Squirrel: Reconstructing semi-directed phylogenetic level-1 networks from four-leaved networks or sequence alignments

Niels Holtgrefe[1], Katharina T. Huber[2], Leo van Iersel[1], Mark Jones[1], Samuel Martin[3], and Vincent Moulton[2,*]

[1] *Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628CD, Delft, The Netherlands*
[2] *School of Computing Sciences, University of East Anglia, NR4 7TJ, Norwich, United Kingdom*
[3] *European Bioinformatics Institute, CB10 1SD, Hinxton, United Kingdom*
[*] *Corresponding author: v.moulton@uea.ac.uk*

Abstract

With the increasing availability of genomic data, biologists aim to find more accurate descriptions of evolutionary histories influenced by secondary contact, where diverging lineages reconnect before diverging again. Such reticulate evolutionary events can be more accurately represented in phylogenetic networks than in phylogenetic trees. Since the root location of phylogenetic networks can not be inferred from biological data under several evolutionary models, we consider semi-directed (phylogenetic) networks: partially directed graphs without a root in which the directed edges represent reticulate evolutionary events. By specifying a known outgroup, the rooted topology can be recovered from such networks. We introduce the algorithm Squirrel (Semi-directed Quarnet-based Inference to Reconstruct Level-1 Networks) which constructs a semi-directed level-1 network from a full set of quarnets (four-leaf semi-directed networks). Our method also includes a heuristic to construct such a quarnet set directly from sequence alignments. We demonstrate Squirrel's performance through simulations and on real sequence data sets, the largest of which contains 29 aligned sequences close to 1.7 Mbp long. The resulting networks are obtained on a standard laptop within a few minutes. Lastly, we prove that Squirrel is combinatorially consistent: given a full set of quarnets coming from a triangle-free semi-directed level-1 network, it is guaranteed to reconstruct the original network. Squirrel is implemented in Python, has an easy-to-use graphical user-interface that takes sequence alignments or quarnets as input, and is freely available at https://github.com/nholtgrefe/squirrel.

Keywords: semi-directed phylogenetic network, rooted phylogenetic network, quarnet, travelling salesman problem, sequence alignment, network reconstruction

Introduction

Secondary contact, where diverging lineages come into contact and hybridize before continuing to diverge, is commonplace in evolution. This process is poorly described by most phylogenetic reconstruction methods which generally assume a bifurcating tree model. Secondary contact has been widely documented for diverse sets of taxa, including viruses (e.g. HIV and SARS-CoV-2, see Worobey et al. 2008; Pekar et al. 2021; Jiao et al. 2024), bacteria (e.g. Diop et al. 2022), plants (e.g. Ehrendorfer 1959; Rieseberg et al. 2003), birds (e.g. Taylor and Larson 2019), fish (e.g. Meier et al. 2019; Du et al. 2024), invertebrates (e.g. Zhang et al. 2016) and primates, including humans (e.g. Patterson et al. 2006; Green et al. 2010). Through secondary contact, introgression — the exchange of genetic material between hybridizing lineages — may occur by means of complex processes, often involving multiple rounds of backcrossing.

Evolutionary histories shaped by secondary contact can be more accurately represented by rooted phylogenetic level-1 networks than by strictly bifurcating rooted phylogenetic trees. Rooted phylogenetic level-1 networks are directed acyclic graphs that are largely tree-like in structure, describing patterns of divergence, but include localized reticulations where lineages have merged through reticulate events (see e.g. Figure 1(a) and see the Materials and Methods for a more formal definition). Application of these networks is highly desirable, but their construction is computationally intensive, and their use has remained out of reach for most biologists. Results reported here, including an efficient algorithm and software, address the challenge of building phylogenetic level-1 networks, thus offering the possibility of finding a more realistic description of biological diversity.

Our results are achieved by considering *semi-directed (phylogenetic) networks* (Solís-Lemus and Ané 2016), in which there is no root and only branches representing reticulate events carry information about direction (see the Materials and Methods for a more formal definition). These networks have gained considerable interest recently (see e.g. Solís-Lemus and Ané 2016; Allman et al. 2019; Frohn et al. 2024; Kong et al. 2024; Warnow et al. 2024; Wu and Solís-Lemus 2024), as it has been shown that under certain models of evolution it is theoretically impossible to infer the root of a rooted phylogenetic network

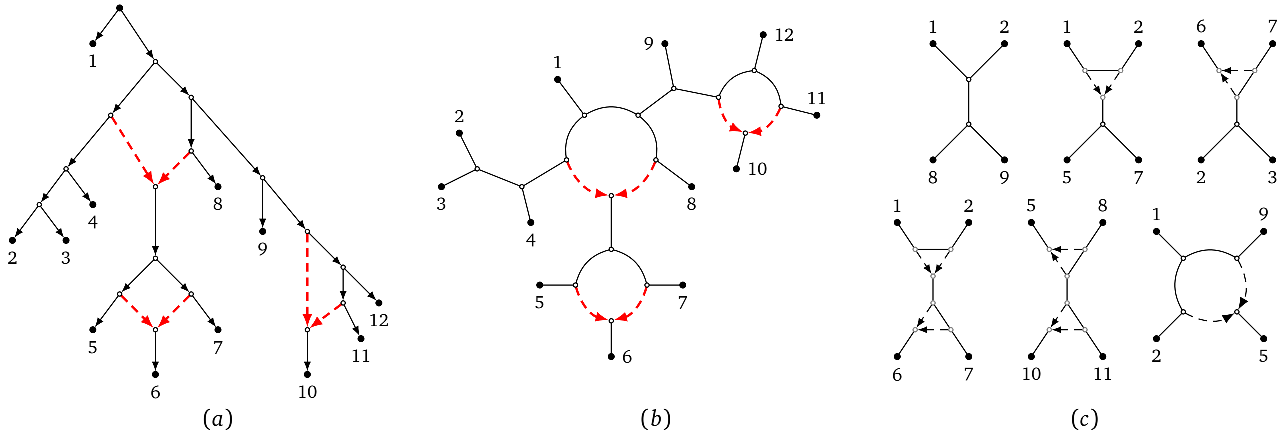


Figure 1: (a): A rooted phylogenetic level-1 network on 12 taxa represented by numbers $1-12$, with the dashed reticulation edges pointing towards reticulation vertices which represent reticulate events. (b): The semi-directed topology of the rooted network, which is a triangle-free semi-directed level-1 network on 12 leaves, again with the reticulation edges dashed. This network uniquely determines the rooted network by specifying leaf 1 as an outgroup. (c): Some of the quarnets induced by the semi-directed network. When ignoring the leaf labels, these are all six possible level-1 quarnet shapes. The top left quarnet is a quartet tree, the bottom right quarnet is the only one that contains a cycle of length 4 (4-cycle), and the other four quarnets contain one or two triangles (3-cycles). The tf-quarnets (triangle-free quarnets) can be obtained from the quarnets by contracting each of the triangles to a single node. The quartet tree and quarnet with a 4-cycle are both already triangle-free.

directly from data (Baños 2019; Gross et al. 2021; Xu and Ané 2023). For an example of a semi-directed level-1 network see Figure 1(b). In case an outgroup is available, this can be used to root the semi-directed network (Solís-Lemus and Ané 2016), as illustrated in Figure 1(a) and (b). Several identifiability results have been recently proven for semi-directed level-1 networks. In particular, it was shown that such networks can be theoretically recovered from data under various models of evolution (Baños 2019; Gross et al. 2021; Xu and Ané 2023). By focusing on semi-directed networks, we offer a tractable way for reconstructing phylogenetic level-1 networks.

Recently, two algebraic approaches have been introduced to construct semi-directed level-1, four-leaved networks, or *quarnets* (see Figure 1(c)): QNR-SVM (Barton et al. 2022) and an algorithm in Martin et al. (2023). These methods take as input sequence data and both employ algebraic invariants to infer quarnets under the Jukes-Cantor model (Barton et al. 2022; Martin et al. 2023) and the Kimura 2-parameter model (Martin et al. 2023). To infer evolutionary relationships for larger data sets, methods are therefore required to puzzle together such quarnets into larger networks (see e.g. Schmidt et al. (2002) and Oldman et al. (2016) for two of the earliest algorithms where this approach was used for trees and rooted networks, respectively). It is known that the quarnets coming from a semi-directed level-1 network uniquely characterize the network (Huber et al. 2024) and that theoretically they can be puzzled together efficiently to reconstruct the network (Frohn et al. 2024). However, a set of quarnets stemming from real data will unavoidably contain erroneous quarnets, thus creating the need for a more robust algorithm.

In this paper we introduce SQUIRREL (Semi-directed Quarnet-based Inference to Reconstruct Level-1 Networks): an efficient software tool and algorithm that builds a semi-directed level-1 network from a given full set of quarnets (that is, a *dense* set that contains one quarnet for each subset of four taxa). We complement SQUIRREL with a fast heuristic method to construct quarnets from sequence data: the δ-heuristic (see the Materials and Methods for a formal description). Note that various existing algorithms and programs can be used to infer level-1 networks (both rooted and semi-directed) from biological data that are based on alternative approaches. For example, PHYLONET (Than et al. 2008; Yu and Nakhleh 2015), SNAQ (Solís-Lemus and Ané 2016; Solís-Lemus et al. 2017) and PHYNEST (Kong et al. 2024) are all software tools using likelihood-based algorithms operating under a coalescent model. SNAQ builds semi-directed networks, whereas both PHYNEST and PHYLONET focus on rooted networks. These methods assume an upper bound on the number of reticulate events and either take gene trees (PHYLONET and SNAQ) or sequence data (PHYNEST) as input, after which they perform a potentially time-consuming search through the space of networks to optimize a likelihood criterion. On the other hand, NANUQ (Allman et al. 2019) and the recent extension NANUQ$^+$ (Allman et al. 2024a) do not employ a likelihood-framework and instead use concordance factors on four-taxon subsets to produce a semi-directed level-1 network up to contracting triangles (3-cycles) and identifying the locations of reticulations in 4-cycles. This approach is faster but requires other methods to compute the input gene trees first, which itself can be a challenging step (Chifman and Kubatko 2014; Simmons and Gatesy 2015; Zhang and Mirarab 2022; Steenwyk et al. 2023). Other approaches use Bayesian methodology to construct rooted networks (e.g. SPECIESNETWORK (Zhang et al. 2018a)) but are not yet able to scale to larger data sets. Lastly, LEV1ATHAN (Huber et al. 2010) and TRILONET (Oldman et al. 2016) take a combinatorial stance towards the network construction problem; they take as input a set of rooted three-leaf trees (LEV1ATHAN) or rooted three-leaf networks (TRILONET) and output a rooted level-1 network, with TRILONET including a heuristic to generate rooted three-leaf networks from sequence data.

We now present a brief overview of how SQUIRREL works; a formal description of the algorithm (plus supporting figures) is given in the Materials and Methods section. As with NANUQ and to a lesser extent SNaQ, Squirrel constructs networks up to the contraction of triangles (see Figure 1(b)), thus resulting in a binary triangle-free semi-directed level-1 network (i.e. a network with no cycles that contain just three vertices). Since triangles are relatively difficult to infer correctly (Gross et al. 2021), Squirrel does not use the location of any triangles in the quarnets and instead only employs *tf-quarnets* (triangle-free quarnets; see Figure 1(c)). As shown in Frohn et al. (2024), by considering tf-quarnets, we still maintain enough information to theoretically construct the complete semi-directed level-1 network up to contracting its triangles. If quarnets with triangles are given in the input, tf-quarnets are obtained by contracting the triangles. Hence, each tf-quarnet is either a quartet tree or contains a 4-cycle.

Given a dense set of weighted tf-quarnets, Squirrel first uses all of the tf-quarnets that are quartet trees to build a sequence of non-binary phylogenetic trees, using an algorithm from Berry and Gascuel (2000) and employing techniques from the QuartetJoining algorithm (Grünewald et al. 2009) that constructs phylogenetic trees from quartet trees. Within each of the non-binary phylogenetic trees in the sequence, the internal vertices with high degree are replaced by a suitable cycle. In particular, Squirrel repeatedly solves the Travelling Salesman Problem (TSP, see e.g. Bellman 1962; Held and Karp 1962) with suitably defined distances to create a cyclic ordering of the subnetworks around the cycles. This results in a sequence of candidate level-1 networks, from which Squirrel returns the one that agrees, in a well-defined sense, with most of the original tf-quarnets. If an outgroup is specified, this network can in turn be transformed into a rooted network.

We emphasize that any method that is able to create a dense set of tf-quarnets from biological data (possibly incorporating e.g. incomplete lineage sorting) could be used to generate input for Squirrel. Furthermore, Squirrel takes into account weights the tf-quarnets might have, which can be used to model confidence or bootstrap support. Reassuringly, Squirrel is consistent in the sense that it will reconstruct the correct network if all tf-quarnets are derived from a triangle-free semi-directed level-1 network, a fact that we prove in Theorem 1 in the Materials and Methods section.

Results

Simulation study

Following the simulation studies for Lev1athan (Huber et al. 2010) and TriLoNet (Oldman et al. 2016), we analyze what effect noise in a set of tf-quarnets has on the performance of Squirrel. To this end, we generate 100 random triangle-free semi-directed level-1 networks for every number $n \in \{10, 15, 20, 25, 30, 35\}$ of leaves (see Section B of the Supplementary Material for the generating algorithm). For each network $\mathcal{N}$, the reticulation number $r(\mathcal{N})$ (i.e. the number of reticulations) is chosen uniformly at random from $\{0, \dots, \lfloor n/3 \rfloor\}$. This results in a set of 600 random networks $\mathcal{N}$, each inducing a set $\mathcal{Q}(\mathcal{N})$ of tf-quarnets. For each network $\mathcal{N}$ and each perturbation ratio $\varepsilon \in \{0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$, we create a noisy set of tf-quarnets $\mathcal{Q}_\varepsilon(\mathcal{N})$ by changing the undirected underlying topology of a fraction of the tf-quarnets uniformly at random which is given by ε. Then, if this creates a 4-cycle, we pick a random location for the reticulation. We use this scheme for the creation of noise to prevent 4-cycles from only changing their reticulation and keeping their circular ordering. Such a perturbation will barely influence the output of the algorithm, since reticulations of 4-cycle tf-quarnets are only used to determine the location of reticulations in 4-cycles of the final networks. The resulting $5400 = 600 \cdot 9$ sets of unweighted tf-quarnets $\mathcal{Q}_\varepsilon(\mathcal{N})$ are used as input for Squirrel. The average computation times ranged from below a second for the networks with the fewest leaves to below two minutes for the networks with 35 leaves.

To measure how well Squirrel reconstructs the original networks from these noisy tf-quarnet sets we compute two similarity scores for every input network $\mathcal{N}$ and output network $\mathcal{M}$. The first score is the *tf-quarnet consistency score* (modeled after a similar score in Huber et al. (2010) and Oldman et al. (2016)) which is defined as

$$C(\mathcal{N}, \mathcal{M}) = \frac{|\mathcal{Q}(\mathcal{N}) \cap \mathcal{Q}(\mathcal{M})|}{|\mathcal{Q}(\mathcal{N})|}. \tag{1}$$

This score measures what fraction of the tf-quarnets induced by $\mathcal{N}$ are also induced by the constructed network $\mathcal{M}$. We also consider its symmetric counterpart: the *tf-quarnet symmetric consistency score*, defined as

$$S(\mathcal{N}, \mathcal{M}) = \frac{|\mathcal{Q}(\mathcal{N}) \cap \mathcal{Q}(\mathcal{M})|}{|\mathcal{Q}(\mathcal{N}) \cup \mathcal{Q}(\mathcal{M})|}. \tag{2}$$

Both scores are always in the interval $[0, 1]$ and attain a value of 1 if and only if $\mathcal{N} = \mathcal{M}$, which follows from Frohn et al. (2024). The boxplots in Figure 2 show the distribution of the two scores for different perturbation ratios ε and leaf set sizes n. As expected, both scores decrease for larger values of ε. However, the decrease seems fairly limited, with both consistency scores averaging above 0.91 even for sets containing only 50% of the original tf-quarnets.

