set of leaves of G by $L(G)$. We call a vertex v of G a *tree vertex* if the indegree of v is at most one and its outdegree is at least two, and a *hybrid vertex* of G if the indegree of v is two and its outdegree is not zero. The set of interior vertices of G that are not hybrid vertices of G is denoted by $V(G)^{-}_{int}$. We say that N is *binary* if, with the exception of ρ_N, the indegree and outdegree of each of its interior vertices add up to three. Finally, we say that two DAG's N and N' with leaf set X are *isomorphic* if there exists a bijection from $V(N)$ to $V(N')$ that extends to a (directed) graph isomorphism between N and N' which is the identity on X.

2.2 Phylogenetic Networks and Last Common Ancestors

A *(rooted) phylogenetic network* N (*on* X) is a rooted DAG with leaf set X that does not contain a vertex that simultaneously has indegree and outdegree one. In the special case that a phylogenetic network N is such that each of its interior vertices belongs to at most one cycle we call N a *a level-1 (phylogenetic) network (on* X). Note that a phylogenetic network may contain cycles of length three and that a phylogenetic network that does not contain a cycle is called a *phylogenetic tree* T (*on* X).

For the following, let N denote a level-1 network on X. For $Y \subseteq X$ with $|Y| \geq 3$, we denote by $N|_Y$ the subDAG of N induced by Y (suppressing any resulting vertex that have indegree and outdegree one). Clearly, $N|_Y$ is a phylogenetic network on Y.

Suppose v is a non-leaf vertex of N. We say that a further vertex $w \in V(N)$ is *below* v if there is a directed path from v to w and call the set of leaves of N below v the *offspring set* of v, denoted by $\mathscr{F}(v)$. Note that $\mathscr{F}(v)$ is closely related to the hardwired cluster of N induced by v (see e.g. [13]). For a leaf $x \in \mathscr{F}(v)$, we refer to v as an *ancestor* of x. In case N is a phylogenetic tree, we define the *lowest common ancestor* $lca_N(x, y)$ of two distinct leaves $x, y \in L(N)$ to be the (necessarily unique) vertex $v \in V(N)$ such that $\{x, y\} \subseteq \mathscr{F}(v)$ and $\{x, y\} \not\subseteq \mathscr{F}(v')$ holds for all children $v' \in V(N)$ of v. More generally, for $Y \subseteq X$ with $2 \leq |Y| \leq |X|$, we denote by $lca_N(Y)$ the unique vertex v of N such that $Y \subseteq \mathscr{F}(v)$, and $Y \not\subseteq \mathscr{F}(v')$ holds for all children $v' \in V(N)$ of v. Note that in case the tree N we are referring to is clear from the context, we write $lca(Y)$ rather than $lca_N(Y)$.

It is easy to see that the notion of a lowest common ancestor is not well-defined for phylogenetic networks in general. However the situation changes in case the network in question is a level-1 network, as the following central result shows. Since its proof is straight-forward, we omit it.

Lemma 1 *Let N be a level-1 network on X and assume that $Y \subseteq X$ such that $|Y| \geq 2$. Then there exists a unique interior vertex $v_Y \in V(N)$ such that $Y \subseteq \mathscr{F}(v_Y)$ but $Y \not\subseteq \mathscr{F}(v')$, for all children $v' \in V(N)$ of v_Y. Furthermore, there exists two distinct elements $x, y \in Y$ such that $v_Y = lca(x, y)$.*

Continuing with the terminology of Lemma 1, we refer to v_Y as the *lowest common ancestor of Y in N*, denoted by $lca_N(Y)$. As in the case of a phylogenetic tree, we write $lca(Y)$ rather than $lca_N(Y)$ if the network N we are referring to is clear from the context.

2.3 Symbolic Dissimilarities and Labelled Level-1 Networks

Suppose $k \in \{2, 3\}$. We denote by $\binom{X}{k}$ the set of subsets of X of size k, and by $\binom{X}{\leq k}$ the set of nonempty subsets of X of size at most k. We call a map $\delta : \binom{X}{\leq k} \to M^{\odot} := M \cup \{\odot\}$ a *symbolic k-dissimilarity on X with values in $M^{\odot}$* if, for all $A \in \binom{X}{\leq k}$, we have that $\delta(A) = \odot$ if and only if $|A| = 1$. To improve clarity of exposition, we refer to δ as a *symbolic* 3-*dissimilarity on* X if the set M is of no relevance to the discussion. Moreover, for $Y = \{x_1, \ldots, x_l\}$, $l \geq 2$, we write $\delta(x_1, \ldots, x_l)$ rather than $\delta(Y)$ where the order of the elements x_i, $1 \leq i \leq l$, is of no relevance to the discussion.

A *labelled (phylogenetic) network* $\mathscr{N} = (N, t)$ (*on* X) is a pair consisting of a phylogenetic network N on X and a labelling map $t : V(N)^{-}_{int} \to M$. If N is a level-1 network then $\mathscr{N}$ is called a *labelled level-1 network* (see e.g. Fig. 3). To improve clarity of exposition we use calligraphic font to denote a labelled phylogenetic network.

Suppose $\mathscr{N} = (N, t)$ is a labelled level-1 network on X such that the vertices in $V(N)^{-}_{int}$ are labelled in terms of M. Then we denote by $\delta_{\mathscr{N}} : \binom{X}{\leq 3} \to M^{\odot}$ the symbolic 3-dissimilarity on X induced by $\mathscr{N}$ given by $\delta_{\mathscr{N}}(Y) = t(lca(Y))$ if $|Y| \neq 1$, and $\delta_{\mathscr{N}}(Y) = \odot$ otherwise. For $\mathscr{N}' = (N', t')$ a further labelled level-1 network on X, we say that $\mathscr{N}$ and $\mathscr{N}'$ are *isomorphic* if N and N' are isomorphic and $\delta_{\mathscr{N}} = \delta_{\mathscr{N}'}$.

Conversely, suppose δ is a symbolic 3-dissimilarity on X. In view of Lemma 1, we call a labelled level-1 network $\mathscr{N} = (N, t)$ on X a *level-1 representation* of δ if

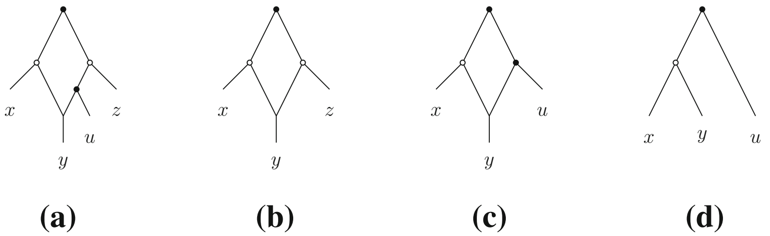

Fig. 3 **a** A labelled level-1 network $\mathcal{N}$ on $X = \{x, y, z, u\}$. **b**, **d** Semi-discriminating level-1 representations of $\delta_{\mathcal{N}}$ restricted to $\{x, y, z\}$ and $Y = \{u, x, y\}$, respectively. **c** A level-1 representation of $\delta_{\mathcal{N}}|_Y$ in the form of a labelled trinet that is is not a $\delta_{\mathcal{N}}$-trinet

$\delta = \delta_{\mathcal{N}}$. For ease of terminology, we sometimes say that δ is *level-1 representable* if the labelled network we are referring to is of no relevance to the discussion.

As is straight-forward to see, any labelled network $\mathcal{N} = (N, t)$ that contains a directed edge e both of whose endvertices have the same label induces the same symbolic 3-dissimilarity as the labelled network obtained from $\mathcal{N}$ by collapsing e. From a uniqueness point of view this is clearly undesirable. We therefore call a level-1 representation of δ *semi-discriminating* if N does not contain a directed edge (u, v) such that $t(u) = t(v)$ except for when there exists a cycle C of N with $|V(C) \cap \{u, v\}| = 1$. For example, all three labelled level-1 networks depicted in Fig. 1 are level-1 representations of $\delta_{\mathcal{N}_2}$ where $\mathcal{N}_2$ is the labelled level-1 network depicted in Fig. 1. Furthermore, the representations $\mathcal{N}_1$ and $\mathcal{N}_3$ of $\delta_{\mathcal{N}_2}$ presented in Fig. 1 are semi-discriminating whereas $\mathcal{N}_2$ is not as the parents of j and i belong to the same cycle, are joined by an edge, and have same label.

Note that in case N is a phylogenetic tree on X the definition of a semi-discriminating level-1 representation for δ reduces to that of a discriminating symbolic representation for the restriction $\delta_2 = \delta|_{\binom{X}{\leq 2}}$ of δ to $\binom{X}{\leq 2}$ (see [3] and also [9,24] for more on such representations, called *discriminating symbolic representations* in [24]). Using the concept of a *symbolic ultrametric*, that is, a symbolic 2-dissimilarity $\delta : \binom{X}{\leq 2} \to M^{\odot}$ for which, in addition, the following two properties are satisfied

(U1) $|\{\delta(x, y), \delta(x, z), \delta(y, z)\}| \leq 2$ for all $x, y, z \in X$;
(U2) there exists no four elements $x, y, z, u \in X$ such that

$$\delta(x, y) = \delta(y, z) = \delta(z, u) \neq \delta(z, x) = \delta(x, u) = \delta(u, y);$$

such representations were characterized by the authors of [3] as follows.

Theorem 1 ([3, Theorem 7.6.1]) *Suppose $\delta : \binom{X}{\leq 2} \to M^{\odot}$ is a symbolic 2-dissimilarity on X. Then there exists a discriminating symbolic representation of δ if and only if δ is a symbolic ultrametric.*

Clearly, it is too much to hope for that any symbolic 3-dissimilarity δ has a level-1 representation. The question therefore becomes: Which symbolic 3-dissimilarities have such a representation? A first partial answer is provided by Theorem 1 and Lemma 1 for not δ but its restriction δ_2. More precisely, δ has a discriminating symbolic

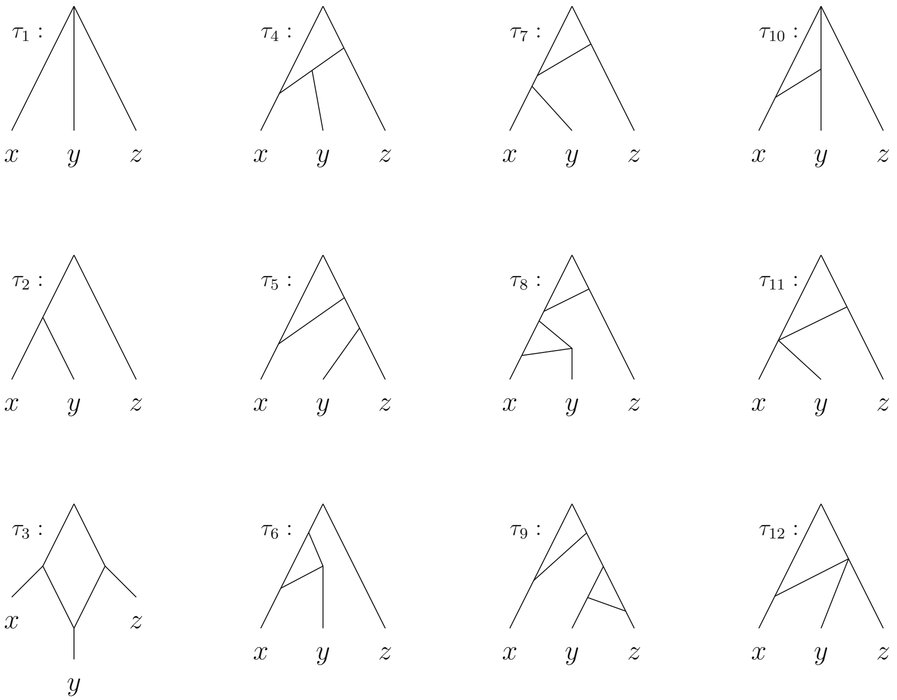

Fig. 4 The twelve trinets in the form of level-1 networks. The two omitted trinets from [11] are not level-1 networks in our sense

representation if and only if δ_2 is a symbolic ultrametric and, for all $x, y, z \in X$ distinct, $\delta(x, y, z)$ is the (unique) element appearing at least twice in the multiset $\{\delta_2(x, y), \delta_2(x, z), \delta_2(y, z)\}$.