To investigate in what way noise in a set of tf-quarnets influences the structure of the reconstructed networks, we compute the difference in the reticulation numbers $r(\mathcal{N}) - r(\mathcal{M})$ between the input networks $\mathcal{N}$ and output networks $\mathcal{M}$. The boxplots in Figure 3 show the result of this experiment, again for different values of ε and n. Up to a value of $\varepsilon = 0.1$, Squirrel reconstructs networks with the correct reticulation number in almost all cases. For higher values, the differences are more spread out, while the average difference slowly becomes positive. Thus, it seems that Squirrel slightly favors networks with fewer reticulations for high values of ε, although the average absolute differences remain below a reasonably small 1.5. A possible explanation could be that by not considering triangles in the quarnets, the signal in the data indicating reticulate events is weakened.

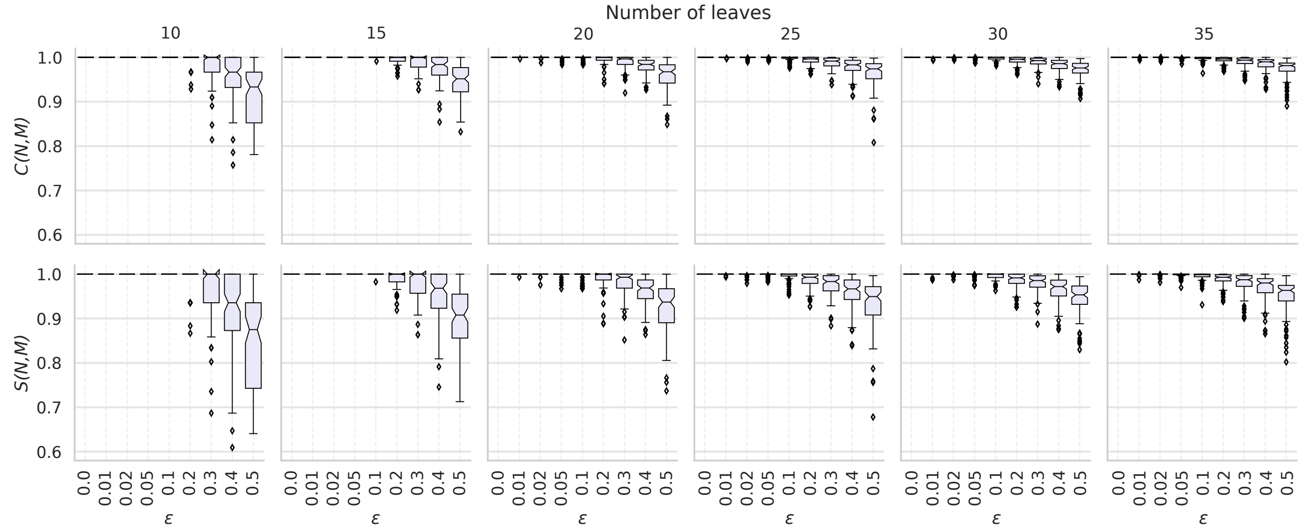


Figure 2: Boxplots showing the spread of C- and S-scores between the input network $\mathcal{N}$ and output network $\mathcal{M}$, when applying SQUIRREL to sets of tf-quartets with leaf set sizes n and perturbation ratios ε. The boxplots show the quartiles of the data and its outliers. A single outlier in the case of $n = 10$ and $\varepsilon = 0.5$ has a C- and S-score below 0.6 and is omitted from the figure for clarity.

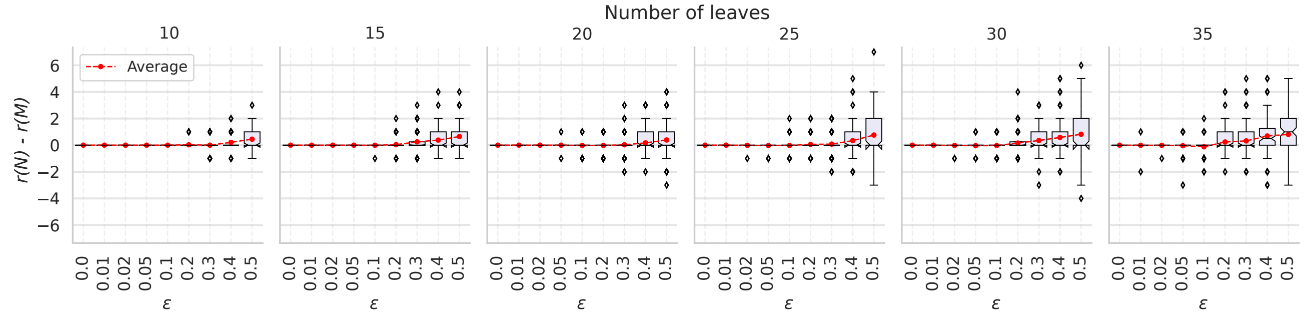


Figure 3: Boxplots showing the variation of the difference in reticulation number $r(\mathcal{N}) - r(\mathcal{M})$ of the input network $\mathcal{N}$ and output network $\mathcal{M}$, when applying SQUIRREL to sets of tf-quartets with leaf set sizes n and perturbation ratios ε. The boxplots show the quartiles of the data, its outliers and the averages in red.

We also perform a study with simulated nucleotide sequences to test the performance of the δ-heuristic combined with SQUIRREL, using a similar approach to the simulations presented in Holland et al. (2002) and Oldman et al. (2016). For each of our 600 previously generated networks, we simulate one multiple sequence alignment (MSA) for every sequence length $k \in \{1\,\text{kbp}, 10\,\text{kbp}, 100\,\text{kbp}, 1\,\text{Mbp}\}$ as follows. Briefly, we first root every semi-directed network $\mathcal{N}$ uniformly at random on some edge (making sure that it is a valid root-location) to create a rooted phylogenetic network. We then use the software tool SEQ-GEN (Rambaut and Grass 1997) to simulate MSAs of equal length along all displayed trees of the rooted phylogenetic network under the K2P model with transition-transversion bias 4 (as in Holland et al. 2002; Oldman et al. 2016). The MSAs of the displayed trees are then concatenated to create one MSA with the desired length k. Since our δ-heuristic treats every site of the MSA independently, this way of generating MSAs is asymptotically equivalent to generating MSAs under the K2P network-based Markov model with reticulation parameters of 0.5 (see e.g. Gross et al. 2021).

The branch lengths (i.e. the expected number of substitutions along each edge) that are used for the simulations are determined as follows. Given an edge (u, v) of one of the rooted phylogenetic networks, we let $p_{(u,v)}$ be the average length (in terms of number of edges) of all unique paths from the root to any leaf that contain the edge (u, v). Then, we assign the edge (u, v) a branch length of $0.3/p_{(u,v)}$, which ensures that every path in the network from a root to a leaf roughly has a total length of 0.3, as is the case in the simulations by Holland et al. (2002) and Oldman et al. (2016).

We then use the $2400 = 600 \cdot 4$ simulated MSAs as input for our δ-heuristic to construct dense sets of weighted tf-quarnets, which are in turn used to construct semi-directed networks with SQUIRREL. As before, we compare every constructed semi-directed network $\mathcal{M}$ with the original semi-directed network $\mathcal{N}$ in terms of C-score, S-score and difference in reticulation number $r(\mathcal{N}) - r(\mathcal{M})$. The results are depicted in Figure 4 and Figure 5, respectively. We observe that both consistency scores increase as the sequence length changes from 1 kbp to 10 kbp. Additionally, both the average and the variation of the difference in reticulation number decrease. Interestingly, the increase of the sequence length from 10 kbp to 100 kbp or 1 Mbp does not seem to have much further effect. As was the case in our previous experiment, an increase in the number of leaves n of the original semi-directed network improves the two considered consistency scores, yet also results in a greater spread of the difference in reticulation number between the original and constructed network. The latter

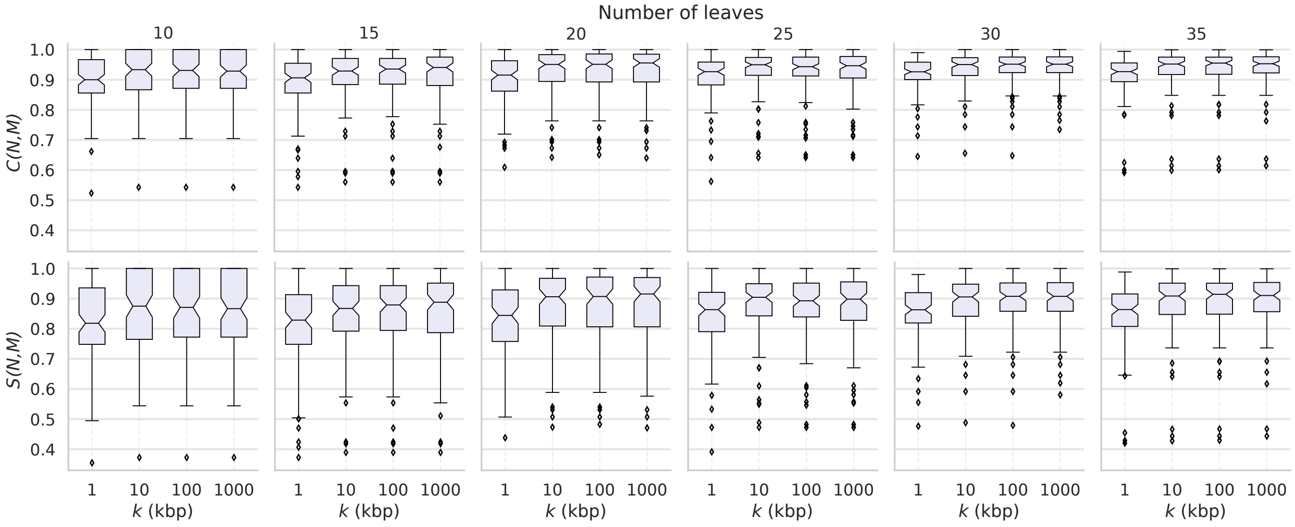


Figure 4: Boxplots showing the spread of C- and S-scores between the input network $\mathcal{N}$ and output network $\mathcal{M}$, when applying the δ-heuristic and Squirrel to MSAs with leaf set sizes n and sequence lengths k. The boxplots show the quartiles of the data and its outliers.

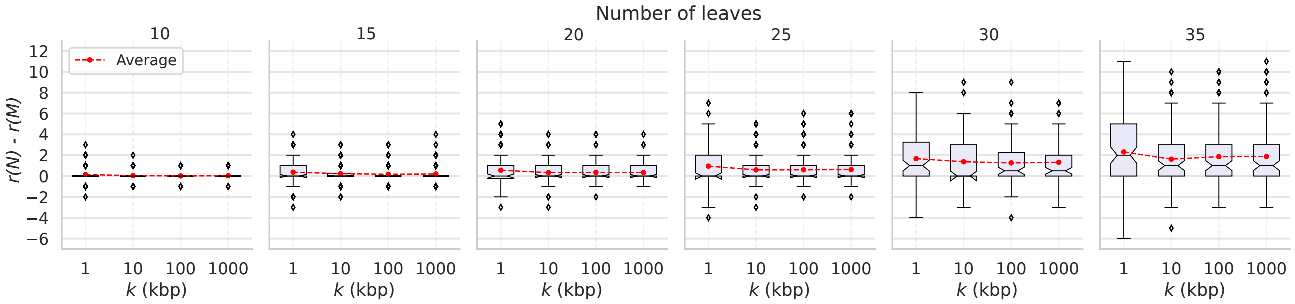


Figure 5: Boxplots showing the variation of the difference in reticulation number $r(\mathcal{N}) - r(\mathcal{M})$ of the input network $\mathcal{N}$ and output network $\mathcal{M}$, when applying the δ-heuristic and Squirrel to MSAs with leaf set sizes n and sequence lengths k. The boxplots show the quartiles of the data and its outliers.

point can be explained by the fact that smaller networks simply allow for fewer reticulations, thus also bounding the largest possible difference in reticulation number.

Biological data

To illustrate the applicability of Squirrel to biological data, we consider three data sets on groups of taxa with evidence of secondary contact in their evolutionary histories: a large set of tf-quarnets generated with the MML algorithm from Martin et al. (2023) (named after the authors), a short multiple sequence alignment on few taxa from Salemi and Vandamme (2003), and a long multiple sequence alignment on many taxa from Vanderpool et al. (2020).

Xiphophorus We first test the applicability of Squirrel to a set of tf-quarnets that was generated with the MML algorithm (Martin et al. 2023). For each four-taxon subset, this algorithm creates a ranking of the possible 4-cycles according to some scoring criterion (with the lowest score being the best). Based on the scores it either detects a quartet tree (which we give a weight of 1), or it chooses the best 4-cycle, which we give a weight of $\min(1, s_2/s_1 - 1)$, where s_1, s_2 are the two lowest (and thus best) scores. In this manner we take into account how close the scores for the two best scoring 4-cycles are.

The data set we consider contains 14,950 weighted tf-quarnets on a set of 25 swordtail fish and platyfish (genus *Xiphophorus*) and the single outgroup *Pseudoxiphophorus jonesii*. This genus has been widely studied and much evidence has been presented for widespread hybridization within the genus (see e.g. Rosenthal et al. 2003; Culumber et al. 2011; Cui et al. 2013; Kang et al. 2013; Schumer et al. 2013; Solís-Lemus and Ané 2016, and the references therein), making it difficult to capture the full evolutionary history. Traditionally, the genus is divided into four major lineages: northern swordtails, southern swordtails, northern platyfishes and southern platyfishes (Meyer et al. 2006; Cui et al. 2013). The best network generated by Squirrel (taking less than two minutes) had a weighted tf-quarnet consistency score of 0.974 and is shown in Figure 6. However, many of the other candidate networks had scores that were very close to the score of the best scoring network.

Since the weighted tf-quarnet consistency score measures how consistent the network is with the tf-quarnets, taking their weights into account (see eq. (3) in the Materials and Methods), it should be noted that a weighted consistency score close to 1 does not necessarily imply a close to 100% level of confidence that the network is correct. Instead, it reflects whether the quarnets with high weight (i.e. high confidence in their correctness) are consistent with the constructed network, making it most

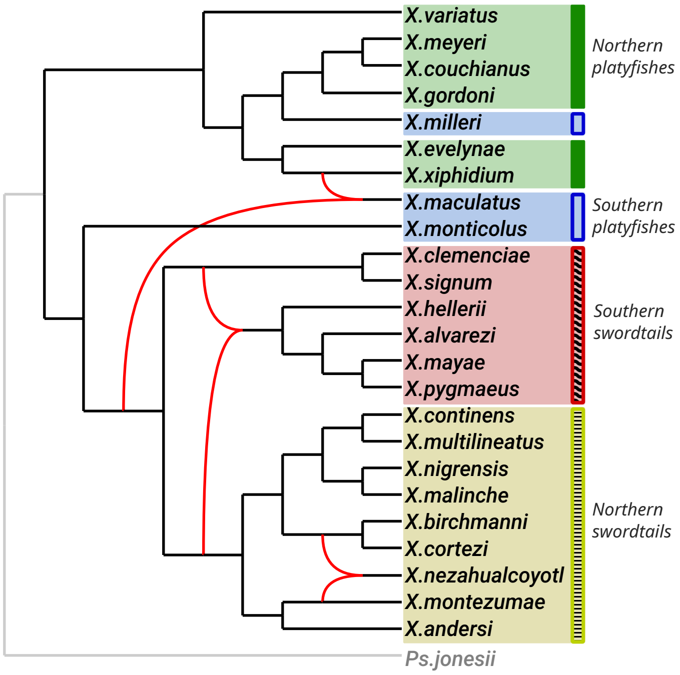


Figure 6: Phylogenetic network inferred by SQUIRREL from a dense set of weighted tf-quarnets on the genus *Xiphophorus* (generated from a multiple sequence alignment with the MML algorithm from Martin et al. (2023)). The four major lineages are indicated by the different shaded areas. The reticulation edges are curved, while the edges leading to the outgroup *Pseudoxiphophorus jonesii* are in grey.

useful as a relative measure to assess if there is a clear best network or if multiple networks perform similarly well. In contrast, the unweighted consistency score (see eq. (1)) can be more easily interpreted as an absolute measure of performance, but it may discard useful information about quarnet confidence if such information is available. A more statistically sound way to generate weights for the tf-quarnets inferred with the MML algorithm from Martin et al. (2023) (similar to the bootstrap support in Barton et al. (2022)) would possibly increase the confidence of SQUIRREL in a single best network. Hence, we would welcome further research efforts into computing confidence scores for inferred tf-quarnets which can be used as input weights for SQUIRREL.

The constructed network clearly divides the three major *Xiphophorus* clades (northern swordtails, southern swordtails and platyfishes) but similar to other studies (Meyer et al. 2006; Cui et al. 2013) intertwines northern and southern platyfishes. Our network has one reticulation edge involving an ancestor of both the northern and the southern swordtails. Another reticulate event places the northern swordtail *X.cortezi* both as a sibling of *X.nezahualcoyotl* and of the clade (*X.malinche*, *X.birchmanni*). This reticulate event aligns with previous work in Cui et al. (2013), where the precise placement of *X.cortezi* within this subset of the species (including *X.montezumae*) was also uncertain and depended on the inference methods used. Furthermore, one of the subtrees displayed in our network for this subset of the species (i.e. the subtree that includes *X.montezumae*) is the same as the subtree of the network inferred by SNAQ (Solís-Lemus and Ané 2016; Solís-Lemus et al. 2017). The last reticulate event involves the southern platyfish *X.maculatus*, for which Cui et al. (2013) report difficulties placing it in the mitochondrial DNA tree. Judging from the many different inferred networks and possible reticulate events (see again Rosenthal et al. 2003; Culumber et al. 2011; Cui et al. 2013; Kang et al. 2013; Schumer et al. 2013; Solís-Lemus and Ané 2016), capturing the evolutionary history of the complete genus as a level-1 network might be too much to ask for because the truth may not be level-1. As an example, evolutionary histories containing many hybridization events between more distantly related species (such as horizontal gene transfer) can not always be captured well by a level-1 network, since such events often result in complex networks with many nested reticulation events (see e.g. Soucy et al. 2015, Fig. 5).

HIV We now consider a multiple sequence alignment (MSA) of the HIV-1 virus data set containing 9 sequences of length 9,953 bp which first appeared in Salemi and Vandamme (2003). This data set is well-studied (Lemey et al. 2009; Huber et al. 2010; Oldman et al. 2016) and contains sequences of the HIV-1 M-group subtypes *A*, *B*, *C*, *D*, *F*, *G*, *H* and *J* as well as a sequence for *KAL153* which is believed to be a recombinant of subtypes *A* and *B* (see Lemey et al. 2009, Ch. 16). We use our δ-heuristic (formally described in the Materials and Methods) to obtain a weighted set of tf-quarnets from the MSA and then apply SQUIRREL to construct a network, which we root using the outgroup *C* (as in Salemi and Vandamme 2003; Huber et al. 2010). The δ-heuristic and SQUIRREL constructed a clear best scoring network (shown in Figure 7(a)) with a weighted tf-quarnet consistency of 0.58 within one second.

Indeed, SQUIRREL, combined with the δ-heuristic, is able to identify *KAL153* as a recombinant of subtypes *A* and *B*, agreeing with the analysis in (Lemey et al. 2009, Ch. 16). This compares favourably to TRILONET (Old-

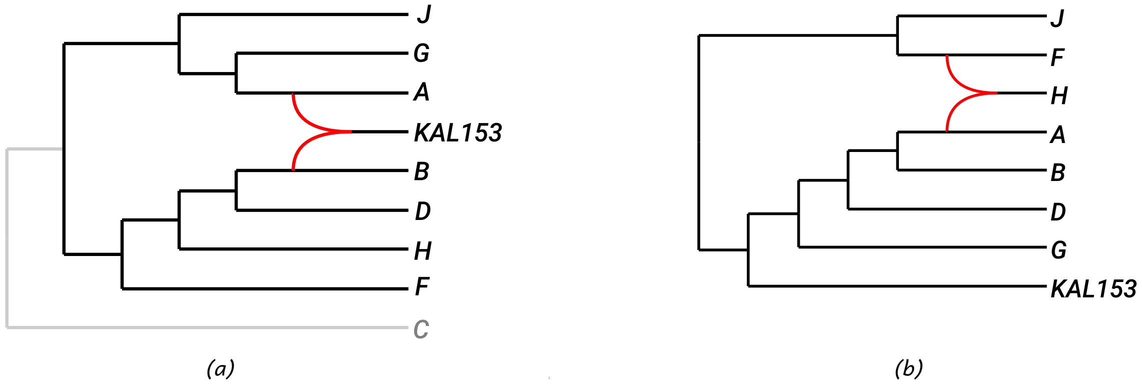


Figure 7: (a): Phylogenetic network inferred by SQUIRREL (using the δ-heuristic to create tf-quarnets) from a multiple sequence alignment of the HIV-1 data set under consideration. The reticulation edges are curved, while the edges leading to the outgroup C are in grey. (b): Phylogenetic network inferred by TRILONET (Oldman et al. 2016) on the same HIV-1 data set (without the outgroup C), again with curved reticulation edges.

man et al. 2016), where the subtype H was identified as a recombinant (see the constructed network in Figure 7(b)). LEV1ATHAN (Huber et al. 2010) was able to identify *KAL153* as a recombinant, but it relies on other algorithms to make the step from sequences to gene trees.

Primates To investigate the performance of SQUIRREL and the δ-heuristic on data sets with many taxa and long sequences, we consider an MSA from Vanderpool et al. (2020) of length 1,761,114 bp that contains concatenated sequences for 26 primate species, 2 closely related non-primate species and the outgroup *Mus musculus*. We first apply the δ-heuristic to the MSA to obtain a set of 23,751 weighted tf-quartets. Subsequently, we use SQUIRREL (specifying *Mus musculus* as the outgroup to root it) and obtain the tree in Figure 8(a) after a few minutes on a standard laptop. The tree coincides exactly with the species tree obtained in Vanderpool et al. (2020) using the gene tree based algorithm ASTRAL III (Zhang et al. 2018b), while largely agreeing with two previously inferred phylogenies (Perelman et al. 2011; Springer et al. 2012). The weighted tf-quarnet consistency score of the tree is 0.995, but some of the other generated candidate networks (which contain reticulations) have scores within 0.003 from this best value, suggesting that reticulate events might have occurred.

We investigate this further by looking only at the 8 primates in the *Cercopithecinae* subfamily, for which Vanderpool et al. (2020) have demonstrated possible reticulate events. Combining the δ-heuristic and SQUIRREL we generated a set of candidate networks for these 8 species and the outgroup *Colobus angolensis pallatus*. Two of the networks had a much higher score than the others and they only differed from each other by the addition of a reticulation edge. In particular, the second best scoring network (shown in Figure 8(b)) had a score of 0.956, while the best scoring network was the subtree of the original network with score 0.974 (also shown in Figure 8(b), by ignoring the curved reticulation edge). The *blobtree* of the network (obtained by contracting the cycle into a single node) exactly matches one of the blobtrees inferred with TINNIK (Allman et al. 2024b). The particular reticulate event we found was not reported in Vanderpool et al. (2020). However, our reticulate event might be more probable since it is between species in the same continent (Africa), while the study by Vanderpool et al. (2020) mentions possible reticulate events between species on different continents (Asia and Africa). Lastly, Vanderpool et al. (2020) found evidence for a “complex pattern of ancient introgression” (p. 14) within the subfamily and state that roughly 40% of the species within the subfamily are known to hybridize (Tung and Barreiro 2017), which suggests that the true nature of the subfamily might not be well-represented by a level-1 network. This is further supported by the fact that the analysis done in Vanderpool et al. (2020) with PHYLONET (Than et al. 2008; Yu and Nakhleh 2015) and SNAQ (Solís-Lemus and Ané 2016; Solís-Lemus et al. 2017) also gave ambiguous results, while PHYNEST (Kong et al. 2024) yet again concludes with a different network.

The *Cercopithecinae* subfamily (again with outgroup *Colobus angolensis pallatus*) also featured in Barton et al. (2022) in the context of using the QNR-SVM algorithm for inferring quarnets from a data set. The reason for restricting to a subset was stated as the lack of an algorithm that puzzles together many quarnets. Instead, the authors puzzle them together by hand to obtain a network with a single reticulation that induces 81% of the well-supported quarnets. Using their quarnet weighting scheme, SQUIRREL was able to identify a tree inducing 85% of the well-supported quarnets. (Here, we used a variation of SQUIRREL that takes into account the triangles of the quarnets to choose the best scoring network, instead of the default of just focusing on the tf-quarnets). Therefore, SQUIRREL might be a viable tool to puzzle together quarnets obtained with an algorithm such as QNR-SVM, while still being able to scale to larger data sets unfit for resolving conflicting quarnets by hand.

Discussion

We have introduced SQUIRREL: a combinatorially consistent algorithm that can puzzle together a dense set of quarnets to create a semi-directed level-1 network. In addition, when combined with the model-based method QNR-SVM (Barton et al. 2022) or the MML algorithm (Martin et al. 2023) for inferring quarnets, SQUIRREL provides a method to create a level-1 network directly from

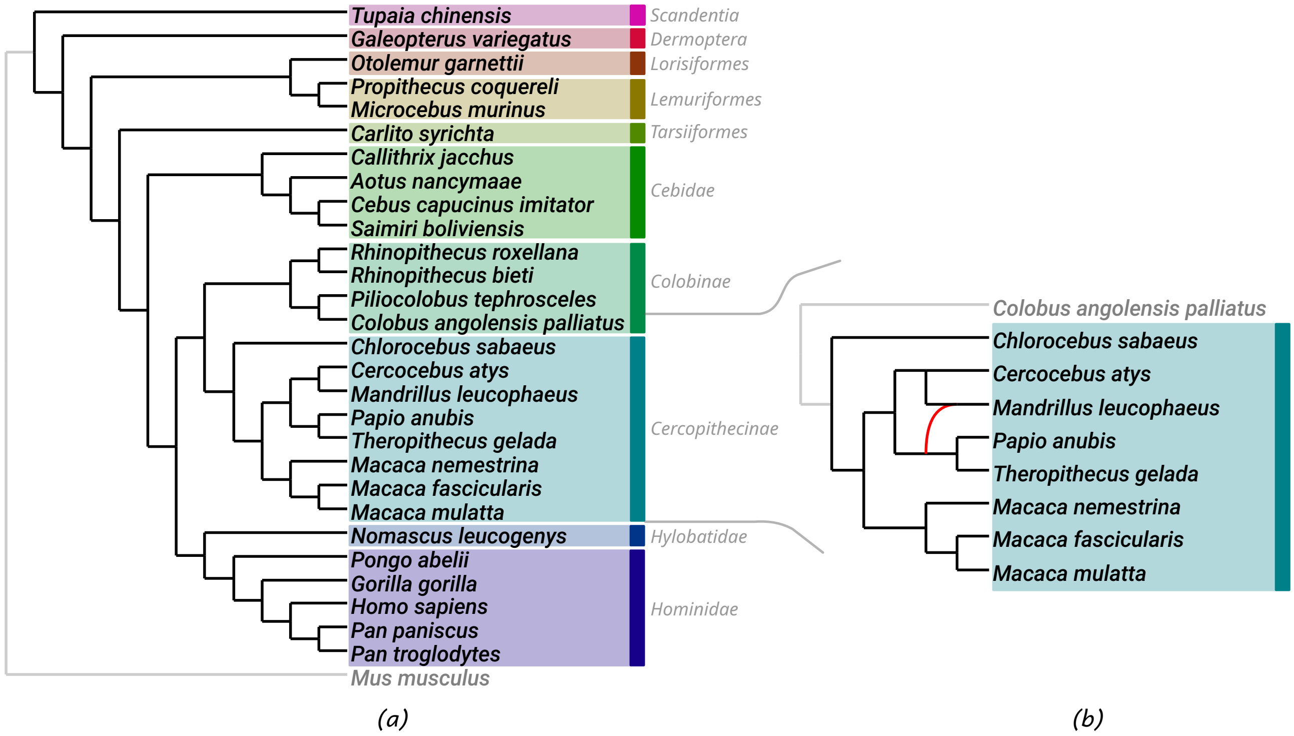


Figure 8: (a): Phylogenetic tree inferred by SQUIRREL (using the δ-heuristic to create tf-quarnets) from a multiple sequence alignment of the primate data set under consideration, with the edges leading to the outgroup *Mus musculus* in grey. The different shaded areas indicate different taxonomical groups as they appear in Vanderpool et al. (2020). The two non-primate species are *Tupaia Chinensis* and *Galeopterus variegatus*. (b): In conjunction with the δ-heuristic to create tf-quarnets, SQUIRREL inferred two networks with very close weighted tf-quarnet consistency scores from the considered multiple sequence alignment of the subfamily of *Cercopithecinae* (using *Colobus angolensis palliatus* as outgroup). One of them is the depicted network and the other is the phylogenetic tree obtained from that network by ignoring the curved reticulation edge.

sequence data. To the best of our knowledge, SQUIRREL is one of the first methods that allows the construction of semi-directed level-1 networks from biological data using collections of quartets. The only other approaches we are aware of that use quarnet information are NANUQ (Allman et al. 2019) and the recently presented NANUQ$^+$ (Allman et al. 2024a). Although NANUQ$^+$ uses a similar distance-based strategy to SQUIRREL to expand the cycles in a network, both NANUQ and NANUQ$^+$ take as input a collection of gene trees, rather than a dense set of quarnets or a sequence alignment.