3 δ-Triplets, δ-Tricycles, and δ-Forks

To make a first inroad into the aforementioned question, we next investigate structurally very simple level-1 representations of symbolic 3-dissimilarities. As we shall see, these turn out to be of fundamental importance for our algorithm Network- Popping (see Sect. 6) as well as for our analysis of its properties. In the context of this, it is important to note that although *triplets* (i. e. binary phylogenetic trees on 3 leaves) are well-known to uniquely determine (up to isomorphism) phylogenetic trees this does not hold for level-1 networks in general [7]. To overcome this problem, *trinets*, that is, phylogenetic networks on three leaves were introduced in [11]. For the convenience of the reader, we depict in Fig. 4 all 12 trinets $\tau_1, \ldots, \tau_{12}$ on $X = \{x, y, z\}$ from [11] that are also level-1 networks in our sense. In the same paper, it was observed that even the slightly more general 1-nested networks are uniquely determined by their induced trinet sets (see also [12] for more on constructing level-1 networks from trinets, and [30] for an extension of this result to other classes of phylogenetic networks).

Table 1 For $\delta : \binom{X}{\leq 3} \to M^{\odot}$ a symbolic 3-dissimilarity we list all labelled trinets on $X = \{x, y, z\}$ in terms of the size of E

$\|\{\delta(x, y), \delta(x, z), \delta(y, z)\}\|$	$\delta(x, y, z) = \ldots$	N
1	$\delta(x, y) = \delta(x, z) = \delta(y, z)$	Fork
3	$\delta(y, z)$	$x\|\|yz$
2	$\delta(y, z) \neq \delta(x, y) = \delta(x, z)$	$x\|\|yz$
2	$\delta(x, y) = \delta(x, z)$	$x\|yz$

Perhaps not surprisingly, trinets on their own are not strong enough to uniquely determine labelled level-1 networks in the sense that any two level-1 representations of a symbolic 3-dissimilarity must be isomorphic. To see this, suppose $|X| = 3$ and consider the symbolic 3-dissimilarity $\delta : \binom{X}{\leq 3} \to \{A, \odot\}$ that maps X and every 2-subset of X to A. Then the labelled network (τ_1, t) where t maps the unique vertex in $V(\tau_1)^{-}_{int}$ to A is a level-1 representation of δ and so is the labelled network (τ_4, t'), where every vertex in $V(\tau_4)^{-}_{int}$ is mapped to A by t'. Note that similar arguments may also be applied to the level-1 representations involving the trinets τ_4 to τ_{12} depicted in Fig. 4.

To be able to state the next result (Lemma 2), we say that a symbolic 3-dissimilarity δ satisfies the *Helly-type Property* if, for any three elements $x, y, z \in X$, we have $\delta(x, y, z) \in \{\delta(x, y), \delta(x, z), \delta(y, z)\}$. Note that we sometimes also refer to the Helly-type property as Property (P1).

Lemma 2 *Suppose δ is a symbolic 3-dissimilarity on a set $X = \{x, y, z\}$ taking values in $M^{\odot}$. Then there exists a level-1 representation $\mathcal{N}$ of δ if and only if δ satisfies the Helly-type Property. In that case $\mathcal{N}$ can be (uniquely) chosen to be semi-discriminating and, (up to permutation of the leaves of the underlying level-1 network N) N is isomorphic to one of the trinets τ_1, τ_2 and τ_3 depicted in Fig. 4.*

Proof Suppose first that $\mathcal{N} = (N, t)$ is a level-1 representation of δ. Then, in view of Lemma 1, $\delta(x, y, z) \in \{\delta(x, y), \delta(x, z), \delta(y, z)\}$ must hold.

Conversely, suppose that $\delta(x, y, z) \in E := \{\delta(x, y), \delta(x, z), \delta(y, z)\}$ holds. By analyzing the size of E it is straight-forward to show that one of the situations indicated in the rightmost column of Table 1 must apply. With defining a labelling map $t : V(N)^{-}_{int} \to M^{\odot}$ in the obvious way using the second column of that table, it follows that $\mathcal{N}$ is a level-1 representation for δ. □

Interestingly, all of trinets τ_1 through to τ_{12} can be labelled in such a way that line 1 in Table 1 is satisfied. Similarly, all of trinets τ_4 through to τ_{11} and τ_2 can be labelled so that line 4 holds and only τ_3 can be labelled so that lines 2 or 3 apply. Reflecting our assumption that the amount of non-treelike signals in a dataset is small, we evoke parsimony regarding the number of cycles for the former two cases and focus for the remainder of this paper on the trinets τ_1, τ_2 and τ_3. We shall refer to them as *fork* on $X = \{x, y, z\}$, *triplet* $z|xy$, and *tricycle* $y||xz$, respectively.

Armed with Lemma 2, we make the following central definition. Suppose that $|Y| = 3$, that δ is a symbolic 3-dissimilarity on Y, and that $\mathcal{N} = (N, t)$ is a semi-discriminating level-1 representation of δ. Then we call $\mathscr{N}$ a δ-*fork* if N is a fork on Y, a δ-*triplet* if N is a triplet on Y, and a δ-*tricycle* if N is a tricycle on Y. For ease of terminology, we collectively refer to all three of them as a δ-*trinet*. Note that as the example of the labelled trinet depicted in Fig. 3c shows, there exist trinets that are not δ-trinets. By abuse of terminology, we refer for a symbolic 3-dissimilarity δ on X and any 3-subset $Y \subseteq X$ to a $\delta|_Y$-trinet as a δ-trinet.

4 Recognizing Cycles: The Algorithm Find-Cycles

In this section, we introduce and analyze algorithm Find- Cycles (see Algorithm 1 for a pseudo-code version). Its purpose is to recognize cycles in a level-1 representation of a symbolic 3-dissimilarity δ if such a representation exists. As we shall see, this algorithm relies on Property (P1) and a certain graph $\mathscr{C}(\delta)$ that can be canonically associated to δ. Along the way, we also establish two further crucial properties enjoyed by a level-1 representable symbolic 3-dissimilarity.

We start with introducing further terminology. Suppose N is a level-1 network and C is a cycle of N. Then we denote by $r(C)$ the unique vertex in C for which both children are also contained in C and call it the *root* of C. In addition, we call the hybrid vertex of N contained in C the *hybrid* of C and denote it by $h(C)$. Furthermore, we denote the set of all elements of X below $r(C)$ by $R(C)$ and the set of all elements of X below $h(C)$ by $H(C)$. Clearly, $H(C) \subsetneq R(C)$. Moreover, for any leaf $x \in R(C) - H(C)$, we denote by $v_C(x)$ the last ancestor of x in C. Note that $v_C(x)$ is the parent of x if and only if x is incident with a vertex in C. Last-but-not-least, we call the vertex sets of the two edge-disjoint directed paths from $r(C)$ to $h(C)$ the *sides* of C. Denoting these two paths by P_1 and P_2, respectively, we say that two leaves x and y in $R(C) - H(C)$ *lie on the same side* of C if the vertices $v_C(x)$ and $v_C(y)$ are both interior vertices of P_1 or P_2, and that they *lie on different sides* if they are not. For example, for C the underlying cycle of the cycle $\mathscr{C}_2$ indicated in the labelled network $\mathscr{N}_1$ pictured in Fig. 1, we have $R(C) = \{b, \dots, g\}$ and $H(C) = \{c\}$. Furthermore, the sides of C are $\{r(C), v_C(b), h(C)\}$ and $\{r(C), v_C(d), v_C(e), h(C)\}$ and $d, \dots, g$ lie on one side of C whereas b and d lie on different sides of C.

Suggested by Property (U2), the following property is of interest to us where δ denotes again a symbolic 3-dissimilarity on X:

(P2) For all $x, y, z, u \in X$ distinct for which $\delta(x, y) = \delta(y, z) = \delta(z, u) \neq \delta(z, x) = \delta(x, u) = \delta(u, y)$ holds there exists exactly one subset $Y \subseteq \{x, y, z, u\}$ of size 3 such that a tricycle on Y underlies a level-1 representation of $\delta|_Y$.

As a first result, we obtain

Lemma 3 *Suppose δ is a level-1 representable symbolic 3-dissimilarity on X. Then δ satisfies the Helly-type Property as well as Property (P2).*

Proof Note first that Property (P1) is a straight-forward consequence of Lemma 1.

To see that Property (P2) holds, note first that since δ is level-1 representable there exists a labelled level-1 network (N, t) such that $\delta(Y) = t(lca(Y))$, for all subsets

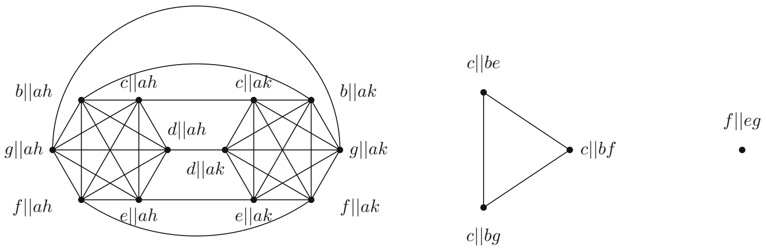

Fig. 5 The graph $\mathscr{C}(\delta_{\mathscr{N}_1})$, where $\mathscr{N}_1$ is the labelled level-1 network depicted in Fig. 1

$Y \subseteq X$ of size 2 or 3. Suppose $x, y, z, u \in X$ distinct are such that $\delta(x, y) = \delta(y, z) = \delta(z, u) \neq \delta(z, x) = \delta(x, u) = \delta(u, y)$. To see that there exists some $Y \subseteq Z := \{x, y, z, u\}$ for which $(N|_Y, t|_Y)$ is a δ-tricycle, assume for contradiction that there exists no such set Y. By Theorem 1, N cannot be a phylogenetic tree on X and, so, N must contain at least one cycle C. Without loss of generality, we may assume that $x \in H(C)$, and y lies on one of the two sides of C. By assumption $\delta(y, z) \neq \delta(x, z)$ and so either z and y lie on opposite sides of C, or z and y lie on the same side of C and $v_C(y)$ lies on the directed path from $r(C)$ to $v_C(z)$. As can be easily checked, either one of these two cases yields a contradiction since then $\delta(z, u) \neq \delta(x, u) = \delta(y, u)$ cannot hold for u, as required.

To see that there can exist at most one such tricycle on Z, assume for contradiction that there exist two tricycles τ and τ' with $L(\tau) \cup L(\tau') \subseteq Z$. Then $|L(\tau) \cap L(\tau')| = 2$. Choose $x, y \in L(\tau) \cap L(\tau')$. Note that the assumption on the elements of Z implies that x or y must be below the hybrid vertex of one of τ and τ' but not the other. Without loss of generality we may assume that y is below the hybrid vertex of τ but not below the hybrid vertex of τ'. Then y must lie on a side of the unique cycle C' of τ'. But this is impossible since the unique cycle of τ and C' are induced by the same cycle of N. □

We remark in passing that the proof of uniqueness in the proof of Lemma 3 combined with the structure of a level-1 network, readily implies the following result.

Lemma 4 *Suppose that δ is a symbolic 3-dissimilarity on X that is level-1 representable by a labelled network (N, t) and that $x, y, z \in X$ are three distinct elements such that $x||yz$ is a δ-tricycle. Let C denote the unique cycle in N such that $x \in H(C)$ and $y, z \in R(C) - H(C)$, and let $x' \in X$. If $x'||yz$ is a δ-tricycle then $x' \in H(C)$ and if $x||x'z$ is a δ-tricycle then $x' \in R(C)$ and x' and y lie on the same side of C.*

To better understand the structure of a symbolic 3-dissimilarity δ, we next associate to δ a graph $\mathscr{C}(\delta)$ defined as follows. The vertices of $\mathscr{C}(\delta)$ are the δ-tricycles and any two δ-tricycles τ and τ' are joined by an edge if $|L(\tau) \cap L(\tau')| = 2$. For example, consider the symbolic 3-dissimilarity $\delta_{\mathscr{N}_1}$ induced by the labelled level-1 network $\mathscr{N}_1$ pictured in Fig. 1. Then the graph presented in Fig. 5 is $\mathscr{C}(\delta_{\mathscr{N}_1})$.

The example in Fig. 5 suggests the following property for a symbolic 3-dissimilarity δ to be level-1 representable:

(P3) If τ and τ' are δ-tricycles contained in the same connected component of $\mathscr{C}(\delta)$, then

$$\delta(L(\tau)) = \delta(L(\tau')).$$

We collect first results concerning Property (P3) in the next proposition.