Any method that creates a dense set of quarnets from biological data could be used as input for SQUIRREL. In particular, if such a method is statistically consistent under some model (possibly incorporating e.g. incomplete lineage sorting), the combinatorial consistency of SQUIRREL ensures that the combined inference is consistent as well. Furthermore, SQUIRREL could in principle be combined with methods that may not scale well to larger taxa sets but are still able to construct partial semi-directed level-1 networks (containing some but not all of the studied taxa) from biological data. Indeed, as with supertree methods, partial networks on larger sets of taxa could be converted to quarnets for SQUIRREL by restricting those partial networks to four taxa. This would require a rule to decide what to do in case partial networks overlap on more than four taxa and they induce conflicting quarnets. Hence, a possible direction for future research would be adapting SQUIRREL to work with non-dense sets of quarnets which could contain any number of quarnets for each subset of four taxa.

Using the δ-heuristic, SQUIRREL is able to quickly construct a level-1 network directly from sequence data. Our sequence simulations show that the δ-heuristic is likely not statistically consistent under the tested K2P model. In particular, an increase in sequence length beyond 10 kbp does not give a visible improvement under our simulation settings, which one would expect for a statistically consistent quarnet inference method. Despite the lack of a statistical basis of the δ-heuristic, it already shows promising similarity scores for multiple sequence alignments with a length of 1 kbp when combined with SQUIRREL. Furthermore, a major advantage is its speed. As an example, this approach was able to construct a network with 29 taxa from a multiple sequence alignment of length 1.7 Mbp within a few minutes on a standard laptop (see the Results section). Hence, we do not see the δ-heuristic (combined with SQUIRREL) as an alternative for known model-based methods, but rather as a complementary tool. For one, this approach can be used to generate reasonable starting networks for the time-intensive search through the network-space of likelihood-based methods (such as PHYLONET (Than et al. 2008; Yu and Nakhleh 2015), SNAQ (Solís-Lemus and Ané 2016; Solís-Lemus et al. 2017) and PHYNEST (Kong et al. 2024)). On the other hand, it can be used to quickly gain insight into sequence data without the need to first infer gene trees with a different tool, as is the case for NANUQ (Allman et al. 2019), which requires many accurate gene trees to make a good estimate of the concordance factors.

With the increasing availability of genome and transcriptome data, biologists are also likely to explore the reconstruction of separate phylogenetic networks for multiple sets of short orthologous sequences. Rapid construction of such networks for the same set of taxa across different sets of orthologues opens up the possibility for comparative analyses. A possible research direction in this area would be to combine SQUIRREL's speed for constructing semi-directed level-1 networks with the tf-quarnet consistency score or the recently introduced dissimilarity measure for semi-directed networks that generalizes the widely-used Robinson-Foulds distance for phylogenetic trees (Maxfield et al. 2024), which would permit the rapid comparison of networks computed for different sets of orthologues. It also leads to the interesting problem of finding a consensus of a collection of semi-directed networks, which to our best knowledge has not yet been addressed in the literature. One approach to this problem could be to treat it as a supernetwork question where all input networks have the same leaf set, and use the approach suggested earlier in this section.

Our simulations indicate that SQUIRREL can construct networks closely resembling an underlying network in terms of tf-quarnets, even if many of the tf-quarnets are wrongly inferred. In particular, both of the considered consistency scores average above 0.91 even for sets containing only 50% of the original tf-quarnets. This is a significant improvement compared to a similar experiment to the triplet/trinet-based LEV1ATHAN and TRILONET algorithms, where the *trinet consistency score* (the rooted three-leaf network analogue of our tf-quarnet consistency score) dropped below 0.5 for sets still containing 75% of the trinets (Oldman et al. 2016). These results can be considered as evidence that SQUIRREL is able to construct networks with a high topological resemblance to the original network in terms of tf-quarnets, even for a high percentage of incorrect tf-quarnets. As mentioned in the Results section, even though the tf-quarnets are theoretically enough to construct a triangle-free semi-directed level-1 network, in practice, contracting the triangles might somewhat weaken the signal of reticulation events. Note that theoretically (that is, when all quarnets come from a single network with n leaves) only $\mathcal{O}(n \log n)$ tf-quarnets are required to reconstruct the network, instead of the full set of $\mathcal{O}(n^4)$ tf-quarnets (Frohn et al. 2024). Thus, even sets with many incorrect tf-quarnets might still hold enough information to reconstruct the original network. This could also explain why a higher number of leaves seems to have a positive effect on the similarity score: $\mathcal{O}(n \log n)$ grows slower than $\mathcal{O}(n^4)$, so the fraction of tf-quarnets necessary to reconstruct a network decreases when n grows.

Although several methods can construct semi-directed level-1 networks, the assumption that a network is level-1 might be too restrictive in many cases for biological data. A major breakthrough would be to develop a practical algorithm that is able to construct networks that are more complex than level-1 networks. Some theoretical results have already appeared towards tackling this problem. For example, it is known that semi-directed level-2 networks are uniquely encoded by the quarnets they induce (Huber et al. 2024). In addition, under several models, the circular ordering around the blobs of outerlabeled planar networks (a class of semi-directed networks more general than semi-directed level-1 networks) is also shown to be identifiable (Rhodes et al. 2025). Furthermore, the recently introduced TINNIK algorithm (Allman et al. 2024b) uses concordance factors computed from gene trees to construct the blobtree of networks with arbitrary level under the network multispecies coalescent model. Although such a blobtree still remains a tree, it does indicate in what areas of the underlying network reticulations may have occurred. It might also be worth looking for an extension of SQUIRREL to non-binary networks, where high-degree vertices are allowed which do not necessarily represent reticulate events.

In conclusion, SQUIRREL provides an efficient and combinatorially sound approach for reconstructing semi-directed level-1 networks from dense sets of quarnets. The promising consistency scores achieved in our tests underscore SQUIRREL's ability to retain network topology even when faced with noisy data. Together with our δ-heuristic, SQUIRREL allows rapid insight into large-scale sequence data. Looking forward, we hope that this approach can complement more time-intensive methods and support the preliminary exploration of network hypotheses.

Materials and Methods

We start this section by presenting formal definitions surrounding phylogenetic networks and quarnets in the first subsection. The high level idea of SQUIRREL is described in the second subsection, while its subroutines are formalized in the third and fourth subsection. We end with the description of the δ-heuristic in the fifth subsection, and a brief description of the consistency and implementation of SQUIRREL in the sixth and seventh subsection, respectively.

Phylogenetic networks and quarnets

Phylogenetic networks A *rooted phylogenetic network* on a set of at least four leaves $\mathcal{X}$ (representing a set of taxa) is a directed graph with a single root, no parallel edges and no directed cycles such that (i): the root has two children; (ii): each leaf (i.e. a vertex with no children) has one parent and is uniquely labeled by an element from $\mathcal{X}$; (iii): all other vertices either have one parent and two children, or two parents and one child. A vertex of the latter type is a *reticulation (vertex)*, and the two edges directed towards it are *reticulation edges*. See Figure 9(a) for an example. *Semi-directed phylogenetic networks*, the type of network this paper is concerned with, can be obtained from a rooted phylogenetic network by suppressing its root and undirecting all edges except for the reticulation edges. For the sake of brevity, we refer to these networks simply as *semi-directed networks*. Since the reticulation edges remain directed, we can still refer to the reticulation vertices and edges of a semi-directed network (see Figure 9(b)). We call a semi-directed network *triangle-free* if it does not contain any triangles (3-cycles). Note that a semi-directed network without any reticulations is an (unrooted) phylogenetic tree in the usual sense.

In this paper, we consider semi-directed networks which are *level-1* (again see Figure 9(b)), meaning that every

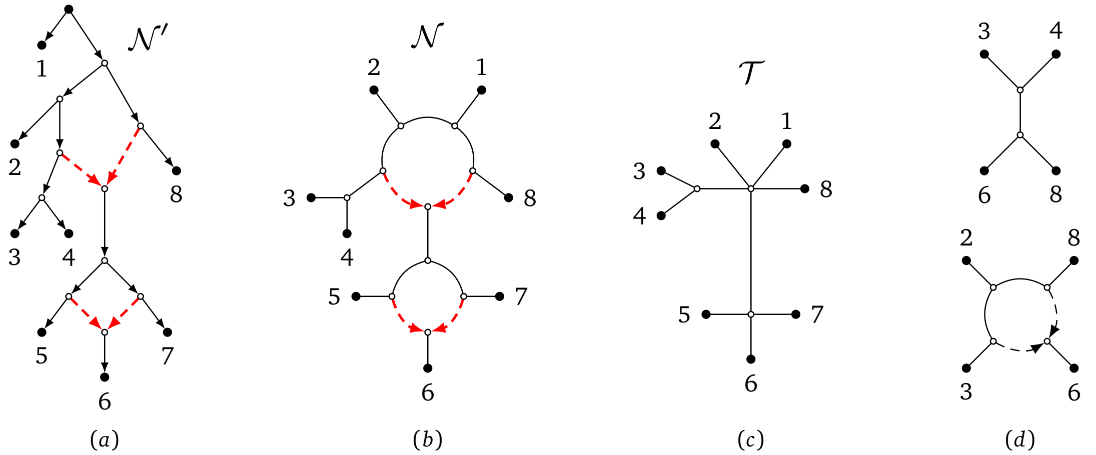


Figure 9: (a): A rooted phylogenetic level-1 network $\mathcal{N}'$ on leaf set $\mathcal{X} = \{1, \dots, 8\}$, with the dashed reticulation edges pointing towards its reticulation vertices. (b): The triangle-free semi-directed level-1 network $\mathcal{N}$ which can be obtained from $\mathcal{N}'$ by suppressing its root and keeping only the dashed reticulation edges directed. (c): The blobtree $\mathcal{T}$ of the semi-directed network $\mathcal{N}$, obtained by collapsing all cycles into single vertices. (d): Two of the tf-quarnets induced by $\mathcal{N}$. When ignoring the leaf labels, these are the two possible tf-quarnet shapes. The top tf-quarnet is a quartet tree and the bottom tf-quarnet is a 4-cycle.

reticulation is part of exactly one undirected cycle (ignoring the directions of the reticulation edges). The (possibly non-binary) phylogenetic tree obtained by collapsing every such cycle into a single vertex is called the *blobtree* (or *tree of blobs*) of the semi-directed network (see Figure 9(c)).

Given a semi-directed network $\mathcal{N}$ on $\mathcal{X}$, a partition $A|B$ of $\mathcal{X}$ (with A and B both non-empty) is a *split* of $\mathcal{N}$ if there exists an edge of $\mathcal{N}$ whose removal disconnects the leaves in A from those in B. Such a split is *non-trivial* if the corresponding partition is non-trivial, that is, if $|A|, |B| \geq 2$. As an example, $\{1,2,3,4,8\}|\{5,6,7\}$ is a non-trivial split of the network from Figure 9(b). We sometimes omit the set notation for splits with few elements, meaning that we write $ab|cd$ instead of the split $\{a,b\}|\{c,d\}$ of the set $\{a,b,c,d\}$.

Quarnets A semi-directed network q on a set of four leaves $\mathcal{L}(q) = \{a,b,c,d\}$ is called a *(semi-directed) quarnet*. Recall that up to relabeling the leaves, there are six different level-1 quarnets (see Figure 1(c)). Here, we mostly focus on *tf-quarnets*: triangle-free level-1 quarnets. For a given leaf set $\mathcal{X} = \{a,b,c,d\}$ and up to relabeling of the leaves, there are only two such tf-quarnets on $\mathcal{X}$: the *quartet tree* and the *4-cycle* (see Figure 9(d)). We often denote a quartet tree by its induced split (e.g. $ab|cd$), while we describe a 4-cycle by its circular ordering (e.g. (a,b,c,d)) and mention the leaf below the reticulation separately. Note that tf-quarnets either have no non-trivial split at all, or they have exactly one non-trivial split (e.g. for $\mathcal{X} = \{a,b,c,d\}$ the splits $ab|cd$, $ac|bd$, or $ad|bc$).

SQUIRREL: main algorithm

SQUIRREL uses as input a set $\mathcal{Q}$ of tf-quarnets on some leaf set $\mathcal{X}$ with $n = |\mathcal{X}| \geq 4$. In particular, this set needs to be *dense*, meaning that it contains exactly one tf-quarnet for each subset of four leaves of $\mathcal{X}$ (see also the Introduction). Such a set can be created from a multiple sequence alignment using QNR-SVM (Barton et al. 2022), the MML algorithm (Martin et al. 2023) or our own δ-heuristic (see the fifth subsection of this section). We also allow for a function $w : \mathcal{Q} \to [0,1]$ to give weights to the tf-quarnets, which can e.g. be used to model confidence or bootstrap support. Unweighted tf-quarnets are assumed to have unit weights.

The main idea behind the SQUIRREL algorithm is to first build a sequence of $n-3$ phylogenetic trees on the given n leaves, each one less refined than the other (see Algorithm 2). These trees will function as candidate blobtrees. By expanding all the high degree nodes in these trees into cycles (and introducing reticulations), we obtain a set of semi-directed candidate networks (see Algorithm 3). Finally, out of these networks, we choose the network $\mathcal{N}$ with the highest *weighted tf-quarnet consistency score*, defined as

$$C'(\mathcal{Q}, \mathcal{N}) = \frac{w(\mathcal{Q} \cap \mathcal{Q}(\mathcal{N}))}{w(\mathcal{Q})}. \tag{3}$$

Here, $\mathcal{Q}$ is the input set of tf-quarnets and $\mathcal{Q}(\mathcal{N})$ is the set of tf-quarnets which are induced by the output network $\mathcal{N}$. A tf-quarnet q is *induced* by the network $\mathcal{N}$ if it is the restriction of $\mathcal{N}$ to $\mathcal{L}(q)$, which is formally defined as the network obtained from $\mathcal{N}$ by deleting all leaves not in $\mathcal{L}(q)$ and exhaustively applying the following operations: deleting unlabeled leaves, deleting degree-2 reticulations, suppressing non-reticulate degree-2 vertices, suppressing parallel edges, and suppressing triangles. For completeness, we mention that an induced quarnet can be defined similarly but without suppressing the triangles.

The pseudo-code of SQUIRREL is shown as Algorithm 1. The blobtree construction algorithm (Algorithm 2) and the cycle expansion algorithm (Algorithm 3) are explained in detail in the following two subsections. Even though this is not specified in the pseudo-code, SQUIRREL does allow the user to specify an outgroup as input. Then, it makes sure that all candidate networks can be rooted using this outgroup (see also the fourth subsection of this section).

```
Algorithm 1: SQUIRREL
  Input: dense set $\mathcal{Q}$ of weighted tf-quarnets on
         $\mathcal{X} = \{x_1, \dots, x_n\}$
  Output: triangle-free semi-directed level-1 network
          on $\mathcal{X}$
  $(\mathcal{T}_1, \dots, \mathcal{T}_{n-3}) \leftarrow$ candidate blobtrees, using
    Algorithm 2
  $(\mathcal{N}_1, \dots, \mathcal{N}_{n-3}) \leftarrow$ semi-directed candidate
    networks obtained from the candidate blobtrees
    $\mathcal{T}_i$, using Algorithm 3
  return network $\mathcal{N}_i$ with highest weighted tf-quarnet
    consistency score
```

SQUIRREL: constructing candidate blobtrees

In the following three steps, we describe how SQUIRREL creates the sequence of candidate blobtrees on leaf set $\mathcal{X}$ from the dense set $\mathcal{Q}$ of tf-quarnets. The pseudocode of this procedure is shown as Algorithm 2 at the end of this subsection.

Step A1: We first create a phylogenetic tree $\mathcal{T}^*$ on $\mathcal{X}$ as described in Berry and Gascuel (2000). Their algorithm takes as input a (possibly non-dense) set $\mathcal{Q}'$ of quartet trees and returns as $\mathcal{T}^*$ the unique most refined phylogenetic tree on $\mathcal{X}$ which does not induce a quartet with a different non-trivial split than one of the quartets in $\mathcal{Q}'$ (see Section A of the Supplementary Material for a more formal definition). By taking $\mathcal{Q}'$ to be the subset of quartet trees in our set of dense tf-quarnets $\mathcal{Q}$ (see line 1 of Algorithm 2) we can employ the algorithm from Berry and Gascuel (2000) to obtain $\mathcal{T}^*$ (see line 2 of Algorithm 2). As we show in Lemma A.2 of the Supplementary Material, in the case all tf-quarnets are induced by a unique network, $\mathcal{T}^*$ coincides with the blobtree of that network.

Step A2: Since the set $\mathcal{Q}$ (and thus $\mathcal{Q}'$) is constructed from real data, we expect there to be a fair amount of quartets that contradict each other. Hence, in practice, the tree $\mathcal{T}^*$ constructed in Step A1 will be highly unresolved. To remedy this problem, we use a method to refine the tree $\mathcal{T}^*$, specifically, an adapted version of the QUARTETJOINING algorithm (Grünewald et al. 2009). QUARTETJOINING takes as input a function ω that assigns a non-negative real number to each possible non-trivial split of four leaves in $\mathcal{X}$. Starting with the star-tree with central vertex v and leaf set $\mathcal{X}$, QUARTETJOINING sequentially introduces edges between v and two of its neighbours (according to some criterion involving the function ω) until the tree is fully resolved.

In our case, we instead start with the tree $\mathcal{T}^*$ (which might already be partially resolved) and adapt QUARTETJOINING to resolve $\mathcal{T}^*$ further. This eventually leads to a fully resolved phylogenetic tree $\mathcal{T}_1$ on $\mathcal{X}$, which functions as the first tree in our sequence of candidate blobtrees (see line 3 of Algorithm 2). In our adaptation, instead of considering all combinations of neighbours of the central vertex v, we consider all such combinations of neighbours of any of the internal (i.e. non-leaf) vertices with degree at least 4. We construct the function ω used as input to QUARTETJOINING as follows. For any tf-quartet $q \in \mathcal{Q}$ with leaf set $\mathcal{L}(q) = \{a, b, c, d\}$ such that q is a quartet tree (say with split $ab|cd$), we set $\omega(ab|cd) = w(q)$ for the input weight function w mentioned at the beginning of the previous subsection. All other non-trivial splits of four leaves of $\mathcal{X}$ are assigned an ω-value of 0.

Step A3: Finally, we explain how we create the full sequence of candidate blobtrees from the phylogenetic tree $\mathcal{T}_1$. Given an edge uv of the tree $\mathcal{T}_1$ that induces a non-trivial split $A|B$, we collect all the quartet trees in $\mathcal{Q}$ for which their induced splits restrict to quartet splits of $A|B$ in a set $\mathcal{Q}'(A|B)$ by first defining $\mathcal{Q}(A|B) = \{q \in \mathcal{Q} : |A \cap \mathcal{L}(q)| = 2, |B \cap \mathcal{L}(q)| = 2\}$ and then $\mathcal{Q}'(A|B) = \{q \in \mathcal{Q}(A|B) : q \text{ has split } A \cap \mathcal{L}(q) | B \cap \mathcal{L}(q)\}$. This allows us to define the *split-support* of uv as

$$\mathrm{supp}(uv) = \frac{w(\mathcal{Q}'(A|B))}{w(\mathcal{Q}(A|B))}, \tag{4}$$

i.e. as the weighted ratio of the tf-quarnets in $\mathcal{Q}$ that support the split induced by the edge uv. For each of the $n-3$ edges of the tree $\mathcal{T}_1$ we then compute this split-support (see line 4 of Algorithm 2). Afterwards, we sort the edges of $\mathcal{T}_1$ in increasing order, according to their split-support. To create the trees $(\mathcal{T}_2, \dots, \mathcal{T}_{n-3})$, we keep contracting the least supported edge (see line 6 of Algorithm 2). In other words, the tree $\mathcal{T}_i$ is obtained from $\mathcal{T}_1$ by contracting the $i-1$ least supported edges. Crucial for our consistency proof in Section A of the Supplementary Material is that $\mathcal{T}_1$ is a refinement of $\mathcal{T}^*$, and therefore one of the trees in the sequence $(\mathcal{T}_1, \dots, \mathcal{T}_{n-3})$ will be the tree $\mathcal{T}^*$.

```
Algorithm 2: Constructing candidate blobtrees
  Input: dense set $\mathcal{Q}$ of weighted tf-quarnets on
         $\mathcal{X} = \{x_1, \dots, x_n\}$
  Output: sequence of candidate blobtrees
          $(\mathcal{T}_1, \dots, \mathcal{T}_{n-3})$ on $\mathcal{X}$
  /* Step A1 */
1 $\mathcal{Q}' \leftarrow$ set of all quartet trees in $\mathcal{Q}$
2 $\mathcal{T}^* \leftarrow$ phylogenetic tree on $\mathcal{X}$ obtained from $\mathcal{Q}'$, as
    described in Berry and Gascuel (2000)
  /* Step A2 */
3 $\mathcal{T}_1 \leftarrow$ phylogenetic tree on $\mathcal{X}$ obtained by applying
    the adapted QUARTETJOINING algorithm to $\mathcal{T}^*$
    and $\mathcal{Q}$
  /* Step A3 */
4 compute the split-support for every edge in $\mathcal{T}_1$
5 for each $i \in \{2, \dots, n-3\}$ do
6     $\mathcal{T}_i$ is constructed from $\mathcal{T}_{i-1}$ by contracting the
      least supported edge
7 return $(\mathcal{T}_1, \dots, \mathcal{T}_{n-3})$
```

SQUIRREL: expanding cycles in a tree

Once SQUIRREL has constructed the sequence of candidate blobtrees using Algorithm 2, we transform them into triangle-free semi-directed level-1 networks using the

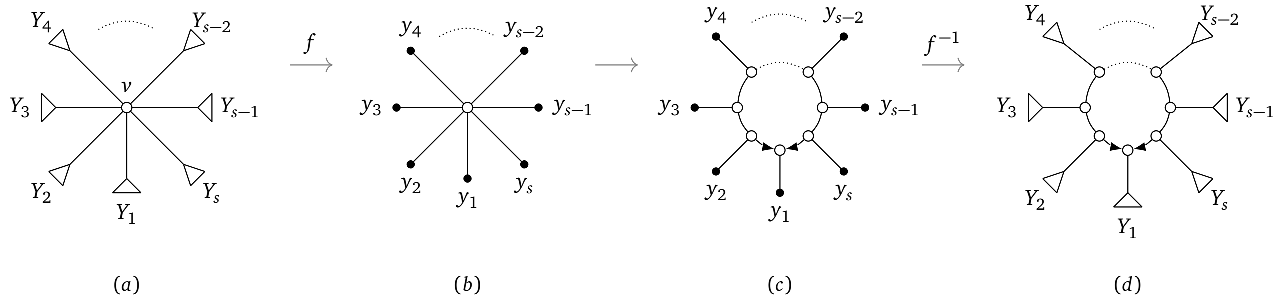


Figure 10: (a): A blobtree on some leaf set $\mathcal{X}$ with an internal vertex v inducing the partition $Y_1|\dots|Y_s$ of $\mathcal{X}$. (b): Illustration of the mapping f which maps every leaf x of $\mathcal{X}$ to a leaf in $\{y_1,\dots,y_s\}$, depending on which set Y_i contains x. (c): Illustration of Step B2 and B3 of SQUIRREL, where the single internal vertex is replaced by a cycle. (d): Illustration of how the cycle on the leaves y_i is mapped back to a cycle on the sets Y_i with the inverse function f^{-1}.

dense set of tf-quarnets $\mathcal{Q}$. In this subsection we describe how we transform a phylogenetic tree $\mathcal{T}$ - representing one of our candidate blobtrees - into such a network $\mathcal{N}$. In particular, we replace every internal vertex of the given tree by a suitable cycle. Since our aim is to build triangle-free networks, we replace vertices incident to $s \geq 4$ edges by an s-cycle with a reticulation (see also the illustration in Figure 10). To this end, we repeat the following three steps for every such internal vertex v (starting with the ones with the highest degree). The corresponding high-level pseudo-code is shown as Algorithm 3.

Step B1: The first step in our approach is to assign a dense set of *representative tf-quarnets* $\tilde{\mathcal{Q}}_v$ to each internal vertex v of $\mathcal{T}$ with degree $s \geq 4$. In particular, the set $\tilde{\mathcal{Q}}_v$ will be a dense set of tf-quarnets on the leaf set $\mathcal{Y} = \{y_1,\dots,y_s\}$, where each y_i represents the set Y_i which is part of the partition $Y_1|\dots|Y_s$ of $\mathcal{X}$ induced by v (see Figure 10(a) and (b)). In the next step, these sets will then be used to determine by what cycle to replace v.

First, let $f : \mathcal{X} \to \mathcal{Y}$ be the function that maps every leaf $x \in \mathcal{X}$, with x being in some set Y_i, to the leaf y_i (see line 3 of Algorithm 3). To construct the tf-quarnets in $\tilde{\mathcal{Q}}_v$ (see line 4 of Algorithm 3), we repeat the following procedure for every subset $\{y_i, y_j, y_k, y_l\}$ of four leaves in $\mathcal{Y}$. Let $\mathcal{Q}_{\{i,j,k,l\}} = \{q \in \mathcal{Q} : \mathcal{L}(q) = \{x_i, x_j, x_k, x_l\}$ with $x_p \in Y_p$ for all $p \in \{i,j,k,l\}\}$ be the subset of $\mathcal{Q}$ containing only tf-quarnets with one leaf in each of the four sets Y_i, Y_j, Y_k and Y_l. By relabeling the leaves of all tf-quarnets in $\mathcal{Q}_{\{i,j,k,l\}}$ with the function f, we obtain a multiset of tf-quarnets which all have the same leaf set $\{y_i, y_j, y_k, y_l\}$. With slight abuse of notation, we denote this multiset by $f(\mathcal{Q}_{\{i,j,k,l\}})$. Then, we choose one of the tf-quarnets in the multiset $f(\mathcal{Q}_{\{i,j,k,l\}})$ to assign to $\tilde{\mathcal{Q}}_v$ as the tf-quarnet on the four-leaf set $\{y_i, y_j, y_k, y_l\}$ (see next paragraph). As mentioned before, this is repeated for every subset $\{y_i, y_j, y_k, y_l\}$ of four leaves in $\mathcal{Y}$, resulting in a dense set of tf-quarnets on $\mathcal{Y}$.

To choose a tf-quarnet from the multiset $f(\mathcal{Q}_{\{i,j,k,l\}})$, we first choose its *skeleton*: its underlying undirected graph. In particular, for each of the six possible skeletons t (three quartet trees and three undirected 4-cycles) we let $w(t)$ be the sum of weights of all tf-quarnets in $f(\mathcal{Q}_{\{i,j,k,l\}})$ with the given skeleton t. We then choose the skeleton t with the highest weight (with ties resolved randomly) and assign it a new weight of $w(t)/w(f(\mathcal{Q}_{\{i,j,k,l\}}))$. Note that in the unweighted case this simply means that we choose the skeleton that appears most in the multiset. We first choose the skeleton since determining the location of the reticulation in a quarnet from data seems especially hard (Martin et al. 2023). If our chosen skeleton is one of the quartet trees, we assign that as our tf-quarnet on $\{y_i, y_j, y_k, y_l\}$. On the other hand, if one of the undirected 4-cycles appears most, we still need to determine the location of the reticulation. This is done by checking which leaf appears most often below the reticulation in all 4-cycles with the chosen skeleton.

As an example of this voting procedure to choose a tf-quarnet from the multiset $f(\mathcal{Q}_{\{i,j,k,l\}})$, suppose our multiset $f(\mathcal{Q}_{\{i,j,k,l\}})$ contains only tf-quarnets with weight 1 and is as in Figure 11.

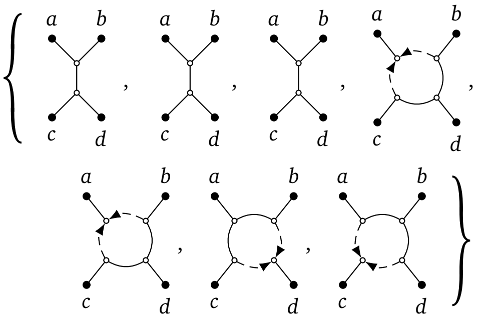


Figure 11: A multiset of 7 tf-quarnets on leaf set $\{a,b,c,d\}$.

Then, we choose the 4-cycle with circular ordering (a,b,c,d) as our skeleton, after which we assign a to be the leaf below the reticulation. The new tf-quarnet is then a 4-cycle with a weight of 4/7 because that 4-cycle appears 4 times out of a total of 7 tf-quarnets.

Step B2: The next step of our approach is to determine a circular ordering of the leaves in the set $\mathcal{Y}$ based on the tf-quarnets in $\tilde{\mathcal{Q}}_v$. Note that we repeat this for every internal vertex v of $\mathcal{T}$ with degree at least 4. First, we use the set $\tilde{\mathcal{Q}}_v$ to create a distance $D_{\tilde{\mathcal{Q}}_v}$ between every pair of leaves in $\mathcal{Y}$ (see line 5 of Algorithm 3). Formally, given two leaves a and b in $\mathcal{Y}$, we define the distance $D_{\tilde{\mathcal{Q}}_v}$ as follows:

$$D_{\tilde{\mathcal{Q}}_v}(a,b) = \begin{cases} 0 & \text{if } a = b, \\ \sum_{q \in \tilde{\mathcal{Q}}_v : a,b \in \mathcal{L}(q)} \tau_q(a,b) & \text{if } a \neq b. \end{cases} \tag{5}$$

For every tf-quarnet $q \in \tilde{\mathcal{Q}}_v$ the exact value of τ depends on the weight of q and the position of the leaves a and b within it. In particular, the values are defined on the skeleton of the tf-quarnets and hence do not depend on the position of the reticulations. Given two leaves a and b of a tf-quarnet q (with weight $w(q) \in [0,1]$), we define τ_q as follows:

$$\tau_q(a,b) = \begin{cases} (3 - w(q))/2 & \text{if } q \text{ is the quartet tree } ab|cd \\ & \text{or if } q \text{ is a 4-cycle with } a, b \\ & \text{as neighbours,} \\ (3 + w(q))/2 & \text{otherwise.} \end{cases} \tag{6}$$

Here, we say that two leaves of a 4-cycle are *neighbours* if they are not on opposite sides of the cycle. The τ_q-values reduce to 1 or 2 for tf-quarnets q with a weight of 1. Specifically, two leaves on the same side of a split in a quartet tree q have a τ_q-value of 1, otherwise they have a τ_q-value of 2. Similarly, two neighbouring leaves in a 4-cycle q have a τ_q-value of 1, while two opposite leaves have a τ_q-value of 2. See Figure 12 for an illustration of these values. Note that these pairwise distances between leaves resemble the *quartet distances* used in NANUQ (Allman et al. 2019) and NANUQ$^+$ (Allman et al. 2024a).