Proposition 1 *Suppose $\delta : \binom{X}{\leq 3} \rightarrow M^{\odot}$ is a symbolic 3-dissimilarity. If δ is level-1 representable or $|M| = 2$ holds then Property (P3) must hold. In particular, if $\mathscr{N}$ is a level-1 representation for δ then there exists a canonical injective map from the set of connected components of $\mathscr{C}(\delta)$ to the set of cycles of the level-1 network underlying $\mathscr{N}$.*

Proof Suppose first that δ is level-1 representable. Let $\mathscr{N} = (N, t)$ denote a level-1 representation of δ. Then $\delta = \delta_{\mathscr{N}}$. Since $\delta_{\mathscr{N}}(x, y, z) = t(r(C))$ holds for all cycles C of N, and any $x \in H(C)$ and any $y, z \in R(C)$ that lie on different sides of C, Property (P3) follows.

Suppose next that $|M| = 2$. It suffices to show that Property (P3) holds for any two adjacent vertices of $\mathscr{C}(\delta)$. Suppose τ and τ' are two such vertices and that $x, y, z \in X$ are such that $\tau = x||yz$. Then there exists some $u \in X$ such that either $\tau' = u||yz$ or $\tau' = x||ru$ where $r \in \{y, z\}$. Without loss of generality we may assume that $r = y$. In view of Table 1, we clearly have $\delta(x, y) \neq \delta(x, y, z) = \delta(y, z)$. Since, in addition, $\delta(u, y, z) = \delta(y, z)$ holds in the former case it follows that $\delta(L(\tau)) = \delta(L(\tau'))$. In the latter case, we obtain $\delta(x, y, u) \neq \delta(x, y)$ and thus, $\delta(L(\tau)) = \delta(L(\tau'))$ follows in this case too as $|M| = 2$.

The claimed injective map is a straight-forward consequence of Lemma 4. □

Algorithm Find-Cycles exploits the injection mentioned in Proposition 1 by interpreting for a symbolic 3-dissimilarity δ a connected component C of $\mathscr{C}(\delta)$ in terms of two sets H_C and R'_C. Note that if C' is a cycle in the level-1 network underlying a level-1 representation of δ (if such a representation exists!), the sets $H(C')$ and H_C coincide and $R'_C \subseteq R(C')$ holds.

For example, for the symbolic 3-dissimilarity $\delta_{\mathscr{N}_1}$ induced by the labelled network $\mathscr{N}_1$ depicted in Fig. 1, algorithm Find-Cycles returns the three pairs $(bcdefg, abcdefgk)$, $(c, bcefg)$ and (f, efg) where we write $x_1 \ldots x_{|A|}$ for a set $A = \{x_1, \ldots, x_{|A|}\}$.

5 Constructing Cycles: The Algorithm Build-Cycles

We next turn our attention toward reconstructing a structurally very simple level-1 representation of a symbolic 3-dissimilarity (should such a representation exist). For this, we use algorithm Build-Cycles which takes as input a symbolic 3-dissimilarity δ and a pair returned by Find-Cycles when given δ.

To state Build-Cycles, we require further terminology. Suppose N is a level-1 network. Then we say that N is *partially resolved* if all vertices in a cycle of N have degree three. Note that partially-resolved level-1 networks may have interior vertices

```
   Input: A symbolic 3-dissimilarity $\delta$ on $X$.
   Output: An integer $m \geq 0$ and $m$ pairs of subsets $(H_i, R'_i)$ of $X$, $1 \leq i \leq m$, or the statement "$\delta$ is
           not level-1 representable".
 1 if $\delta$ satisfies Property (P1) then
 2     Build the graph $\mathscr{C}(\delta)$;
 3     Denote by $m$ the number of connected components of $\mathscr{C}(\delta)$;
 4     for $i \in \{1, \dots, m\}$ do
 5         Let $K_i$ denote a connected component of $\mathscr{C}(\delta)$;
 6         set $H_i = \{x \in X :$ there exist $y, z \in X$ such that $x||yz$ is a vertex of $K_i\}$;
 7         set $R'_i = H_i \cup \{y \in X :$ there exist $x, z \in X$ such that $x||yz$ is a vertex of $K_i\}$;
 8     end
 9     return $m, (H_1, R'_1), \dots, (H_m, R'_m)$;
10 end
11 else
12     return $\delta$ is not level-1 representable;
13 end
```

Algorithm 1: Find-Cycles – Property (P1) is checked in Line 1.

not contained in a cycle that have degree greater than three. Thus such networks need not be binary. If, in addition to being partially resolved, N is such that it contains a unique cycle C such that every non-leaf vertex of N is a vertex of C then we call N *simple*.

Algorithm Build-Cycle (see Algorithm 2 for a pseudo-code version) relies on a further graph called the TopDown graph associated to a symbolic 3-dissimilarity δ. For (H, R') a pair returned by algorithm Find-Cycle when given δ and $x \in H$ and $S \subseteq R'$, that graph essentially orders the vertices of S. Thus, for each connected component K of $\mathscr{C}(\delta)$, Build-Cycle computes a level-1 representation of δ corresponding to K (should such a representation exist).

We start with presenting a central observation concerning labelled level-1 networks.

Lemma 5 *Suppose $\mathcal{N} = (N, t)$ is a labelled level-1 network, and C is a cycle of N. Suppose also that $x, y, z \in X$ are three elements such that $x \in H(C)$, $y, z \in R(C) - H(C)$ and $t(v_C(z)) = t(r(C)) \neq t(v_C(y))$. Then, $v_C(z)$ lies on the directed path from $v_C(y)$ to $h(C)$ if and only if $y|xz$ is a $\delta_{\mathcal{N}}$-triplet.*

Proof Put $\delta = \delta_{\mathcal{N}}$. Suppose first that $v_C(z)$ lies on the directed path from $v_C(y)$ to $h(C)$. Then $lca(x, y, z) = lca(x, y) = lca(y, z) = v_C(y)$ and $lca(x, z) = v_C(z)$. Hence, $\delta(x, y, z) = \delta(x, y) = \delta(y, z) = t(v_C(y)) \neq t(v_C(z)) = \delta(x, z)$. By Table 1, $y|xz$ is a δ-triplet.

Conversely, suppose that $y|xz$ is a δ-triplet. Then, by Table 1, we have $\delta(x, y, z) = \delta(x, y) = \delta(y, z) \neq \delta(x, z)$. Since $\delta(x, y) = t(v_C(y))$ and $\delta(x, z) = t(v_C(z))$, it follows that $\delta(x, y, z) = t(v_C(y)) \neq t(v_C(z))$. But then y and z must lie on the same side of C as otherwise $\delta(y, z) = t(r(C))$ follows which is impossible by assumption on x, y and z. Thus, either $v_C(y)$ must lie on a directed path P from $v_C(z)$ to $h(C)$ or $v_C(z)$ must lie on a directed path P' from $v_C(y)$ to $h(C)$. However $v_C(y)$ cannot be a vertex on P as otherwise $lca(y, z) = v_C(z)$ holds and, so, $\delta(y, z) = \delta(x, z)$ follows, which is impossible. Thus $v_C(z)$ must be a vertex on P'. □

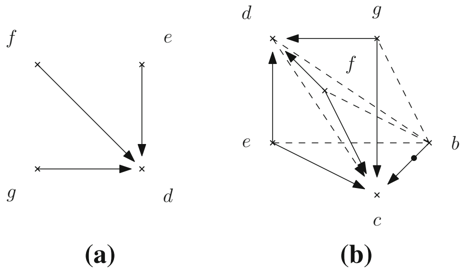

Fig. 6 For $\delta_{\mathscr{N}_1}$ the symbolic 3-dissimilarity induced by the labelled network $\mathscr{N}_1$ pictured in Fig. 1, we depict in (**a**) the TopDown graph $TD(\{d, e, f, g\}, c)$ and in (**b**) the CheckLabels graph $CL(\{c\}, \{b\}, \{d, e, f, g\})$ which we formally introduce in Sect. 7. In both graphs, the vertices are indicated by *times* symbol. In the latter graph the value assigned to two vertices under $\delta_{\mathscr{N}_1}$ is indicated in terms of *dashed* and *non-dashed* edges (ignoring directions for the moment). See text for details

With $\mathscr{N}$ and C as in from Lemma 5, it follows from Lemma 4, that whenever algorithm Find-Cycles is given $\delta_{\mathscr{N}}$ as input, it returns a pair (H, R') such that $H = H(C)$ and $R' = H(C) \cup \{y \in R(C) : t(v_C(y)) \neq t(r(C))\}$. Moreover giving (H, R') and $\delta_{\mathscr{N}}$ as input to algorithm Build-Cycle, Lemma 5 implies that Build-Cycle finds all elements $z \in R(C) - R'$ for which there exists some $y \in R'$ such that $v_C(z)$ lies on the path from $v_C(y)$ to $h(C)$. However it should be noted that if $z \in R(C) - H(C)$ is such that $t(v) = t(r(C)) = t(v_C(z))$ holds for all vertices v on the path from $r(C)$ to $v_C(z)$ then the information captured by $\delta_{\mathscr{N}}$ for x, y, and z is in general not sufficient to decide if z and y lie on the same side of C or not. In fact, it is easy to see that, in general, $z \in R(C)$ need not even hold.

We now turn our attention to the aforementioned TopDown graph associated to a symbolic 3-dissimilarity δ on X which is defined as follows. Suppose that $S \subsetneq X$, and that $x \in X - S$. Then the vertex set of the *TopDown graph* $TD(S, x)$ is S and two elements $u, v \in S$ distinct are joined by a direct edge (u, v) if $u|vx$ is a δ-triplet.

Rather than continuing with our analysis of algorithm Build- Cycle we break for the moment and illustrate it by means of an example. For this we return again to the symbolic 3-dissimilarity $\delta_{\mathscr{N}_1}$ on $X = \{a, \dots, k\}$ induced by the labelled level-1 network $\mathscr{N}_1$ depicted in Fig. 1. Suppose $(c, bcefg)$ is a pair returned by algorithm Find-Cycle and $c||be$ is the δ-tricycle chosen in line 2 of Build-Cycle. Then $H = \{c\}$, $S'_b = \{b\}$ and $S'_e = \{e, f, g\}$ (lines 3 and 4), and $S_b = \{b\}$ and $S_e = \{d, e, f, g\}$ (lines 8 and 9). The graph $TD(S_e, c)$ is depicted in Fig. 6a. It implies that for the cycle C associated to the pair $(c, bcefg)$ in a level-1 representation of $\delta_{\mathscr{N}_1}$, we must have $v_C(e) = v_C(f) = v_C(g)$ and that one of the two sides of C is $\{d, e, f, g\}$. Since $|S_b| = 1$, the other side of C is $\{b\}$ (lines 11 to 33).