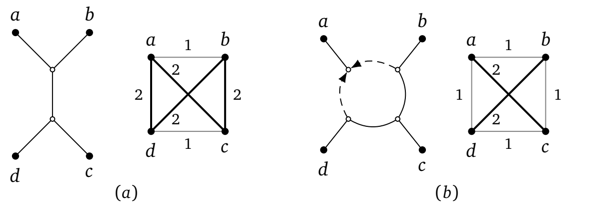


Figure 12: Two tf-quarnets q with leaf set $\{a,b,c,d\}$: a quartet tree (a) and 4-cycle (b). The values τ_q (as defined by eq. (6), assuming the quarnets have weight 1) between any two leaves are illustrated by the two complete graphs, where the thin grey edges have length 1 and the thick black length 2.

Once the distances $D_{\tilde{\mathcal{Q}}_v}$ are computed, we create a complete graph G with vertex set $\mathcal{Y}$, where the distances between the vertices are given by $D_{\tilde{\mathcal{Q}}_v}$. By solving the TRAVELLING SALESMAN PROBLEM (TSP) on this graph we obtain a circular ordering of the elements in $\mathcal{Y}$ (see line 6 of Algorithm 3). The goal of a TSP instance is to find a shortest *Hamiltonian cycle* (or *TSP-tour*): a cycle that visits each vertex exactly once. The default setting for SQUIRREL is to use the Held-Karp algorithm (Bellman 1962; Held and Karp 1962) for up to and including 13 leaves and to use simulated annealing to heuristically solve instances with more leaves. To obtain true consistency (see Section A of the Supplementary Material) this setting can be changed to always solve TSP to optimality, at the cost of a longer running time.

Step B3: After solving TSP, SQUIRREL obtains a circular ordering θ of the leaves in $\mathcal{Y}$. It remains to determine which leaf y_i needs to be the leaf below the reticulation in the resulting cycle. To ensure SQUIRREL always returns a valid (that is, rootable) semi-directed network, we create a *reticulation ranking* ρ of the leaves in $\mathcal{Y}$ instead of picking a single leaf (see line 7 of Algorithm 3). If the set $\mathcal{Y}$ contains at least five elements, we order them according to how often they appear in a 4-cycle of $\tilde{\mathcal{Q}}_v$ (as defined in Step B1). That is, the first leaf in our ranking ρ appears most often in a 4-cycle and is our first option to be the leaf below the reticulation. The case where $|\mathcal{Y}| = 4$ is special, since $\tilde{\mathcal{Q}}_v$ then only contains a single quarnet. If this is a 4-cycle, then the leaf below the reticulation of that 4-cycle is the first leaf in our ranking ρ. The other three leaves (or in the case that the single tf-quarnet is a quartet tree: all four leaves) are ordered randomly.

Finally, we map every leaf y_i back to the corresponding leaf set Y_i of the original tree $\mathcal{T}$ with the inverse function f^{-1}. While slightly abusing notation, this results in an ordering $f^{-1}(\theta)$ of the sets Y_i. Then, we replace the internal vertex v in the tree $\mathcal{T}$ by a cycle that follows this ordering $f^{-1}(\theta)$ (see line 9 of Algorithm 3 and Figure 10(b) and (c) for an illustration). We determine the location of the reticulation by looking at the first element ρ_1 of the reticulation ranking ρ. In particular, we let the leaf set in $f^{-1}(\rho_1)$ be below the reticulation (again see line 9 of Algorithm 3). This could possibly create a partially constructed network that is *invalid*: one without a valid root location (e.g. if two reticulations are oriented towards each other). Hence, if this is the case we instead pick the leaves in $f^{-1}(\rho_2)$. If this is still an invalid option, we keep iterating through the ranking ρ until we find a valid partial network (see line 10 of Algorithm 3). Note that This procedure ensures that we always return a valid semi-directed network at the end of Algorithm 3. Our implementation of SQUIRREL also allows the user to specify a known outgroup. Then, a (partially constructed) semi-directed network is only valid if it is not only rootable, but if it can also be rooted at the edge incident to the outgroup. Iterating through the reticulation ranking ensures that we always return a valid semi-directed network at the end of Algorithm 3, even in the case of a specified outgroup (see Lemma A.6 in Section A of the Supplementary Material for a proof).

δ-heuristic: inferring quarnets from sequence data

As explained before, two model-based methods that use algebraic invariants exist to generate tf-quarnets (Barton et al. 2022; Martin et al. 2023). To allow SQUIRREL to function as a stand-alone tool, we also include a method to infer weighted tf-quarnets from a multiple sequence alignment (MSA) on a set of taxa $\mathcal{X}$: the δ-heuristic. Our δ-heuristic is based on the concept of δ-plots, which function as a measure of treelikeness for sets of four taxa and which were able to pick out recombinants in many simulations (Holland et al. 2002). The algorithm also resembles some aspects of the heuristic to generate trinets from sequences in Oldman et al. (2016). We are now ready to present the steps to create a dense set of weighted tf-quarnets from an MSA on leaf set $\mathcal{X}$.

Algorithm 3: Expanding cycles in a tree

```
Input: dense set $\mathcal{Q}$ of weighted tf-quarnets on
       $\mathcal{X} = \{x_1, \dots, x_n\}$, phylogenetic tree $\mathcal{T}$ on $\mathcal{X}$
Output: triangle-free semi-directed level-1 network
        on $\mathcal{X}$
for each internal vertex $v$ of $\mathcal{T}$ with degree $\geq 4$ do
  // in decreasing order of degree
    /* Step B1 */
    $Y_1|\dots|Y_s \leftarrow$ partition of $\mathcal{X}$ induced by $v$
    $f \leftarrow$ function that maps a leaf $x \in \mathcal{X}$ to a leaf
      $y_i$, depending on the set $Y_i$ that contains $x$
    $\tilde{\mathcal{Q}}_v \leftarrow$ set of representative quarnets of $v$ on leaf
      set $\mathcal{Y} = \{y_1, \dots, y_s\}$
    /* Step B2 */
    compute the distances $D_{\mathcal{Q}_v}(y_i, y_j)$ for all
      $i, j \in \{1, \dots, s\}$
    $\theta \leftarrow$ optimal TSP-tour on $\{y_1, \dots, y_k\}$ with
      respect to distances $D_{\mathcal{Q}_v}$
    /* Step B3 */
    $\rho \leftarrow$ reticulation ranking of the leaves in $\mathcal{Y}$
    for each $j \in \{1, \dots, s\}$ do
        replace $v$ in $\mathcal{T}$ by a cycle $C$ with ordering
          $f^{-1}(\theta)$ and with $f^{-1}(\rho_j)$ below the
          reticulation
        if $\mathcal{T}$ has a valid root location then
            break
return $\mathcal{T}$
```

Step I: For each pair of taxa $\{a, b\}$ we consider the gap-free subalignment of the MSA on $\{a, b\}$. That is, we consider only the columns where both taxon a and b contain no gaps. Using this subalignment, we assign a distance value h_{ab} to the pair $\{a, b\}$. In particular, h_{ab} is the *normalized Hamming distance*: the number of columns of the subalignment where taxon a and b differ, divided by the total length of the subalignment. Recall that if a tf-quarnet on $\{a, b, c, d\}$ has a non-trivial split, it has one of the three splits $ab|cd$, $ac|bd$ or $ad|bc$. For each four taxon subset and for each of these three splits, say $ab|cd$, we then let $h_{ab|cd} = h_{ab} + h_{cd}$.

The δ-value (introduced in Holland et al. 2002) of such a subset $\{a, b, c, d\}$ of $\mathcal{X}$ is now defined as follows (assuming we have that $h_{ab|cd} \geq h_{ac|bd} \geq h_{ad|bc}$):

$$\delta_{\{a,b,c,d\}} = \frac{h_{ab|cd} - h_{ac|bd}}{h_{ab|cd} - h_{ad|bc}}, \tag{7}$$

where $\delta_{\{a,b,c,d\}} = 0$ if $h_{ab|cd} = h_{ac|bd} = h_{ad|bc}$. Intuitively, the δ-value indicates how much support there is from the subalignment that the tf-quarnet on $\{a, b, c, d\}$ has a split. That is, if the value of $\delta_{\{a,b,c,d\}}$ is close to 1, we expect the split $ab|cd$ to be present.

Step II: With the δ-values computed for each subset of four taxa, we partition the 4-taxa sets into two subsets S_λ and F_λ for a predefined threshold value $\lambda \in (0, 1)$. The set S_λ will contain all 4-taxa subsets for which the δ-value is at least λ, while the set F_λ contains those sets with an δ-value smaller than λ. We then expect the sets in S_λ to come from a tf-quarnet with a non-trivial split, while those in F_λ are likely to have come from 4-cycle tf-quarnets. Experiments from Holland et al. (2002) show that an average δ-value higher than 0.3 is often enough to determine whether recombination was present (or equivalently, whether a tf-quarnet has a non-trivial split). Hence, we settle for a value of $\lambda = 0.3$.

Step III: Every 4-taxa set $\{a, b, c, d\}$ in S_λ is assigned a quartet tree. Its split is simply determined by the split $s \in \{ab|cd, ac|bd, ad|bc\}$ for which h_s is the highest. On the other hand, the sets in F_λ will be assigned a 4-cycle. Observe that any 4-cycle tf-quarnet with circular ordering (a, c, b, d) (irrespective of the position of the reticulation) can be turned into the quartet trees with splits $ac|bd$ or $ad|bc$ by deleting exactly one reticulation edge, while this is not possible for the quartet tree with split $ab|cd$. Assuming that the taxa set $\{a, b, c, d\}$ is in the set F_λ and that $h_{ab|cd} \geq h_{ac|bd} \geq h_{ad|cb}$, we therefore assign a 4-cycle with circular ordering (a, c, b, d) to the taxa set. This aligns with the group-based models (see e.g. Gross et al. 2021; Barton et al. 2022) which also assume that DNA independently evolves along the trees that can be obtained from a network by deleting reticulation edges.

We also assign a weight $w(q)$ to each tf-quarnet q, corresponding to the difference its δ-value has from λ. In some sense, this weight signifies the confidence we have in having estimated the correct tf-quarnet. In particular,

$$w(q) = \begin{cases} \frac{|\delta_q - \lambda|}{\lambda} & \text{if } \delta_q \leq \lambda, \\ \frac{|\delta_q - \lambda|}{1-\lambda} & \text{if } \delta_q > \lambda. \end{cases} \tag{8}$$

Step IV: It remains to determine where to place the reticulations in the 4-cycles obtained from the set F_λ. Taking inspiration from Holland et al. (2002) and Oldman et al. (2016), we first compute the value $\delta(x)$ for each taxon x, defined as the mean value of all δ-values for four-taxon sets containing x. For each 4-cycle, we then let the leaf x with the highest $\delta(x)$-value be below the reticulation.

Consistency of Squirrel

In Section A of the Supplementary Material we prove that Squirrel is combinatorially consistent given an unweighted dense set of tf-quarnets. We use the word 'combinatorially' to emphasize that we do not make any claims regarding statistical consistency. More formally, we prove the following theorem.

Theorem 1. *Let $\mathcal{N}$ be a triangle-free semi-directed level-1 network and let $\mathcal{Q}$ be the set of unweighted tf-quarnets induced by $\mathcal{N}$, then* Squirrel *applied to $\mathcal{Q}$ reconstructs $\mathcal{N}$.*

The first ingredient of the proof is the fact that if a set of tf-quarnets is induced by a network, the tree $\mathcal{T}^*$ is equal to the blobtree of that network. The other important step of the proof is to show that in this case the distances D (as defined in eq. (6)) form a *Kalmanson metric* (Kalmanson 1975), which have nice properties with respect to the Travelling Salesman Problem.

Implementation

A graphical user interface (implemented in Python) of SQUIRREL and the δ-heuristic is freely available at https://github.com/nholtgrefe/squirrel. The program takes as input a sequence alignment in NEXUS or FASTA format, or a file specifying a dense set of tf-quarnets (e.g. coming from QNR-SVM (Barton et al. 2022) or the MML algorithm (Martin et al. 2023)). The interface allows the user to specify an optional outgroup, view the different generated candidate networks, and export them in the eNewick file-format (Cardona et al. 2008) (with an arbitrary rooting if no outgroup was specified).

Supplementary Material

Supplementary material is available at Molecular Biology and Evolution online (http://www.mbe.oxfordjournals.org/).

Data availability statement: The generated networks, Python scripts, sequence alignments and numerical results of the experiments in this paper are available at https://github.com/nholtgrefe/squirrel.

Acknowledgements

This work received funding from grants OCENW.M.21.306 (NH, LvI & MJ) and OCENW.KLEIN.125 (LvI & MJ) of the Netherlands Organisation for Scientific Research (NWO). Part of this work was done while some of the authors were in residence at the Institute for Computational and Experimental Research in Mathematics (ICERM) in Providence (RI, USA) during the *Theory, Methods, and Applications of Quantitative Phylogenomics* program (supported by grant DMS-1929284 of the National Science Foundation (NSF)). We thank the authors of Allman et al. (2024a) for making us aware of their new approach and the reviewers for their helpful comments and suggestions to improve the paper.

References

Allman ES, Baños H, Rhodes JA, Wicke K. 2024a. NANUQ$^+$: A divide-and-conquer approach to network estimation. bioRxiv:10.1101/2024.10.30.621146.

Allman ES, Baños H, Mitchell JD, Rhodes JA. 2024b. TINNiK: inference of the tree of blobs of a species network under the coalescent model. *Algorithms for Molecular Biology*. 19:23.

Allman ES, Baños H, Rhodes JA. 2019. NANUQ: a method for inferring species networks from gene trees under the coalescent model. *Algorithms for Molecular Biology*. 14:1–25.

Bandelt HJ, Dress A. 1986. Reconstructing the shape of a tree from observed dissimilarity data. *Advances in Applied Mathematics*. 7:309–343.

Baños H. 2019. Identifying species network features from gene tree quartets under the coalescent model. *Bulletin of Mathematical Biology*. 81:494–534.

Barton T, Gross E, Long C, Rusinko J. 2022. Statistical learning with phylogenetic network invariants. arXiv:2211.11919.

Bellman R. 1962. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM (JACM)*. 9:61–63.

Berry V, Gascuel O. 2000. Inferring evolutionary trees with strong combinatorial evidence. *Theoretical Computer Science*. 240:271–298.

Cardona G, Rosselló F, Valiente G. 2008. Extended newick: it is time for a standard representation of phylogenetic networks. *BMC bioinformatics*. 9:1–8.

Chifman J, Kubatko L. 2014. Quartet Inference from SNP Data Under the Coalescent Model. *Bioinformatics*. 30:3317–3324.

Colonius H, Schulze HH. 1981. Tree structures for proximity data. *British Journal of Mathematical and Statistical Psychology*. 34:167–180.

Cui R, Schumer M, Kruesi K, Walter R, Andolfatto P, Rosenthal GG. 2013. Phylogenomics reveals extensive reticulate evolution in xiphophorus fishes. *Evolution*. 67:2166–2179.

Culumber Z, Fisher H, Tobler M, Mateos M, Barber P, Sorenson M, Rosenthal G. 2011. Replicated hybrid zones of xiphophorus swordtails along an elevational gradient. *Molecular Ecology*. 20:342–356.

Deĭneko VG, van der Veen JA, Rudolf R, Woeginger GJ. 1997. Three easy special cases of the euclidean travelling salesman problem. *RAIRO-Operations Research*. 31:343–362.

Diop A, Torrance EL, Stott CM, Bobay LM. 2022. Gene flow and introgression are pervasive forces shaping the evolution of bacterial species. *Genome Biology*. 23:239.

Du K, Ricci JMB, Lu Y, Garcia-Olazabal M, Walter RB, Warren WC, Dodge TO, Schumer M, Park H, Meyer A, et al. 2024. Phylogenomic analyses of all species of swordtail fishes (genus xiphophorus) show that hybridization preceded speciation. *Nature Communications*. 15:6609.

Ehrendorfer F. 1959. Differentiation-hybridization cycles and polyploidy in achillea. In: Cold Spring Harbor Symposia on Quantitative Biology. Cold Spring Harbor Laboratory Press, volume 24, pp. 141–152.

Frohn M, Holtgrefe N, van Iersel L, Jones M, Kelk S. 2024. Reconstructing semi-directed level-1 networks using few quarnets. arXiv:2409.06034.

Green RE, Krause J, Briggs AW, Maricic T, Stenzel U, Kircher M, Patterson N, Li H, Zhai W, Fritz MHY, et al. 2010. A draft sequence of the neandertal genome. *Science*. 328:710–722.

Gross E, van Iersel L, Janssen R, Jones M, Long C, Murakami Y. 2021. Distinguishing level-1 phylogenetic networks on the basis of data generated by markov processes. *Journal of Mathematical Biology*. 83:1–24.

Grünewald S, Moulton V, Spillner A. 2009. Consistency of the qnet algorithm for generating planar split networks from weighted quartets. *Discrete Applied Mathematics*. 157:2325–2334.

Held M, Karp RM. 1962. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*. 10:196–210.

Holland BR, Huber KT, Dress A, Moulton V. 2002. δ plots: a tool for analyzing phylogenetic distance data. *Molecular Biology and Evolution*. 19:2051–2059.

Huber KT, van Iersel L, Jones M, Moulton V, Veenema-Nipius L. 2024. When are quarnets sufficient to reconstruct semi-directed phylogenetic networks? arXiv:2408.12997.

Huber KT, van Iersel L, Kelk S, Suchecki R. 2010. A practical algorithm for reconstructing level-1 phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 8:635–649.

Jiao Y, An M, Zhang N, Zhang H, Zheng C, Chen L, Li H, Zhang Y, Gan Y, Zhao J, et al. 2024. Multiple third-generation recombinants formed by crf55_01b and crf07_bc in newly diagnosed hiv-1 infected patients in shenzhen city, china. *Virology Journal*. 21:306.

Kalmanson K. 1975. Edgeconvex circuits and the traveling salesman problem. *Canadian Journal of Mathematics*. 27:1000–1010.

Kang JH, Schartl M, Walter RB, Meyer A. 2013. Comprehensive phylogenetic analysis of all species of swordtails and platies (pisces: Genus xiphophorus) uncovers a hybrid origin of a swordtail fish, xiphophorus monticolus, and demonstrates that the sexually selected sword originated in the ancestral lineage of the genus, but was lost again secondarily. *BMC evolutionary biology*. 13:1–19.

Kong S, Swofford DL, Kubatko LS. 2024. Inference of Phylogenetic Networks from Sequence Data using Composite Likelihood. *Systematic Biology*. p. syae054.

Lemey P, Salemi M, Vandamme AM. 2009. The phylogenetic handbook: a practical approach to phylogenetic analysis and hypothesis testing. Cambridge University Press.

Martin S, Moulton V, Leggett RM. 2023. Algebraic invariants for inferring 4-leaf semi-directed phylogenetic networks. bioRxiv:10.1101/2023.09.11.557152.

Maxfield M, Xu J, Ané C. 2024. A dissimilarity measure for semidirected networks. arXiv:2405.16035.

Meier JI, Stelkens RB, Joyce DA, Mwaiko S, Phiri N, Schliewen UK, Selz OM, Wagner CE, Katongo C, Seehausen O. 2019. The coincidence of ecological opportunity with hybridization explains rapid adaptive radiation in lake mweru cichlid fishes. *Nature communications*. 10:5391.

Meyer A, Salzburger W, Schartl M. 2006. Hybrid origin of a swordtail species (teleostei: Xiphophorus clemenciae) driven by sexual selection. *Molecular Ecology*. 15:721–730.

Oldman J, Wu T, van Iersel L, Moulton V. 2016. TriLoNet: Piecing Together Small Networks to Reconstruct Reticulate Evolutionary Histories. *Molecular Biology and Evolution*. 33:2151–2162.

Patterson N, Richter DJ, Gnerre S, Lander ES, Reich D. 2006. Genetic evidence for complex speciation of humans and chimpanzees. *Nature*. 441:1103–1108.

Pekar J, Worobey M, Moshiri N, Scheffler K, Wertheim JO. 2021. Timing the sars-cov-2 index case in hubei province. *Science*. 372:412–417.

Perelman P, Johnson WE, Roos C, Seuánez HN, Horvath JE, Moreira MAM, Kessing B, Pontius J, Roelke M, Rumpler Y, et al. 2011. A molecular phylogeny of living primates. *PLOS Genetics*. 7:1–17.

Rambaut A, Grass NC. 1997. Seq-gen: an application for the monte carlo simulation of dna sequence evolution along phylogenetic trees. *Bioinformatics*. 13:235–238.

Rhodes JA, Baños H, Xu J, Ané C. 2025. Identifying circular orders for blobs in phylogenetic networks. *Advances in Applied Mathematics*. 163:102804.

Rieseberg LH, Raymond O, Rosenthal DM, Lai Z, Livingstone K, Nakazato T, Durphy JL, Schwarzbach AE, Donovan LA, Lexer C. 2003. Major ecological transitions in wild sunflowers facilitated by hybridization. *Science*. 301:1211–1216.

Rosenthal GG, de la Rosa Reyna XF, Kazianis S, Stephens MJ, Morizot DC, Ryan MJ, García de León FJ. 2003. Dissolution of sexual signal complexes in a hybrid zone between the swordtails xiphophorus birchmanni and xiphophorus malinche (poeciliidae). *Copeia*. 2003:299–307.

Salemi M, Vandamme AM. 2003. The phylogenetic handbook: a practical approach to DNA and protein phylogeny. Cambridge University Press.

Schmidt HA, Strimmer K, Vingron M, von Haeseler A. 2002. Tree-puzzle: maximum likelihood phylogenetic analysis using quartets and parallel computing. *Bioinformatics*. 18:502–504.

Schumer M, Cui R, Boussau B, Walter R, Rosenthal G, Andolfatto P. 2013. An evaluation of the hybrid speciation hypothesis for xiphophorus clemenciae based on whole genome sequences. *Evolution*. 67:1155–1168.

Simmons MP, Gatesy J. 2015. Coalescence vs. concatenation: Sophisticated analyses vs. first principles applied to rooting the angiosperms. *Molecular Phylogenetics and Evolution*. 91:98–122.

Solís-Lemus C, Ané C. 2016. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS genetics*. 12:e1005896.

Solís-Lemus C, Bastide P, Ané C. 2017. Phylonetworks: a package for phylogenetic networks. *Molecular Biology and Evolution*. 34:3292–3298.

Soucy SM, Huang J, Gogarten JP. 2015. Horizontal gene transfer: building the web of life. *Nature Reviews Genetics*. 16:472–482.

Springer MS, Meredith RW, Gatesy J, Emerling CA, Park J, Rabosky DL, Stadler T, Steiner C, Ryder OA, Janečka JE, et al. 2012. Macroevolutionary dynamics and historical biogeography of primate diversification inferred from a species supermatrix. *PloS one*. 7:1–23.

Steenwyk JL, Li Y, Zhou X, Shen XX, Rokas A. 2023. Incongruence in the phylogenomics era. *Nature Reviews Genetics*. 24:834–850.

Taylor SA, Larson EL. 2019. Insights from genomes into the evolutionary importance and prevalence of hybridization in nature. *Nature ecology & evolution*. 3:170–177.

Than C, Ruths D, Nakhleh L. 2008. Phylonet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC bioinformatics*. 9:1–16.

Tung J, Barreiro LB. 2017. The contribution of admixture to primate evolution. *Current opinion in genetics & development*. 47:61–68.

Vanderpool D, Minh BQ, Lanfear R, Hughes D, Murali S, Harris RA, Raveendran M, Muzny DM, Hibbins MS, Williamson RJ, et al. 2020. Primate phylogenomics uncovers multiple rapid radiations and ancient interspecific introgression. *PLOS Biology*. 18:1–27.

Warnow T, Tabatabaee Y, Evans SN. 2024. Advances in estimating level-1 phylogenetic networks from unrooted snps. *Journal of Computational Biology*. 32:3–27.

Worobey M, Gemmel M, Teuwen DE, Haselkorn T, Kunstman K, Bunce M, Muyembe JJ, Kabongo JMM, Kalengayi RM, Van Marck E, et al. 2008. Direct evidence of extensive diversity of hiv-1 in kinshasa by 1960. *Nature*. 455:661–664.

Wu Z, Solís-Lemus C. 2024. Ultrafast learning of four-node hybridization cycles in phylogenetic networks using algebraic invariants. *Bioinformatics Advances*. 4:vbae014.

Xu J, Ané C. 2023. Identifiability of local and global features of phylogenetic networks from average distances. *Journal of Mathematical Biology*. 86:12.

Yu Y, Nakhleh L. 2015. A maximum pseudo-likelihood approach for phylogenetic networks. *BMC genomics*. 16:1–10.

Zhang C, Mirarab S. 2022. Weighting by Gene Tree Uncertainty Improves Accuracy of Quartet-based Species Trees. *Molecular Biology and Evolution*. 39:msac215.

Zhang C, Ogilvie HA, Drummond AJ, Stadler T. 2018a. Bayesian inference of species networks from multilocus sequence data. *Molecular Biology and Evolution*. 35:504–517.

Zhang C, Rabiee M, Sayyari E, Mirarab S. 2018b. ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC bioinformatics*. 19:15–30.

Zhang W, Dasmahapatra KK, Mallet J, Moreira GR, Kronforst MR. 2016. Genome-wide introgression among distantly related heliconius butterfly species. *Genome biology*. 17:1–15.

A Proof of consistency

In this section we prove that SQUIRREL is combinatorially consistent given an unweighted dense set of tf-quarnets:

Theorem 1. *Let $\mathcal{N}$ be a triangle-free semi-directed level-1 network and let $\mathcal{Q}$ be the set of unweighted tf-quarnets induced by $\mathcal{N}$, then* SQUIRREL *applied to $\mathcal{Q}$ reconstructs $\mathcal{N}$.*

We start by giving a formal definition (taken from Berry and Gascuel (2000), originally by Bandelt and Dress (1986)) of the tree $\mathcal{T}^*$ constructed in Section 4.3, after which we prove that it is a combinatorially consistent estimator of the blobtree of the network $\mathcal{N}$.

Given a non-trivial split $A|B$ of $\mathcal{X}$, we let $\mathcal{Q}'_{A|B}$ be the set of all possible quartets that *agree* with the split $A|B$. That is, for every $a_1, a_2 \in A$ and $b_1, b_2 \in B$, $\mathcal{Q}'_{A|B}$ contains the quartet q with split $a_1a_2|b_1b_2$. Given a (possibly non-dense) set of quartets $\mathcal{Q}'$ on $\mathcal{X}$, we let S^* be the maximal set of splits of $\mathcal{X}$ such that $\mathcal{Q}'_{A|B} \subseteq \mathcal{Q}'$ for all splits $A|B$ in S^*. In other words, S^* is the maximal set of splits such that $\mathcal{Q}'$ contains all quartets that agree with these splits. Lastly, we define the set $\mathcal{Q}^*$ as the subset of $\mathcal{Q}'$ that contains exactly these quartets, i.e. $\mathcal{Q}^* = \bigcup_{A|B \in S^*} \mathcal{Q}'_{A|B}$. In Berry and Gascuel (2000) it is shown that the set $\mathcal{Q}^*$ can also be characterized as the unique maximum subset of $\mathcal{Q}'$ that is *tree-like*, meaning that there exists a phylogenetic tree $\mathcal{T}$ on $\mathcal{X}$ with $\mathcal{Q}^*$ as the set of resolved quartets (i.e. the quartets contain a non-trivial split) it induces. This is the tree $\mathcal{T}^*(\mathcal{Q}')$ (or simply $\mathcal{T}^*$ if the set $\mathcal{Q}'$ is clear). Note that this tree is unique since a phylogenetic tree is uniquely determined by its quartets (Colonius and Schulze 1981), and thus by its resolved quartets.

We are now ready to prove that the tree $\mathcal{T}^*$ constructed by SQUIRREL is exactly the blobtree of $\mathcal{N}$, provided the tf-quarnets induced by $\mathcal{N}$ are used as input. Recall that to construct the tree $\mathcal{T}^*$ given a dense set $\mathcal{Q}$ of tf-quarnets, SQUIRREL first creates a set of quartets $\mathcal{Q}' \subseteq \mathcal{Q}$ by throwing out all the 4-cycles. This set can then be used by to obtain the tree $\mathcal{T}^*$ as described in Berry and Gascuel (2000).

Lemma A.2. *Let $\mathcal{N}$ be a triangle-free semi-directed level-1 network, let $\mathcal{Q}$ be the set of tf-quarnets induced by $\mathcal{N}$ and let $\mathcal{Q}' \subseteq \mathcal{Q}$ be the set of quartet trees in $\mathcal{Q}$. Then, the blobtree of $\mathcal{N}$ is equal to the tree $\mathcal{T}^*(\mathcal{Q}')$.*

Proof. Let $\tilde{\mathcal{Q}}$ be the set of resolved quartets (i.e. quartets with a non-trivial split) induced by the blobtree $\mathcal{T}$ of $\mathcal{N}$. It is enough to show that $\tilde{\mathcal{Q}} = \mathcal{Q}^*$ since phylogenetic trees are uniquely determined by their induced resolved quartets (Colonius and Schulze 1981) and, by definition, $\mathcal{Q}^*$ is the set of induced resolved quartets of $\mathcal{T}^*(\mathcal{Q}')$.

Let $\tilde{q} \in \tilde{\mathcal{Q}}$ be a resolved quartet with split $\tilde{a}_1\tilde{a}_2|\tilde{b}_1\tilde{b}_2$. This means that there is some non-trivial split $A|B$ in $\mathcal{T}$ (and thus in $\mathcal{N}$) with $\tilde{a}_1, \tilde{a}_2 \in A$ and $\tilde{b}_1, \tilde{b}_2 \in B$. Now let $a_1, a_2 \in A$ and $b_1, b_2 \in B$ be arbitrary. By Theorem 5.1 in Huber et al. (2024) the quarnet q with $\mathcal{L}(q) = \{a_1, a_2, b_1, b_2\}$ that is induced by $\mathcal{N}$ then has split $a_1a_2|b_1b_2$. This directly implies that there is a quartet in $\mathcal{Q}'$ with split $a_1a_2|b_1b_2$. Consequently, $\mathcal{Q}'_{A|B} \subseteq \mathcal{Q}'$ and thus $\mathcal{Q}'_{A|B} \subseteq \mathcal{Q}^*$. Because $\tilde{q}$ agrees with $A|B$ we have $\tilde{q} \in \mathcal{Q}'_{A|B}$, so we obtain that $\tilde{q} \in \mathcal{Q}^*$. Since $\tilde{q} \in \tilde{\mathcal{Q}}$ was arbitrary, $\tilde{\mathcal{Q}} \subseteq \mathcal{Q}^*$.