Continuing with our analysis of algorithm Build- Cycle, we remark that the fact that the TopDown graph $TD(S_e, c)$ in the previous example is non-empty is not a coincidence. In fact, it is easy to see that the graph G defined in line 14 of Build-Cycle is non-empty whenever δ is level-1 representable. Thus, the DAG C returned by algorithm Build- Cycle cannot contain multi-arcs. Note however that there might be tricycles induced by C of the form $x||uz$ with $u \in R' - S'_y$ as, for example, $\delta(x, z) = \delta(x, y) = \delta(z, y) = \delta(x, u)$ might hold and thus $x||uz$ is not a δ-tricycle.

```
    Input: A symbolic 3-dissimilarity $\delta$ on $X$ that satisfies Property (P1) and a pair $(H, R')$ returned by
           algorithm FIND-CYCLE when given $\delta$.
    Output: Either a labelled simple level-1 network $(C, t)$ on a partition of a subset $X'$ of $X$ such that
            $R' \subseteq X'$ and $H(K) = H$ holds for the unique cycle $K$ of $C$, or the statement "$\delta$ is not
            level-1 representable".
 1  set $rep=0$;
 2  Choose a $\delta$-tricycle $x||yz$, where $x \in H$ and $y, z \in R' - H$;
 3  set $S'_y = \{u \in R' : x||uz \text{ is a } \delta\text{-tricycle}\}$;
 4  set $S'_z = \{u \in R' : x||yu \text{ is } \delta\text{-tricycle}\}$;
 5  Initialize $C$ as a graph with three vertices respectively labelled by $r(C)$, $h(C)$ and $H$, and the edge
    $(h(C), H)$;
 6  if for all $x' \in H$, $y' \in S'_y$ and $z' \in S'_z$, $x'||y'z'$ is a $\delta$-tricycle and $\delta(x, y, z) = \delta(x', y', z')$ then
 7      set $t(r(C)) = \delta(x, y, z)$;
 8      set $S_y = S'_y \cup \{u \in X - R' :$ there exists $u' \in S'_y$ such that $u'|ux$ is a $\delta$-triplet$\}$;
 9      set $S_z = S'_z \cup \{u \in X - R' :$ there exists $u' \in S'_z$ such that $u'|ux$ is a $\delta$-triplet$\}$;
10      if for all $u_1 \in S_y, u_2 \in S_z, \delta(u_1, u_2) = t(r(C))$ then
11          for $i \in \{y, z\}$ do
12              set $v_l = r(C)$;
13              if $TD(S_i, x') = TD(S_i, x'')$ for all $x', x'' \in H$ and $TD(S_i, x)$ does not contain a
                directed cycle then
14                  set $G = TD(S_i, x)$;
15                  set $rep=rep+1$;
16                  while $V(G) \neq \emptyset$ do
17                      Add a new child $v$ to $v_l$;
18                      set $\mathscr{F}(v) = \{u \in S_i : u \text{ has indegree } 0 \text{ in } G\}$;
19                      Delete from $G$ all vertices in $\mathscr{F}(v)$;
20                      if for all $u, u' \in \mathscr{F}(v)$, $x', x'' \in H \cup V(G)$, $\delta(u, x') = \delta(u', x'')$ then
21                          Choose some $u \in \mathscr{F}(v)$;
22                          set $t(v) = \delta(x, u)$;
23                          Add the leaf $\mathscr{F}(v)$ as a child of $v$;
24                          set $v_l = v$;
25                      end
26                      else
27                          Remove all vertices from $G$;
28                          set $rep=rep-1$;
29                      end
30                  end
31                  Add the edge $(v_l, h(C))$;
32              end
33          end
34      end
35  end
36  if $rep=2$ then
37      return $C$;
38  end
39  else
40      return $\delta$ is not level-1 representable;
41  end
```

Algorithm 2: BUILD-CYCLE – The set R' is the set $H \cup S_y \cup S_z$, Property (P4) is checked in Lines 6, 10, and 20, and Properties (P3), (P6), (P7) and (P8) are checked in Lines 6, 13, 10 and 20, respectively.– See text for details.

Note that similar reasoning also applies to S'_z and the extensions of S'_y and S'_z to S_y and S_z defined in lines 8 and 9, respectively. Also note that the sets S_y and S_z are dependent on the choice of the δ-tricycle in line 2. However, line 6 ensures that the labelled simple level-1 network returned by algorithm BUILD-CYCLE is independent of the choice of that δ-tricycle.

To establish Proposition 2 which ensures that algorithm BUILD- CYCLE terminates, we next associate to a directed graph G a new graph $P(G)$ by successively removing vertices of indegree zero and their incident edges until no such vertices remain. As a first almost trivial observation concerning that graph we have the following straightforward result whose proof we again omit.

Lemma 6 *Let G be a directed graph. Then $P(G)$ is nonempty if and only if G contains a directed cycle.*

Given as input to algorithm BUILD-CYCLE a symbolic 3-dissimilarity δ that satisfies Property (P1) and a pair (H, R') returned by algorithm FIND-CYCLE for δ we have:

Proposition 2 *Algorithm* BUILD-CYCLE *terminates.*

Proof As is easy to check the only reason for algorithm BUILD-CYCLE not to terminate is the while loop initiated in its line 16. For $i = 1, 2$, this while loop works by successively removing vertices of indegree 0 (and their incident edges) from the graph $TD(S_i, x)$, and terminates if the resulting graph, i. e. $P(TD(S_i, x))$, is empty. Since line 13 ensures that this loop is entered if and only if $TD(S_i, x)$ does not contain a directed cycle, Lemma 6 implies that BUILD-CYCLE terminates. □

It is straight-forward to see that when given a level-1 representable symbolic 3-dissimilarity δ such that the underlying level-1 network is in fact a simple level-1 network the labelled network returned by algorithm BUILD- CYCLE satisfies the following three additional properties (where we use the notations introduced in algorithm BUILD- CYCLE).

(P4) For $i = y, z$, we have $S'_i = \{u \in S_i : \delta(u, x) \neq \delta(y, z)\}$ and $S_y \cap S_z = S_y \cap H = S_z \cap H = \emptyset$.
(P5) For all $u, v \in R := H \cup S_y \cup S_z$ and all $w \in X - R$, we have $\delta(u, w) = \delta(v, w)$.
(P6) For all $u, u' \in H$ and $i \in \{y, z\}$, the graphs $TD(S_i, u)$ and $TD(S_i, u')$ are isomorphic and do not contain a directed cycle.

Since the quantities on which these properties are based also exist for general symbolic 3-dissimilarities we next study Properties (P4)–(P6) for such dissimilarities. As a first consequence of Property (P4) combined with Properties (P1) and (P2), we obtain a sufficient condition under which the TopDown graph $TD(S_i, x)$ considered in algorithm BUILD- CYCLE does not contain a directed cycle (lines 13). For convenience, we employ again the notation used in Algorithm 2.

Proposition 3 *Suppose that $\delta : \binom{X}{\leq 3} \to M^{\odot}$ is a symbolic 3-dissimilarity that satisfies Properties (P1), (P2) and (P4), that (H, R') is a pair returned by algorithm* FIND-CYCLES *when given δ, and that x, y and z are as specified as in line 2 of algorithm* BUILD- CYCLE. *Then the following hold for $i = y, z$.*

(i) If $TD(S_i, x)$ contains a directed cycle then it contains a directed cycle of size 3.
(ii) $TD(S_i, x)$ does not contain a directed cycle of length 3 whenever $|M| = 2$ holds.

Proof (i) By symmetry, it suffices to show the proposition for $i = y$. Suppose $TD(S_y, x)$ contains a directed cycle. Over all such cycles in $TD(S_y, x)$, choose a directed cycle C of minimal length. If $|V(C)| = 3$, then the statement clearly holds.

Suppose for contradiction for the remainder that $|V(C)| \geq 4$. Suppose $a, b, c, d \in V(C)$ are such that (a, b), (b, c), (c, d) are three directed edges in C. We next distinguish between the cases that $|V(C)| \geq 5$ and that $|V(C)| = 4$.

Suppose $|V(C)| \geq 5$. Then since $a, c \in S_y$, Lemma 4 combined with the minimality of C implies that we either have a δ-fork on $\{a, c, x\}$ or the δ-triplet $ac|x$. Hence, $\delta(x, a) = \delta(x, c)$ holds in either case. Note that similar arguments also imply that $\delta(x, b) = \delta(x, d)$. Since $|V(C)| \geq 5$, the directed edges (a, d) and (d, a) cannot be contained in $TD(S_y, x)$ and, using again similar arguments as before, $\delta(x, a) = \delta(x, d)$ must hold. In combination, we obtain $\delta(x, a) = \delta(x, b)$ which is impossible in view of (a, b) being an edge in $TD(S_y, x)$ and thus $\delta(x, a) \neq \delta(x, b)$.

Suppose $|V(C)| = 4$. By the minimality of C, neither (b, d) (d, b), (a, c) nor (c, a) can be a directed edge in $TD(S_y, x)$. Using similar arguments as in the previous case, it follows that $\delta(x, b) = \delta(x, d)$ and $\delta(x, a) = \delta(x, c)$. Combined with the facts that (a, b), (b, c), (c, d) are directed edges in C and that (d, a) must also be an edge in C as $|V(C)| = 4$, it follows that with $A := \delta(c, d)$ and $B := \delta(b, c)$ we have

$$A = \delta(x, c) = \delta(x, a) = \delta(a, b) \neq \delta(x, b) = \delta(x, d) = \delta(d, a) = \delta(b, c) = B. \tag{1}$$

Note that, $\delta(a, c) \in \{A, B\}$ must also hold as otherwise $|\{\delta(a, c), \delta(a, b), \delta(b, c)\}| = 3$ and so, in view of Table 1, $\delta|_{\{a,b,c\}}$ would be level-1 representable by a δ-tricycle on $\{a, b, c\}$. But then $H \cap \{a, b, c\} \neq \emptyset$ which is impossible in view of Property (P4). Similarly, one can show that $\delta(b, d) \in \{A, B\}$. By combining a case analysis as indicated in Table 1 with Eq. 1, it is straight-forward to see that each of the four detailed combinations of $\delta(a, c)$ and $\delta(b, d)$ in that table yields a contradiction in view of Property (P2).

(ii) By symmetry, it suffices to assume $i = y$. Let $|M| = 2$ and assume for contradiction that $TD(S_y, x)$ contains a directed cycle C of size 3. Let s, u, v denote the three vertices of C such that (s, u), (u, v) and (v, s) are the three directed edges of C. Then $\delta(u, x) \neq \delta(s, x) \neq \delta(v, s) = \delta(v, x) \neq \delta(u, v) = \delta(u, x)$ must hold. Since $|M| = 2$, this is impossible. □

6 Constructing Level-1 Representations From Symbolic 3-Dissimilarities: The Algorithm Network-Popping

In this section, we present algorithm Network-Popping which allows us to decide if a symbolic 3-dissimilarity is level-1 representable or not. If it is, then Network-Popping is guaranteed to find a level-1 representation in polynomial time.

Network-Popping takes as input a symbolic 3-dissimilarity δ on X and employs a top-down approach to recursively construct a semi-discriminating level-1 representation for δ (if such a representation exists). For l a leaf whose label set is of size at least two and constructed in one of the previous steps it essentially works by either replacing l with a labelled simple level-1 network or a labelled phylogenetic tree. To compute those networks algorithms FIND-CYCLE and BUILD-CYCLE are used, and to construct such trees algorithm VERTEX-GROWING is employed. At the heart of the latter lie Proposition 4 and algorithm BOTTOM-UP introduced in [9]. The latter takes as input a symbolic 2-dissimilarity δ satisfying Properties (U1) and (U2), and builds the unique discriminating symbolic representation $\mathscr{T}$ for δ (if it exists).

To be able to state algorithm VERTEX-GROWING, we require again further terminology. Following e. g. [24], we call a collection $\mathscr{H}$ of non-empty subsets of X a *hierarchy on* X if $A \cap B \in \{A, B, \emptyset\}$ holds for any two sets $A, B \in \mathscr{H}$. The proof of the following result is straight-forward and thus omitted.

Lemma 7 *Let N be a level-1 network with cycles $C_1, C_2, \ldots, C_k$, $k \geq 1$. Then, $\mathscr{H}_N = \{R(C_1), R(C_2), \ldots, R(C_k)\}$ is a hierarchy on X.*

Suppose $\mathscr{A}$ is a set of non-empty subsets of X. Then we define a relation $\sim_{(X,\mathscr{A})}$ on X by putting $x \sim_{(X,\mathscr{A})} y$ if there exists some $A \in \mathscr{A}$ such that $x, y \in A$, for all $x, y \in X$. Note first that $\sim_{(X,\mathscr{A})}$ is clearly an equivalence relation whenever $\mathscr{A}$ is a hierarchy. In addition, suppose that $\mathscr{A}$ is such that the partition X' of X induced by $\sim_{(X,\mathscr{A})}$ has size two or more. If $\delta : \binom{X}{\leq 3} \rightarrow M^{\odot}$ is a symbolic 3-dissimilarity such that for any two sets $Y, Y' \in X'$ we have $\delta(x, y) = \delta(x', y')$ for all $x, x' \in Y$ and $y, y' \in Y'$, then we associate to δ the map $\hat{\delta}$ given by

$$\begin{aligned} \hat{\delta} : \binom{X'}{\leq 2} &\rightarrow M^{\odot} \\ \{Y_1, Y_2\} &\mapsto \begin{cases} \odot & \text{if } Y_1 = Y_2, \\ \delta(y_1, y_2), \quad \text{where } y_1 \in Y_1, y_2 \in Y_2 & \text{otherwise.} \end{cases} \end{aligned}$$

Note that $\hat{\delta}$ is clearly well-defined and a symbolic 2-dissimilarity on X'. Associating to a level-1 representation $\mathscr{N} = (N, t)$ of δ the set $\mathscr{R} := \{R(C) : C \text{ is a cycle of } N\}$, we have the following result as an immediate consequence.

Proposition 4 *Suppose $\mathscr{N}$ is a labelled level-1 network on X and X' is the partition of X induced by the relation $\sim_{(X,\mathscr{R})}$ on X. If $|X'| \geq 2$ then $\delta_{\hat{\mathscr{N}}}$ is well defined and satisfies Properties (U1) and (U2). In particular, $\delta_{\hat{\mathscr{N}}}$ is a symbolic ultrametric on X'.*

Proof Put $\mathscr{N} = (N, t)$ and $\delta' = \delta_{\hat{\mathscr{N}}}$. Note first that for all $x, y \in X$, Lemma 7 implies that there exists some $R \in \mathscr{R}$ such that $x, y \in R$ if and only if there exists $R' \in \mathscr{R}' := \{R \in \mathscr{R} : R \text{ is set-inclusion maximal in } \mathscr{R}\}$ such that $x, y \in R'$. Let T_N denote the tree obtained from N by first collapsing for every cycle C of N with $R(C) \in \mathscr{R}'$ all vertices below or equal to $r(C)$ into a vertex and then labelling that vertex by $R(C)$. Put $t_N := t|_{V(T_N)}$. Then (T_N, t_N) is clearly a labelled phylogenetic tree on X'. Since $\mathscr{N}$ is a labelled level-1 network, it follows that (T_N, t_N) is a symbolic discriminating representation of $\delta_{\hat{\mathscr{N}}}$. In view of Theorem 1, the proposition follows. □

```
Input: A symbolic 3-dissimilarity $\delta$ on a set $X$, a subset $Y \subseteq X$, and a hierarchy $\mathscr{S}$ of proper
       subsets of $Y$.
Output: A discriminating symbolic representation on the partition of $Y$ induced by $\sim_{(Y,\mathscr{S})}$ or the
        statement "There exists no discriminating symbolic representation".
1 Let $Y'$ denote the partition of $Y$ induced by $\sim_{(Y,\mathscr{S})}$;
2 Apply the BOTTOM-UP algorithm to the symbolic ultrametric $\hat{\delta}$ induced by $\delta$ on $Y'$, as considered in
  Proposition 4;
3 if BOTTOM-UP returns a labelled tree $\mathscr{T}$ then
4     return $\mathscr{T}$;
5 end
6 else
7     return There exists no discriminating symbolic representation. ;
8 end
```

Algorithm 3: Vertex-Growing – Property (P2) is checked in Line 3.

To illustrate algorithm Vertex-Growing consider again the symbolic 3-dissimilarity $\delta_{\mathscr{N}_1}$ induced by the labelled level-1 network on $X = \{a \dots, k\}$ depicted in Fig. 1. Let $\mathscr{M}_1$, $\mathscr{M}_2$, and $\mathscr{M}_3$ denote the three labelled simple level-1 networks returned by algorithm Build-Cycle when given $\delta_{\mathscr{N}_1}$ such that $L(\mathscr{M}_1) = X$, $L(\mathscr{M}_2) = \{b, \dots, g\}$ and $L(\mathscr{M}_3) = \{e, f, g\}$. Then the partition of X found in line 1 of algorithm Vertex-Growing when given $\delta_{\mathscr{N}_1}$ and $\mathscr{R} = \bigcup_{i=1}^{3}\{L(\mathscr{M}_i)\}$ is X itself, since any two leaves of X are in relation with respect to $\sim_{(X,\mathscr{R})}$. Thus, the discriminating symbolic representation returned by Bottom-Up is a single leaf.

Armed with the algorithms Find- Cycles, Build- Cycles, and Vertex-Growing, we next present a pseudo-code version of algorithm Network-Popping (Algorithm 4).

To be able to establish in Proposition 6 that algorithm Network-Popping returns a semi-discriminating level-1 representation for a symbolic 3-dissimilarity (if such a representation exists), we require the following technical result.

Proposition 5 *Let δ be a symbolic 3-dissimilarity on X satisfying Property (P1), and assume that* Network-Popping *returns a labelled level-1 network $\mathscr{N}$ on X when given δ as input. Then the restrictions $\delta|_{\binom{X}{\leq 2}}$ and $\delta_{\mathscr{N}}|_{\binom{X}{\leq 2}}$ of δ and $\delta_{\mathscr{N}}$ to $\binom{X}{\leq 2}$, respectively, coincide if and only if δ and $\delta_{\mathscr{N}}$ coincide.*

Proof Put $\mathscr{N} = (N, t)$. Also, put $\delta' = \delta|_{\binom{X}{\leq 2}}$ and $\delta'_{\mathscr{N}} = \delta_{\mathscr{N}}|_{\binom{X}{\leq 2}}$. Clearly, if δ and $\delta_{\mathscr{N}}$ coincide then $\delta' = \delta'_{\mathscr{N}}$ must hold.

Conversely, assume that $\delta' = \delta'_{\mathscr{N}}$. Let $Z = \{a, b, c\} \in \binom{X}{3}$ and put $m = \delta(Z)$. Note that since $\mathscr{N}$ is clearly a level-1 representation of $\delta_{\mathscr{N}}$, Lemma 3 implies that $\delta_{\mathscr{N}}$ also satisfies Property (P1). Further note that, up to permuting the elements in Z, we either have (i) a δ-fork on Z, (ii) $a|bc$ is a δ-triplet, or (iii) $a||bc$ is a δ-tricycle.

If Case (i) holds then $\delta(a, b) = \delta(a, c) = \delta(b, c) = m$. Since, by assumption, $\delta(Y) = \delta_{\mathscr{N}}(Y)$ for all $Y \in \binom{X}{2}$, we also have $\delta_{\mathscr{N}}(a, b) = \delta_{\mathscr{N}}(a, c) = \delta_{\mathscr{N}}(b, c) = m$. Hence, $\delta_{\mathscr{N}}(Z) = m = \delta(Z)$ as δ satisfies Property (P1).

If Case (ii) holds then $m = \delta(a, b) = \delta(a, c) \neq \delta(b, c)$. Assume for contradiction that $\delta_{\mathscr{N}}(Z) \neq m$. Then, since $\delta_{\mathscr{N}}$ satisfies Property (P1) it follows that $\delta_{\mathscr{N}}(Z) =$

```
   Input: A symbolic 3-dissimilarity $\delta$ on $X$.
   Output: A semi-discriminating level-1 representation $\mathscr{N} = (N, t')$ of $\delta$, if such a representation
           exists, or the statement “$\delta$ is not level-1 representable”.
1  Initialize $N$ as an unique vertex $v$, labelled by $X$;
2  set $r = 1$;
3  Use FIND − CYCLES($\delta$) to obtain $m \geq 0$ pairs $(H_i, R'_i)$ of subsets $H_i$ and $R'_i$ of $X$, $1 \leq i \leq m$;
4  if for all $i \in \{1, \dots, m\}$, BUILD − CYCLE($\delta$; $H_i, R_i$) returns a labelled simple level-1 network
   $(C_i, t_i)$ as described in that algorithm then
5      put $R_i = R(C_i)$, and $\mathscr{R} = \{R_1, \dots, R_m\}$;
6      if for all $i \in \{1, \dots, m\}$, and all $y, z \in R_i$, and $x \notin R_i$, we have $\delta(x, y) = \delta(x, z)$ then
7          while there exists a leaf $l$ of $N$ whose label set $V_l \subseteq X$ has two or more elements AND $r \neq 0$
           do
8              if there exists $i \in \{1, \dots, m\}$ such that $V_l = R_i$ then
9                  identify $l$ with the root of the labelled simple level-1 network corresponding to $R_i$
                   and replace $N$ with the resulting labelled level-1 network;
10             end
11             else
12                 put $\mathscr{S}_l = \{R \in \mathscr{R} : R \subseteq V_l\}$;
13                 if VERTEX-POPPING($\delta$, $V_l$, $\mathscr{S}_l$) returns a discriminating symbolic representation
                   $\mathscr{T} = (T, t)$ then
14                     identify $l$ with the root of $T$ and replace $N$ with the resulting labelled level-1
                       network;
15                 end
16                 else
17                     set $r = 0$;
18                 end
19             end
20         end
21     end
22 end
23 if $r = 1$ AND $N$ is not $v$ then
24     return $\mathscr{N} := (N, t')$ where $t'$ is canonically obtained by combining the maps $t$ and $t_i$,
       $1 \leq i \leq m$;
25 end
26 else
27     return $\delta$ is not level-1 representable;
28 end
```

Algorithm 4: NETWORK-POPPING – Property (P5) is checked in Line 6.

$\delta_{\mathscr{N}}(b, c)$. By Table 1, $a||bc$ must be a $\delta_{\mathscr{N}}$-tricycle. Hence, there must exist a cycle C in N such that $a \in H(C)$, b and c are contained in $R(C)$ but lie on different sides of C, and $t(r(C)) = \delta_{\mathscr{N}}(Z)$. Since algorithm NETWORK-POPPING completes by returning $\mathscr{N}$ it follows that C is constructed in the while-loop starting in line 16 of algorithm BUILD- CYCLE. But then the condition in line 6 of BUILD-CYCLE has to be satisfied which implies that $t(r(C)) = \delta(Z)$ in view of line 7 of that algorithm. Hence, $m \neq \delta_{\mathscr{N}}(Z) = t(r(C)) = \delta(Z) = m$ which is impossible.

If Case (iii) holds then the while-loop initiated in line 16 of algorithm BUILD-CYCLE implies that there must exist a cycle C in N such that $t(r(C)) = \delta(Z) = m$. Since $\mathscr{N}$ is returned by algorithm NETWORK-POPPING when given δ and $\mathscr{N}$ is clearly a level-1 representation for $\delta_{\mathscr{N}}$ it follows that $\delta_{\mathscr{N}}(Z) = t(r(C)) = m = \delta(Z)$. □

As a first result concerning algorithm NETWORK-POPPING, we have

Proposition 6 *Suppose δ is a symbolic 3-dissimilarity on X, and* NETWORK-POPPING *applied to δ returns a labelled level-1 network $\mathscr{N}$. Then $\delta = \delta_{\mathscr{N}}$. In particular, $\mathscr{N}$ is a level-1 representation for δ.*

Proof Put $\mathscr{N} = (N, t)$. In view of Proposition 5, it suffices to show that $\delta(a, b) = \delta_{\mathscr{N}}(a, b)$ holds for all $a, b \in X$ distinct. Let a and b denote two such elements. We distinguish between the cases that either (i) there exists a cycle C of N such that $v_C(a) \neq v_C(b)$, or (ii) that no such cycle exists.

Assume first that Case (i) holds. Then a and b lie either on the same side of C, or one of a and b is below the hybrid $h(C)$ of C and the other lies on the side of C, or a and b lie on different sides of C. If a and b lie on the same side of C or one of them is below $h(C)$ then we may assume without loss of generality that there exists a directed path in C from $v_C(a)$ to $v_C(b)$. Then line 22 of algorithm BUILD-CYCLE implies $t(v_C(a)) = \delta(a, b)$. Since $lca(a, b) = v_C(a)$, it follows that $\delta_{\mathscr{N}}(a, b) = t(v_C(a)) = \delta(a, b)$, as required.

If a and b lie on different sides of C then $x||ab$ is a δ-tricycle, for x as in line 2 of algorithm BUILD-CYCLE. Since that algorithm completes, line 7 of that algorithm implies $\delta(a, b) = t(r(C))$. But then $\delta_{\mathscr{N}}(a, b) = t(r(C)) = \delta(a, b)$, as $\mathscr{N}$ is returned by NETWORK-POPPING.

For the remainder, assume that Case (ii) holds, that is, there exists no cycle C of N such that $v_C(a) \neq v_C(b)$. Consider the vertex $v_0 \in V(N)$ defined as follows: if the path from the root ρ_N of N to $lca(a, b)$ does not contain a vertex that is also contained in a cycle of N, then put $v_0 = \rho_N$. Otherwise let v_0 denote the last vertex on a directed path from ρ_N to $lca(a, b)$ such that v_0 belongs to a cycle Z of N. Note that $v_0 = lca(a, b)$ holds if $lca(a, b)$ is also contained in Z. Put $V = \mathscr{F}(v_1)$ where v_1 is the unique child of v_0 not contained in Z, and let V' denote the partition of V induced by $\sim_{(V, \mathscr{S}_{v_0})}$ where for any vertex $w \in V(N)$ the set $\mathscr{S}_w$ is defined as in line 12 of algorithm NETWORK-POPPING. Let $R_a, R_b \in V'$ such that $a \in R_a$ and $b \in R_b$. Then line 5 of NETWORK-POPPING implies $\hat{\delta_{\mathscr{N}}}(R_a, R_b) = \delta_{\mathscr{N}}(a, b)$ and $\hat{\delta}(R_a, R_b) = \delta(a, b)$. Since $\mathscr{N}$ is returned by NETWORK-POPPING when given δ, line 12 of that algorithm implies $\hat{\delta}(R_a, R_b) = \hat{\delta_{\mathscr{N}}}(R_a, R_b)$. Consequently, $\delta_{\mathscr{N}}(a, b) = \delta(a, b)$ holds in this case too. □

We conclude this section with some remarks concerning the runtime of algorithm NETWORK-POPPING. Suppose X and δ are as in the description of that algorithm. Then the runtime of NETWORK-POPPING manifests itself through (i) pairwise comparisons between δ-tricycles (construction of the graph $\mathscr{C}(\delta)$) and δ-triplets (Algorithm BOTTOM-UP), respectively, and (ii) comparisons between elements x of X and (a) δ-tricycles containing x to determine the pair (H, R') associated to a given connected component of $\mathscr{C}(\delta)$ and (b) δ-triplets to obtain the TopDown graph associated to a given connected component of $\mathscr{C}(\delta)$. Since the number of δ-tricycles and of δ-triplets is bounded by the number $\frac{n(n-1)(n-2)}{6}$ of 3-subsets of X and the number of δ-tricycles and of δ-triplets containing a given element $x \in X$, respectively, is bounded by the number $\frac{(n-1)(n-2)}{2}$ of 2-subsets of $X - \{x\}$, it follows that the runtime of NETWORK-POPPING is $\mathscr{O}(n^6)$.

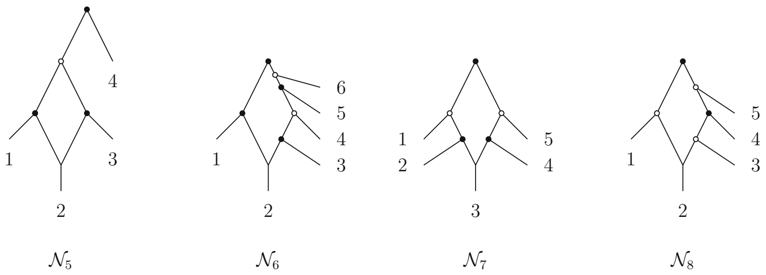

Fig. 7 The networks $\mathcal{N}_i$, $i = 5, 6, 7, 8$, considered in Table 2

7 Uniqueness of Level-1 Representations Returned by NETWORK-POPPING

As is easy to see, there exist symbolic 3-dissimilarities that although they satisfy Properties (P1)–(P6) are not level-1 representable. The reason for this is that such 3-dissimilarities need not satisfy the assumptions of lines 10 and 20 in algorithm BUILD- CYCLE. A careful analysis of that algorithm suggests however two further properties for a symbolic 3-dissimilarity to be level-1 representable. To state them, we next associate to a symbolic 3-dissimilarity its CheckLabels graph.

Suppose Y_0, Y_1, and Y_2 are three pairwise disjoint subsets of X such that for all $x, x' \in Y_0$ and all $i = 1, 2$, the graphs $TD(Y_i, x)$ and $TD(Y_i, x')$ are isomorphic (which is motivated by Property (P6)). Then we denote by $CL(Y_0, Y_1, Y_2)$ the *CheckLabels graph* associated to δ, Y_0, Y_1, and Y_2 defined as follows. The vertex set of $CL(Y_0, Y_1, Y_2)$ is $Y_0 \cup Y_1 \cup Y_2$. Any pair $(u, v) \in Y_1 \times Y_2$ is joined by an (undirected) edge $\{u, v\}$, any pair $(u, v) \in (Y_1 \cup Y_2) \times Y_0$ is joined by a directed edge (u, v), and two elements $u, v \in Y_i$, $i = 1, 2$, are joined by a directed edge (u, v) if there exists a direct path from u to v in $TD(Y_i, x)$. Finally, to each edge of $CL(Y_0, Y_1, Y_2)$ with end vertices u and v or directed edge of that graph with tail u and head v, we assign the label $\delta(u, v)$. We illustrate the CheckLabels graph in Fig. 6b for the network $\mathcal{N}_1$ depicted in Fig. 1.

Using the terminology of algorithm BUILD-CYCLE it is straight-forward to observe that the following two properties are implied by BUILD-CYCLE's lines 10 and 20 whenever its input symbolic 3-dissimilarity is level-1 representable:

(P7) All undirected edges of $CL(H, S_y, S_z)$ have the same label;
(P8) For all vertices u of $CL(H, S_y, S_z)$, all directed edges in $CL(H, S_y, S_z)$ with tail u have the same label.

As indicated in Table 2, Properties (P1)–(P8) are independent of each other. As we shall see, they allow us to characterize level-1 representable symbolic 3-dissimilarities (Theorem 2).

Theorem 2 *Let δ be a symbolic 3-dissimilarity on X. Then the following statements are equivalent (where in (iii)–(v) the input to algorithm* NETWORK-POPPING *is δ):*

Table 2 For sets X and M and δ a symbolic 3-dissimilarity on X as indicated, the property stated in the first column of each row holds whereas the remaining seven properties do not

PROP.	X	M	δ
(P1)	$\{x, y, z\}$	$\{D, S\}$	$\delta(x, y) = \delta(x, z) = \delta(y, z) = D$; $\delta(x, y, z) = S$.
(P2)	$\{x, y, z, u\}$	$\{D, S\}$	$\delta(x, y, z) = \delta(y, z, u) = \delta(x, y) = \delta(y, z) = \delta(z, u) = D$; $\delta(Y) = S$ otherwise.
(P3)	$\{x_1, x_2, y, z\}$	$\{D, S_1, S_2\}$	$\delta(x_i, y, z) = S_i, i \in \{1, 2\}$; $\delta(Y) = D$ otherwise.
(P4)	$\{x, y, z, u\}$	$\{D, S\}$	$\delta(x, y, u) = \delta(x, u) = \delta(y, z) = \delta(x, y, z) = D$; $\delta(Y) = S$ otherwise.
(P5)	$\{1, \ldots, 5\}$	$\{D, S\}$	$\delta(1, 4) = S$; $\delta(Y) = \delta_{\mathcal{N}_5}(Y)$ otherwise.
(P6)	$\{1, \ldots, 6\}$	$\{D, S\}$	$\delta(3, 6) = \delta(2, 3, 6) = D$; $\delta(Y) = \delta_{\mathcal{N}_6}(Y)$ otherwise.
(P7)	$\{1, \ldots, 5\}$	$\{D, S\}$	$\delta(2, 4) = \delta(2, 3, 4) = \delta(1, 2, 4) = \delta(2, 4, 5) = S$; $\delta(Y) = \delta_{\mathcal{N}_7}(Y)$ otherwise.
(P8)	$\{1, \ldots, 5\}$	$\{D, S\}$	$\delta(3, 5) = \delta(3, 4, 5) = D$; $\delta(Y) = \delta_{\mathcal{N}_8}(Y)$ otherwise.

For $i = 5, 6, 7, 8$, the networks $\mathcal{N}_i$ are depicted in Fig. 7

(i) *δ is level-1 representable.*
(ii) *δ satisfies conditions (P1)–(P8).*
(iii) NETWORK-POPPING *returns a labelled level-1 network which is unique up to isomorphism.*
(iv) NETWORK-POPPING *returns a level-1 representation for δ.*
(v) NETWORK-POPPING *returns a semi-discriminating level-1 representation for δ.*

Proof (i) ⇒ (ii): This is an immediate consequence of Lemma 3, Proposition 1, the remark preceding Proposition 3 and the observation preceding Table 2.

(ii) ⇒ (iii): Assume that δ satisfies Properties (P1)–(P8). Then algorithm FIND-CYCLES first constructs the graph $\mathscr{C}(\delta)$ and then finds for each connected component K of $\mathscr{C}(\delta)$ the pair (H_K, R'_K). Since algorithm BUILD-CYCLES relies on Properties (P3), (P4), (P6)–(P8) being satisfied, it follows that BUILD-CYCLES constructs for each pair (H_K, R'_K), K a connected component of $\mathscr{C}(\delta)$, a labelled simple level-1 network as specified in the output of BUILD-CYCLES. By construction, the labelled DAG $\mathcal{N} = (N, t)$ returned by algorithm NETWORK-POPPING is clearly a labelled phylogenetic network. Since, in view of the while loop of that algorithm starting at line 7, no two cycles in N can share a vertex it follows that N is in fact a level-1 network. Proposition 5 combined with the observation that in none of our four algorithms we have to break a tie implies that $\mathcal{N}$ is unique up to isomorphism.

(iii) ⇒ (iv): This is trivial in view of Proposition 6.

(iv) ⇒ (v): Suppose algorithm Network-Popping returns a level-1 representation $\mathscr{N}$ for δ. To see that $\mathscr{N}$ is in fact semi-discriminating, note that algorithms Vertex-Growing and Build-Cycles return a discriminating symbolic representation and a discriminating level-1 representation for its input symbolic 3-dissimilarity, respectively. In combination it follows that $\mathscr{N}$ must be semi-discriminating.

(v) ⇒ (i): This is trivial. □

As suggested by the two semi-discriminating level-1 representations $\mathscr{N}_1$ and $\mathscr{N}_3$ for $\delta_{\mathscr{N}_1}$ depicted in Fig. 1, the output of algorithm Network Popping when given a level-1 representable symbolic 3-dissimilarity δ need not be the labelled level-1 network that induced δ. To help clarify the relationship between both networks, we require further terminology.

Suppose that (N, t) is a labelled level-1 network. Then we say that a cycle C of N is *weakly labelled* if there exists at least one vertex v on either side of C such that $t(v) \neq t(r(\mathscr{C}))$. More generally, we call a labelled level-1 network (N, t) *weakly labelled* if every cycle of N is weakly labelled. For example, the labelled level-1 network $\mathscr{N}_2$ pictured in Fig. 1 is weakly labelled (but not semi-discriminating) whereas the network $\mathscr{N}_3$ depicted in Fig. 1 is semi-discriminating but not weakly labelled.

Armed with this definition, we can characterize weakly labelled cycles as follows.

Lemma 8 *Let $\mathscr{N} = (N, t)$ be a labelled level-1 network, and let C be a cycle of N. Then C is weakly labelled if and only if there exists some $x \in H(C)$ and leaves $y, z \in R(C) - H(C)$ that lie on different sides of C such that $x||yz$ is a $\delta_{\mathscr{N}}$-tricycle. Moreover, $x'||yz$ is a $\delta_{\mathscr{N}}$- tricycle, for all $x' \in H(C)$.*

Proof Put $\delta = \delta_{\mathscr{N}}$. Assume first that there exists some $x \in H(C)$ and leaves $y, z \in R(C) - H(C)$ that lie on two different sides of C such that $x||yz$ is a δ-tricycle. Then $\delta(x, y, z) = \delta(z, y) = t(r(\mathscr{C}))$. Also $\delta(x, y) = t(v_C(y))$ and $\delta(x, z) = t(v_C(z))$. In view of Table 1, $\delta(x, y, z) \notin \{\delta(x, y), \delta(x, z)\}$ and, so, $t(v_C(i)) \neq t(r(C))$, for $i = y, z$.

Conversely, suppose C is weakly labelled. Let $v_1, v_2 \in V(C)$ denote two vertices of N that lie on different directed paths from $r(C)$ to $h(C)$ such that $t(r(C)) \notin \{t(v_1), t(v_2)\}$. Suppose $y, z \in X$ are such that $v_C(y) = v_1$ and $v_C(z) = v_2$. Then $x||yz$ must be a δ-tricycle, for all $x \in H(C)$. Indeed, $\delta(x, y) = t(v_1)$ and $\delta(x, z) = t(v_2)$ holds. Since $\delta(y, z) = \delta(x, y, z) = t(r(C)) \notin \{\delta(x, y), \delta(x, z)\}$, Table 1 implies that $x||yz$ is a δ-tricycle.

The remainder of the lemma follows from the fact that, for all $x' \in H(C)$, we have $\delta(x', y, z) = \delta(x, y, z)$, $\delta(x, y) = \delta(x', y)$ and $\delta(x, z) = \delta(x', z)$. □

As a consequence, we can strengthen Proposition 1 to the following characterization.

Theorem 3 *If $\mathscr{N} = (N, t)$ is a labelled level-1 network, the connected components of $\mathscr{C}(\delta_{\mathscr{N}})$ are in 1–1 correspondence with the weakly labelled cycles of N.*

Implied by Theorem 3, we have

Corollary 1 *Let δ be a level-1 representable symbolic 3-dissimilarity on X, and let $\mathcal{N} = (N, t)$ be the level-1 representation of δ returned by algorithm* Network-Popping *when applied to δ. Then $\mathcal{N}$ is weakly labelled if and only if, for any level-1 representation $\mathcal{N}' = (N', t')$ of δ, the number of cycles in N equals the number of weakly labelled cycles in N'. In particular, the number of cycles in N is minimal.*

```
Input: A labelled level-1 network $\mathcal{N} = (N, t)$ on $X$.
Output: A semi-discriminating, weakly labelled, partially resolved level-1 network $\mathcal{N}' = (N', t')$
        such that $\delta_{\mathcal{N}} = \delta_{\mathcal{N}'}$.
set $\mathcal{N}' = \mathcal{N}$;
while $\mathcal{N}'$ is not semi-discriminating or not weakly labelled or not partially resolved do
    Collapse all edges $(u, v)$ satisfying $t'(u) = t'(v)$ and such that either $u$ and $v$ belong to the same
    cycle of $N'$ or do not belong to a cycle;
    for All vertices $v$ of a cycle $C$ of degree 4 or more do
        Define a new child $w$ of $v$;
        set $t'(w) = t'(v)$;
        if $v = r(C)$ then
            Redefine the children of $v$ in $C$ as children of $w$;
        end
        else
            Redefine the children of $v$ outside of $C$ as children of $w$;
        end
    end
    for All cycles $C$ of $N'$ such that $(r(C), h(C))$ is an edge of $N'$ do
        Remove the edge $(r(C), h(C))$;
    end
    Remove all vertices of degree 2;
end
```

Algorithm 5: Transform

Corollary 2 *Suppose $\mathcal{N}$ is a labelled level-1 network and $\mathcal{N}'$ is the level-1 representation for $\delta_{\mathcal{N}}$ returned by algorithm* Network-Popping. *Then $\mathcal{N}'$ is isomorphic with the labelled level-1 network returned by algorithm* Transform *when given $\mathcal{N}$ as input. In particular, $\mathcal{N}$ and $\mathcal{N}'$ are isomorphic if and only if $\mathcal{N}$ is semi-discriminating, weakly labelled, and partially resolved. Furthermore, if δ is a level-1 representable symbolic 3-dissimilarity, then there exists an unique representation of δ that is semi-discriminating, weakly labelled, and partially resolved.*

8 Characterizing Level-1 Representable Symbolic 3-Dissimilarities

In this section, we present a characterization of level-1 representable symbolic 3-dissimilarities on X in terms of level-1 representable symbolic 3-dissimilarities on subsets of X of size $|X| - 1$ (Theorem 4). Combined with the fact that algorithm Network- Popping has polynomial run time, this suggests that Network- Popping might lend itself to studies involving large data sets using a Divide-and-Conquer approach.

At the heart of the proof of our characterization lies the following technical lemma which concerns the question under what circumstances the restriction of a level-1 representable symbolic 3-dissimilarity δ on X is itself level-1 representable. Central to its proof is the fact that $|X| \neq 4$ since, in general, a symbolic 3-dissimilarity δ on a set X of size 4 need not be level-1 representable but the restriction of δ to any subset of size 3 is level-1 representable. An example for this is furnished by the symbolic 3-dissimilarity δ on $X = \{x, y, z, u\}$, given by $\delta(x, y, z) = \delta(y, z, u) = \delta(x, y) = \delta(y, z) = \delta(z, u) \neq \delta(x, z) = \delta(x, u) = \delta(y, u) = \delta(x, z, u) = \delta(x, y, u)$.

Using the assumptions and definitions for the elements x, y, and z, and the sets H, S_z, and S_y made in algorithm Build- Cycle, we have the following result.

Lemma 9 *Suppose δ is a symbolic 3-dissimilarity on X satisfying Properties (P1), (P2), (P4), and (P6), $x||yz$ is the δ-tricycle chosen in line 2 of algorithm* Build- Cycle, *and $i \in \{y, z\}$. If $u, w \in S_i$ are joined by a direct path from u to w in $TD(S_i, x)$, then either (u, w) is a directed edge of $TD(S_i, x)$ or there exists $v \in S_i$ such that both directed edges (u, v) and (v, w) are contained in $TD(S_i, x)$.*

Proof By symmetry, we may assume $i = y$. Suppose there exists a directed path $v_0 = u, v_1, \ldots, v_k, v_{k+1} = w$, some $k \geq 0$, from u to w in $TD(S_y, x)$ and that (u, w) is not a directed edge on that path. Then $k \geq 1$ and, so, $v_1 \notin \{u, w\}$. It suffices to show that (v_1, w) is a directed edge of $TD(S_y, x)$.

Observe first that, in view of Property (P6), (w, u) is not a directed edge in $TD(S_y, x)$ as otherwise $TD(S_y, x)$ would contain a directed cycle. Combined with the definition of S_y it follows that either $x|uw$ is a δ-triplet or we have a δ-fork on $\{x, u, w\}$. In either case, $\delta(u, x) = \delta(w, x)$ holds. Since (u, v_1) is a directed edge in $TD(S_y, x)$, we also have that $xv_1|u$ is a δ-triplet. Hence, $\delta(v_1, x) \neq \delta(x, u) = \delta(w, x)$ and so we cannot have a δ-fork on $\{x, w, v_1\}$. Since, in view of Property (P4), we cannot have a δ-tricycle on $\{x, w, v_1\}$ either $\delta(w, v_1) = \delta(w, x)$ or $\delta(w, v_1) = \delta(v_1, x)$ follows.

If the first equality holds, then $v_1x|w$ is a δ-triplet and, so, (w, v_1) is a directed edge in $TD(S_y, x)$. Consequently, the directed path $v_1, \ldots, v_k, w$ concatenated with that edge forms a directed cycle in $TD(S_y, x)$, which is impossible in view of Property (P6) holding. Thus, $\delta(w, v_1) = \delta(v_1, x)$ must hold. Consequently, $wx|v_1$ is a δ-triplet and, so, (v_1, w) is an edge in $TD(S_y, x)$, as required. □

To establish the main result of this section (Theorem 4), we need to be able to distinguish between the sets defined in lines 8 and 9 of algorithm Build- Cycle when given a symbolic 3-dissimilarity δ on X and the restriction $\delta|_Y$ of δ to a subset $Y \subseteq X$ with $|Y| \geq 3$. To this end, we augment for a symbolic 3-dissimilarity κ on X the definition of those sets by writing $S_i(\kappa)$ rather than S_i, $i = y, z$.

Observe first that if δ is level-1 representable and $Y \subseteq X$ such that $|Y| \geq 3$, then the restriction $\delta|_Y$ of δ to Y is clearly level-1 representable. Indeed, a level-1 representation $\mathscr{N}(\delta|_Y)$ of $\delta|_Y$ can be obtained from a level-1 representation $\mathscr{N}(\delta)$ of δ using the following 2-step process. First, remove all leaves in $X - Y$ and their respective incoming edges from $\mathscr{N}(\delta)$ and then suppress all resulting degree two vertices. Next, apply algorithm Transform to the resulting network. This begs the question of when level-1 representations of symbolic 3-dissimilarities on subsets of

X give rise to a level-1 representation of a symbolic 3-dissimilarity on X. To answer this question which is the purpose of Theorem 4 we require the next result.

Proposition 7 *Let δ be a symbolic 3-dissimilarity on X. Then the following statements hold.*

(i) If $|X| \geq 6$ and δ does not satisfy Property (Pi), $i \in \{1, 2, \ldots, 8\}$, then there exists some $Y \subseteq X$ with $3 \leq |Y| \leq 5$ such that that property is also not satisfied by $\delta|_Y$.
(ii) If $|X| \geq 6$ and δ is not level-1 representable then there exists some $Y \subseteq X$ with $3 \leq |Y| \leq 5$ such that $\delta|_Y$ is also not level-1 representable.

Proof (i) The proposition is straight-forward to show for Properties (P1) and (P2), since they involve three and four elements of X, respectively. Note that to see Property (Pi), $3 \leq i \leq 8$, we may assume without loss of generality that Properties (Pj), $1 \leq j \leq i-1$, are satisfied by δ. For ease of readability, we put $S_y := S_y(\delta)$.

If δ does not satisfy Property (P3) then there exists a connected component C of $\mathscr{C}(\delta)$ and δ-tricycles $\tau, \tau' \in V(C)$ such that $\delta(L(\tau)) \neq \delta(L(\tau'))$. Without loss of generality, we may assume that τ and τ' are adjacent. Then $|L(\tau) \cap L(\tau')| = 2$. Let $x, y, z \in X$ such that $\tau = x||yz$. Then either $\tau' = x'||yz$ or $\tau' = x||yz'$ where $x', z' \in X$. But then Property (P3) is not satisfied either for δ restricted to the 5-set $Z = \{x, y, z, x', z'\}$.

For the remainder, let (H, R') denote the pair returned by algorithm FIND-CYCLES when given δ and let $x \in H$ and $y, z \in R'$ such that $x||yz$ is a vertex in the connected component C of $\mathscr{C}(\delta)$ corresponding to (H, R'). Suppose δ does not satisfy Property (P4). Assume first that the second part of Property (P4) is not satisfied. Then if there exists an element u contained in $H \cap S_y$ or in $H \cap S_z$ or in $S_z \cap S_y$ then u is also contained in the corresponding intersections involving the sets $S_y(\delta|_Z) \subseteq S_y$ and $S_z(\delta|_Z) \subseteq S_z$ found by BUILD-CYCLE in its lines lines 8 and 9 for δ restricted to $Z = \{x, y, z, u\}$. Thus, the second part of Property (P4) does not hold for $\delta|_Z$.

Now assume that the first part of Property (P4) does not hold for δ, that is, $S'_i \neq A := \{w \in S_i : \delta(w, x) \neq \delta(y, z)\}$. By symmetry, we may assume without loss of generality that $i = y$. Then since $S'_y \subseteq A$ clearly holds there must exists some $w \in A - S'_y$. Put $U = \{x, y, z, w\}$. Then $w \notin S'_y(\delta|_U)$ as $w \notin S'_y$. However we clearly have that $w \in S_y(\delta|_U)$ and $\delta|_U(w, x) \neq \delta|_U(y, z)$. Thus, the first part of Property (P4) is not satisfied with δ replaced by $\delta|_U$.

If δ does not satisfy Property (P5) then since $y \in R := H \cup S_y \cup S_z$ it follows for $u := y$ and v and w as in the statement of Property (P5) that the restriction of δ to $\{x, u, z, v, w\}$ does not satisfy Property (P5) either.

If δ does not satisfy Property (P6) then either (a) there exist elements $u, u' \in H$ such that $TD(S_y, u)$ and $TD(S_y, u')$ are not isomorphic or (b) there exists some $u \in H$ such that $TD(S_y, u)$ has a directed cycle C.

Assume first that Case (a) holds. Then there must exist distinct vertices v and w in S_y such that (v, w) is a directed edge in $TD(S_y, u)$ but not in $TD(S_y, u')$. With $Z = \{v, u, u', w, z\}$ it follows that $S_v(\delta|_Z) = \{v, w\}$. Since the directed edge (v, w) is clearly contained in the TopDown graph $TD(\{v, w\}, u)$ associated to $\delta|_Z$ but not in the TopDown graph $TD(\{v, w\}, u')$ associated to $\delta|_Z$, Property (P6) is not satisfied for $\delta|_Z$.

Thus, Case (b) must hold. In view of Proposition 3(i), we may assume that the size of C is three. Hence, the subgraph G of $TD(S_y, u)$ induced by $Z = V(C) \cup \{z, u\}$ also contains a cycle of length 3. Since G coincides with the TopDown graph $TD(V(C), u)$ for $\delta|_Z$ and $|Z| = 5$ holds, it follows that $\delta|_Z$ does not satisfy Property (P6).

If δ does not satisfy Property (P7) then there must exist undirected edges $e = \{a, b\}$ and $e' = \{a', b'\}$ in $CL(H, S_y, S_z)$ such that $\delta(a, b) \neq \delta(a', b')$. Then for at least one of e and e', say e, we must have that $\delta(a, b) \neq \delta(y, z)$. Put $Z = \{x, y, z, a, b\}$. Then since $\{y, z\}$ is also an undirected edge in $CL(H, S_y(\delta|_Z), S_z(\delta|_Z))$ it follows that $\delta|_Z$ does not satisfy Property (P7) either.

Finally, suppose that δ does not satisfy Property (P8). Considering both alternatives in the statement of Property (P8) together, there must exist vertices $u \in S_y$ and $v, w \in S_y \cup H$ such that both (u, v) and (u, w) are directed edges of $CL(H, S_y, S_z)$ and $\delta(u, v) \neq \delta(u, w)$. Independent of whether $v, w \in S_y$ or $v, w \in H$ or $v \in S_y$ and $w \in H$, it follows that either $\delta(u, x) \neq \delta(u, v)$ or $\delta(u, x) \neq \delta(u, w)$. Assume without loss of generality that $\delta(u, x) \neq \delta(u, v)$. Note that (u, x) is also a directed edge in $CL(H, S_y, S_z)$.

If $v \in H$, then $\delta|_Z$ does not satisfy Property (P8) for $Z = \{x, y, z, u, v\}$. So assume $v \notin H$. Then $v \in S_y$. Since (u, v) is a directed edge in $CL(H, S_y, S_z)$ it follows that there exists a directed path P from u to v in $TD(S_y, x)$. By Lemma 9, either (a) P has a single directed edge or (b) there exists some $v_1 \in S_y$ such that both (u, v_1) and (v_1, v) are directed edges of $TD(S_y, x)$.

If Case (a) holds, then $\delta|_Z$ does not satisfy Property (P8) for $Z = \{x, y, z, u, v\}$. So assume that Case (b) holds. Then $\delta|_{Z'}$ does not satisfy Property (P8) for $Z' = \{x, y, z, u, v, v_1\}$. Since the definition of $TD(S_y, x)$ implies that $xv|v_1$ is a δ-triplet, it follows that $\delta(x, v) \neq \delta(x, v_1)$. Hence, either $\delta(v, x) \neq \delta(v, z)$ or $\delta(v_1, x) \neq \delta(v, z)$. By Properties (P3) and (P4) it follows in the first case that $x||vz$ is a δ-tricycle, and that $x||v_1z$ is a δ-tricycle in the second case. Thus, either v or v_1 can play the role of y in τ. Consequently, δ restricted to $Z = Z' - \{y\}$ does not satisfy Property (P8).

(ii) This is a straight-forward consequence of Theorem 2 and Proposition 7(i). □

Theorem 4 *Let δ be a symbolic 3-dissimilarity on a set X such that $|X| \geq 6$. Then δ is level-1 representable if and only if for all subsets $Y \subseteq X$ of size $|X| - 1$, the restriction $\delta|_Y$ is level-1 representable.*

Proof Suppose first that δ is level-1 representable. Then, by the observation preceding Proposition 7, $\delta|_Y$ is level-1 representable, for all subsets $Y \subseteq X$ of size $|X| - 1$.

Conversely, suppose that X is such that for all subsets $Y \subseteq X$ of size $|X| - 1$, the restriction $\delta|_Y$ is level-1 representable but that δ is not level-1 representable. Then, by Proposition 7 there exists a subset $Y \subseteq X$ with $|Y| \in \{3, 4, 5\}$ such that $\delta|_Y$ is also not level-1 representable. But then δ restricted to any subset Z of X size $|X| - 1$ that contains Y also is not level-1 representable which is impossible. □

9 Conclusion

Orthology relations have been successfully used to shed light into the evolution of gene families. Motivated by the fact that the signal in such relations might be obscured by e. g. noise or error (or indeed true evolutionary signal) we propose to represent them in terms of a phylogenetic network (as opposed to a phylogenetic tree). As a first step towards the development of a general framework for representing orthology relations in terms of phylogenetic networks, we focus on the novel concept of a level-1 representation of such a relation.

Motivated by the biological concept of a "cluster of orthologous gene (COG)", we formalize a orthology relation in terms of the novel concept of a symbolic 3-dissimilarity To compute a level-1 representation from a symbolic 3-dissimilarity, we introduce the novel Network- Popping algorithm. It takes as input a symbolic 3-dissimilarity δ, and finds, in time $\mathscr{O}(|X|^6)$, a level-1 representation of δ precisely if such a representation exists. In addition to this representation being a discriminating symbolic representation of δ precisely if such a tree is supported by δ, Network- Popping enjoys several other attractive properties. As part of our analysis of Network-Popping, we characterize level-1 representable symbolic 3-dissimilarities δ in terms of eight natural properties that δ must satisfy. Last-but-not-least, we also characterize a level-1 representable symbolic 3-dissimilarity δ on some set X with $|X| \geq 6$ in terms of level-1 representable orthology relations induced by δ on subsets of X of size $|X| - 1$. Combined with the polynomial run-time of Network- Popping this suggests that it could potentially be applied to large data sets within a Divide-and-Conquer framework thus providing an alternative to tree-based reconciliation or error correction approaches for orthology relations.

However a number of open questions remain. For example can other types of phylogenetic networks be used to also represent orthology relations. Interesting types of such networks might be tree-child networks [30] as they are uniquely determined by the trinets they induce and also regular networks [32] as they are known to be uniquely determined by the phylogenetic trees they induce, a property that is not shared by phylogenetic networks in general [7]. For those networks it would also be interesting to understand how the representation of an orthology relation in terms of those trees relates to the way such a relation is represented by the labelled network displaying the trees. Motivated by the point made in [6, Chapter 12] on estimates of k-subsets, $k \geq 3$, already mentioned above it might also be interesting to investigate if symbolic k-dissimilarities for $k \geq 4$ lend themselves as useful formalizations of orthology relations.

A further question concerns the fact that by evoking parsimony we only distinguish between three types of trinets associated to an orthology relation. Thus it might be interesting to investigate what can be done if this framework is replaced by e. g. a probabilistic one which assigns probability values to the trinets. Given that in e. g. the case of COG's [27], ortholoy relation detection is sequence based such values could potentially be obtained by adjusting the ideas presented in [15,22,33] which assign likelihood scores to networks. Alternatively, it might be interesting to see if an coalescent type approach along the lines of [31] could be made to work.

Acknowledgements The authors would like to thank M. Taylor for stimulating discussions on orthology relations. In addition, they would like to thank both referee for their helpful comments.

References

1. Altenhoff, A.M., Dessimoz, C.: Phylogenetic and functional assessment of orthologs inference projects and methods. PLoS Comput. Biol. **5**, e1000262 (2009)
2. Bansal, M.S., Alm, E.J., Kellis, M.: Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer, and loss. Bioinformatics **28**, i283–i291 (2012)
3. Böcker, S., Dress, A.W.M.: Recovering symbolically dated, rooted trees from symbolic ultrametrics. Adv. Math. **138**, 105–125 (1998)
4. Chen, K., Durand, D., Farach-Colton, M.: Notung: a program for dating gene duplications and optimizing gene family trees. J. Comput. Biol. **7**, 429–47 (2000)
5. Consortium, T.G.O.: Gene ontology: tool for the unification of biology. Nat. Genet. **25**, 25–29 (2000)
6. Felsenstein, J.: Inferring Phylogenies. Sinauer Associates, Sunderland, Massachusetts (2003)
7. Gambette, P., Huber, K.T.: On encodings of phylogenetic networks of bounded level. J. Math. Biol. **61**(1), 157–180 (2012)
8. Górecki, P., Burleigh, G., Eulenstein, O.: Maximum likelihood models and algorithms for gene tree evolution with duplications and losses. BMC Bioinform. **12**, S15 (2011)
9. Hellmuth, M., Hernandez-Rosales, M., Huber, K.T., Moulton, V., Stadler, P.F., Wieseke, N.: Orthology relations, symbolic ultrametrics and cographs. J. Math. Biol. **66**(1–2), 399–420 (2013)
10. Hellmuth, M., Wieseke, N.: On symbolic ultrametrics, cotree representation, and cograph edge decomposition and partition. Comput. Combin. **9198**, 609–623 (2015)
11. Huber, K.T., Moulton, V.: Encoding and constructing 1-nested phylogenetic networks with trinets. Algorithmica **66**(3), 714–738 (2013)
12. Huber, K.T., van Iersel, L.J.J., Moulton, V., Scornavacca, C.: Reconstructing phylogenetic level-1 networks from nondense binet and trinet sets. Algorithmica (**in press**)
13. Huson, D., Rupp, R., Scornavacca, C.: Phylogenetic Networks. Cambridge University Press, Cambridge (2010)
14. Jacox, E., Chauve, C., Szöllösi, G., Ponty, Y., Scornavacca, C.: ecceTERA: comprehensive gene tree-species tree reconciliation using parsimony. Bioinformatics **32**, 2056–2058 (2016)
15. Jin, G., Nakhleh, L., Snir, S., Tuller, T.: Maximum likelihood of phylogenetic networks. Bioinformatics **22**(21), 2604–2611 (2006)
16. Jun, J., Mandoiu, I.I., Nelson, C.E.: Identification of mammalian orthologs using local synteny. BMC Genom. **10**, 630 (2009)
17. Kordi, M., Bansal, M.: On the complexity of duplication-transfer-loss reconciliation with non-binary gene trees. IEEE/ACM Trans. Comput. Biol. Bioinform. (**in press**)
18. Lafond, M., El-Mabrouk, N.: Orthology relation and gene tree correction: complexity results. In: WABI 2015, Algorithms in Bioinformatics, vol. 9289 of LNCS, pp. 966–979 (2015)
19. Lafond, M., Semeria, M., Swenson, K.M., Tannier, E., El-Mabrouk, N.: Gene tree correction guided by orthology. BMC Bioinform. **14**, S5 (2013)
20. Mahmudi, O., Sjöstrand, J., Sennblad, B., Lagergren, J.: Genome-wide probabilistic reconciliation analysis across vertebrates. BMC Bioinform. **14**, S10 (2013)
21. Nakhleh, L.: Computational approaches to species phylogeny inference and gene tree reconciliation. Trends Ecol. Evol. **28**(12), 719–728 (2013)
22. Oldman, J., Wu, T., van Iersel, L., Moulton, V.: Trilonet: piecing together small networks to reconstruct reticulate evolutionary histories. Mol. Biol. Evol. (2016)
23. Ovadia, Y., Fielder, D., Conow, C., Libeskind-Hadas, R.: The cophylogeny reconstruction problem is NP-complete. J. Comput. Biol. **18**, 59–65 (2011)
24. Semple, C., Steel, M.: Phylogenetics. Oxford University Press, Oxford (2003)
25. Stolzer, M., Lai, H., Xu, M., Sathaye, D., Vernot, B., Durand, D.: Inferring duplications, losses, transfers and incomplete lineage sorting with nonbinary species trees. Bioinformatics **28**, i409–i415 (2012)
26. Tatusov, R., Galperin, M.Y., Natale, D.A., Koonin, E.V.: The COG database: a tool for genome-scale analysis of protein functions and evolution. Nucleic Acids Res. **28**, 33–36 (2000)

27. Tatusov, R., Koonin, E.V., Lipman, D.J.: A genomic perspective on protein families. Science **278**, 631–637 (1997)
28. Tekaia, F.: Inferring orthologs: open questions and perspectives. Genom. Insights **9**, 17–28 (2016)
29. Tofigh, A., Hallett, M., Lagergren, J.: Simultaneous identification of duplications and lateral gene transfers. IEE/ACM Trans. Comput. Biol. Bioinform. **8**(2), 517–535 (2011)
30. van Iersel, L.J.J., Moulton, V.: Trinets encode tree-child and level-2 phylogenetic networks. J. Math. Biol. **68**(7), 1707–1729 (2014)
31. Wen, D., Yu, Y., Nakhleh, L.: Bayesian inference of reticulate phylogenies under the multispecies network coalescent. PLoS Genet. **12**(5), e1006006 (2016)
32. Willson, S.: Regular networks are determined by their trees. IEEE/ACM Trans. Comput. Biol. Bioinform. **8**, 785–796 (2011)
33. Yu, Y., Dong, J., Liu, K.J., Nakhleh, L.: Maximum likelihood inference of reticulate evolutionary histories. Proc. Natl. Acad. Sci. **111**(46), 16448–16453 (2014)