Now let $q^* \in \mathcal{Q}^*$ be an arbitrary resolved quartet with split $a_1^*a_2^*|b_1^*b_2^*$. By definition, q^* must be in $\mathcal{Q}'_{A|B} \subseteq \mathcal{Q}$ for some non-trivial split $A|B$ (with $a_1^*, a_2^* \in A$ and $b_1^*, b_2^* \in B$). Since $\mathcal{Q}'_{A|B} \subseteq \mathcal{Q}'$, for any $a_1, a_2 \in A$ and $b_1, b_2 \in B$ the quartet q with split $a_1a_2|b_1b_2$ is in $\mathcal{Q}'$. By (Huber et al. 2024, Thm. 5.1), this means that $A|B$ is a split in $\mathcal{N}$ and thus in $\mathcal{T}$. But q^* is a resolved quartet of $\mathcal{T}$, so $q^* \in \tilde{\mathcal{Q}}$. Since q^* was arbitrary, this shows that $\mathcal{Q}^* \subseteq \tilde{\mathcal{Q}}$. □

We continue with proving that the approach in Step B2 of SQUIRREL correctly determines the ordering of each cycle. For the rest of this section, we consider $\mathcal{Q}$ to be a dense set of unweighted tf-quarnets induced by a *sunlet network*: a semi-directed level-1 network consisting of a single cycle with pendent leaves. This will then be enough to prove the consistency at the end of this section.

Clearly, a sunlet network induces a circular ordering of its leaf set $\mathcal{Y}$. We now create an explicit formula for the distance function $D_\mathcal{Q}$ defined in Section 4.4, assuming the set $\mathcal{Q}$ contains tf-quarnets induced by a sunlet network. Note that a distance function D on a finite set of elements $\mathcal{Y}$ is a *metric* if (i) it is symmetric; (ii) the triangle inequality holds; (iii) the distance between two elements is zero if and only if the elements are equal.

Lemma A.3. *Let $\mathcal{N}$ be a sunlet network on $\mathcal{Y} = \{y_1, \dots, y_n\}$, let $\theta = (y_1, \dots, y_n)$ be a circular ordering of $\mathcal{Y}$ induced by $\mathcal{N}$ such that y_1 is the leaf below the reticulation. If $\mathcal{Q}$ is the set of unweighted tf-quarnets induced by $\mathcal{N}$, then $D_\mathcal{Q}$ is a metric on $\mathcal{Y}$ and it can be expressed as*

$$D_\mathcal{Q}(y_i, y_j) = \frac{1}{2} \cdot \begin{cases} 0 & \textit{if } i = j; \\ n^2 - 9n - 2j^2 + (2n+4) \cdot j + 6 & \textit{if } i = 1 \textit{ and } i \neq j; \\ n^2 - 11n - i^2 - j^2 + (2n+1) \cdot j + 3i + 10 & \textit{if } i \geq 2 \textit{ and } i \neq j. \end{cases}$$

Proof. Clearly, $D_\mathcal{Q}$ is symmetric and two leaves have distance zero if and only if they are the same. To see that the triangle inequality holds, note that $D_\mathcal{Q}$ is defined as a sum of τ-values. Since those values all adhere to the triangle inequality (assuming the tf-quarnets all have weight 1), it readily follows that $D_\mathcal{Q}$ also has the same property. Hence, it is a metric.

We will now find the expression for the distances defined by $D_{\mathcal{Q}}$. Given two leaves y_i and y_j with $1 \leq i < j \leq n$, let $r_i = i - 2$ be the number of leaves between y_1 and y_i, and let $t_j = n - j$ be the number of leaves between y_j and y_1 (on the side of the cycle containing leaf y_n). Lastly, let $s_{ij} = j - i - 1$ be the number of leaves between y_i and y_j. See Figure A.13 for an illustration.

Now let $i = 1$ and let $i < j \leq n$ be arbitrary. Any tf-quarnet in $\mathcal{Q}$ containing both y_i and y_j will be a 4-cycle. In particular, in $\binom{s_{1j}}{2} + \binom{t_j}{2}$ of these 4-cycles, y_i and y_j will be opposite leaves. On the other hand, in $s_{1j} \cdot t_j$ of these 4-cycles they will be neighbours.

We now let $1 < i < j \leq n$ be arbitrary. In this case, any tf-quarnet in $\mathcal{Q}$ containing both y_i and y_j will either be a quartet tree or a 4-cycle. All 4-cycle tf-quarnets must contain leaf y_1 as well. Then, in $r_i + t_j$ of them y_i and y_j are opposite leaves, while in s_{ij} of them they are neighbours. All quartet trees that include y_i and y_j, do not include leaf y_1. In $\binom{r_i}{2} + \binom{t_j}{2}$ of the quartet trees, y_i and y_j are on the same side of the split, i.e. there is no '$y_i|y_j$ split'. The number of quartet trees that do have a '$y_i|y_j$ split' is $\binom{s_{ij}}{2} + r_i \cdot t_j + r_i \cdot s_{ij} + s_{ij} \cdot t_j$. Filling in the distances defined by $D_{\mathcal{Q}}$, we thus obtain for each $1 \leq i < j \leq n$:

$$D_{\mathcal{Q}}(y_i, y_j) = \begin{cases} \overbrace{\left[\binom{s_{1j}}{2} + \binom{t_j}{2}\right]}^{\#4C: y_i, y_j \text{ neighbours}} + 2 \cdot \overbrace{\left[s_{1j} \cdot t_j\right]}^{\substack{\#4C: y_i, y_j \\ \text{opposite}}} & \text{if } i = 1; \\ \underbrace{\left[r_i + t_j\right]}_{\#4C: y_i, y_j \text{ neighbours}} + 2 \cdot \underbrace{\left[s_{ij}\right]}_{\substack{\#4C: y_i, y_j \\ \text{opposite}}} + \underbrace{\left[\binom{r_i}{2} + \binom{t_j}{2}\right]}_{\#QT \text{ without } y_i|y_j \text{ split}} + 2 \cdot \underbrace{\left[\binom{s_{ij}}{2} + r_i \cdot t_j + r_i \cdot s_{ij} + s_{ij} \cdot t_j\right]}_{\#QT \text{ with } y_i|y_j \text{ split}} & \text{if } i \geq 2. \end{cases}$$

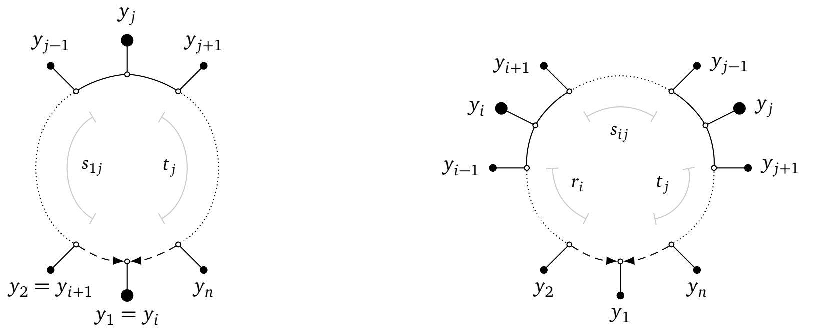
If we fill in the formulas for r_i, t_j and s_{ij}, we obtain

$$D_{\mathcal{Q}}(y_i, y_j) = \begin{cases} \left[\binom{j-2}{2} + \binom{n-j}{2}\right] + 2 \cdot [(j-2) \cdot (n-j)] & \text{if } i = 1; \\ [(i-2) + (n-j)] + 2 \cdot (j-i-1) + \left[\binom{i-2}{2} + \binom{n-j}{2}\right] & \\ \quad +2 \cdot \left[\binom{j-i-1}{2} + (i-2) \cdot (n-j) + (i-2) \cdot (j-i-1) + (j-i-1) \cdot (n-j)\right] & \text{if } i \geq 2, \end{cases}$$

which reduces to

$$D_{\mathcal{Q}}(y_i, y_j) = \begin{cases} \binom{n-j}{2} + \binom{j-2}{2} + 2 \cdot (n-j) \cdot (j-2) & \text{if } i = 1; \\ \binom{n-j+1}{2} + \binom{i-1}{2} + 2 \cdot \binom{j-i}{2} + 2 \cdot (n-j) \cdot (j-3) + 2 \cdot (j-i-1) \cdot (i-2) & \text{if } i \geq 2. \end{cases}$$

After writing out the binomial coefficients and expanding all brackets, one obtains the desired formula for those y_i and y_j with $1 \leq i < j \leq n$. The other cases follow since $D_{\mathcal{Q}}$ is a metric. □



(a) (b)

Figure A.13: Illustration of the definitions of r_i, s_{ij} and t_j in the proof of Lemma A.3. Subfigure (a) is the case where $i = 1$ and subfigure (b) is the case where $i > 1$.

The explicit formula derived in the previous lemma allows us to prove that $D_{\mathcal{Q}}$ is *Kalmanson* (Kalmanson 1975) with respect to the circular ordering θ of the leaves. Such a metric D defined on a set of elements $\mathcal{Y}$ has the following nice property: it is easy to find an optimal TSP-tour in the complete graph on $\mathcal{Y}$ with distances defined by D. Moreover, the corresponding ordering θ defines an optimal TSP-tour. Formally, these metrics are defined as follows.

Definition A.4 (Kalmanson metric). Let $\theta = (y_1, \ldots, y_n)$ be an ordering of a finite set of elements $\mathcal{Y} = \{y_1, \ldots, y_n\}$. A metric D on $\mathcal{Y}$ is *Kalmanson* with respect to θ if both of the following conditions hold:

$$D(y_i, y_j) + D(y_k, y_l) \leq D(y_i, y_k) + D(y_j, y_l) \text{ for all } 1 \leq i < j < k < l \leq n,$$

$$D(y_i, y_l) + D(y_j, y_k) \leq D(y_i, y_k) + D(y_j, y_l) \text{ for all } 1 \leq i < j < k < l \leq n.$$

In the next lemma we prove that $D_{\mathcal{Q}}$ is a Kalmanson metric with respect to a circular ordering of the leaves induced by a sunlet network. From this it follows that TSP can be used to find such an ordering.

Lemma A.5. *Let $\mathcal{N}$ be a sunlet network on $\mathcal{Y} = \{y_1, \dots, y_n\}$ and let $\mathcal{Q}$ be the set of unweighted tf-quarnets induced by $\mathcal{N}$. Then, a circular ordering θ of $\mathcal{Y}$ is induced by $\mathcal{N}$ if and only if θ is an optimal TSP-tour defined by $D_{\mathcal{Q}}$.*

Proof. Let $\theta = (y_1, \dots, y_n)$ be an ordering of $\mathcal{Y}$ induced by $\mathcal{N}$. We will first show that $D_{\mathcal{Q}}$ is Kalmanson with respect to θ. Then, it will follow from Kalmanson (1975) that θ is an optimal TSP-tour. Without loss of generality, we can assume that y_1 is the leaf below the reticulation since being Kalmanson is invariant under cyclic permutations (see e.g. Deĭneko et al. 1997). We will now prove that $D_{\mathcal{Q}}$ is Kalmanson with respect to θ by checking the Kalmanson conditions from Definition A.4. For easier notation, we multiply the expressions by 2. Now let $1 \leq i < j < k < l \leq n$ be arbitrary. Using the explicit formula of Lemma A.3, we then distinguish between the cases where $i = 1$ and $i > 1$. Note that the $n^2 - 9n + 6$ and $n^2 - 11n + 10$ parts cancel out in all conditions.

Case 1: $i = 1$. To prove the first condition we use that $j < k$:

$$\begin{aligned} 2 \cdot \big(D_{\mathcal{Q}}(y_i, y_k) + D_{\mathcal{Q}}(y_j, y_l) - D_{\mathcal{Q}}(y_i, y_j) - D_{\mathcal{Q}}(y_k, y_l)\big) &= \big[-2k^2 + (2n+4) \cdot k\big] + \big[-j^2 - l^2 + (2n+1) \cdot l + 3j\big] \\ &\quad - \big[-2j^2 + (2n+4) \cdot j\big] - \big[-k^2 - l^2 + (2n+1) \cdot l + 3k\big] \\ &= (k - j) \cdot (2n + k - j + 1) > 0. \end{aligned}$$

For the second condition, we use the fact that $3 < k < l$:

$$\begin{aligned} 2 \cdot \big(D_{\mathcal{Q}}(y_i, y_k) + D_{\mathcal{Q}}(y_j, y_l) - D_{\mathcal{Q}}(y_i, y_l) - D_{\mathcal{Q}}(y_j, y_k)\big) &= \big[-2k^2 + (2n+4) \cdot k\big] + \big[-j^2 - l^2 + (2n+1) \cdot l + 3j\big] \\ &\quad - \big[-2l^2 + (2n+4) \cdot l\big] - \big[-j^2 - k^2 + (2n+1) \cdot k + 3j\big] \\ &= (l - k) \cdot (l + k - 3) > 0. \end{aligned}$$

Case 2: $i > 1$. The first condition follows from the fact that $k > j$ and $2n > 2$:

$$\begin{aligned} 2 \cdot \big(D_{\mathcal{Q}}(y_i, y_k) + D_{\mathcal{Q}}(y_j, y_l) - D_{\mathcal{Q}}(y_i, y_j) - D_{\mathcal{Q}}(y_k, y_l)\big) &= \big[-i^2 - k^2 + (2n+1) \cdot k + 3i\big] + \big[-j^2 - l^2 + (2n+1) \cdot l + 3j\big] \\ &\quad - \big[-i^2 - j^2 + (2n+1) \cdot j + 3i\big] - \big[-k^2 - l^2 + (2n+1) \cdot l + 3k\big] \\ &= (2n - 2) \cdot (k - j) > 0. \end{aligned}$$

The second condition is trivial:

$$\begin{aligned} 2 \cdot \big(D_{\mathcal{Q}}(y_i, y_k) + D_{\mathcal{Q}}(y_j, y_l) - D_{\mathcal{Q}}(y_i, y_l) - D_{\mathcal{Q}}(y_j, y_k)\big) &= \big[-i^2 - k^2 + (2n+1) \cdot k + 3i\big] + \big[-j^2 - l^2 + (2n+1) \cdot l + 3j\big] \\ &\quad - \big[-i^2 - l^2 + (2n+1) \cdot l + 3i\big] - \big[-j^2 - k^2 + (2n+1) \cdot k + 3j\big] \\ &= 0. \end{aligned}$$

It remains to show that any circular ordering ϕ of $\mathcal{Y}$ that is not induced by $\mathcal{N}$ is not an optimal TSP-tour. To see this, we argue that the total TSP-distance of any such ordering ϕ can always be decreased by swapping two specific adjacent leaves. In particular, any such ordering ϕ will have four leaves $\{y_i, y_j, y_k, y_l\}$ (with $1 \leq i < j < k < l \leq n$) adjacent in the ordering ϕ but ordered as (y_i, y_k, y_j, y_l) (if $i \neq 1$), or ordered as (y_1, y_k, y_j, y_l) or (y_j, y_1, y_k, y_l) (if $i = 1$). In all three cases, we can swap two of these adjacent leaves to decrease the total distance, since the corresponding three Kalmanson inequalities are strict inequalities. □

We are now ready to present the proof of Theorem 1.

Proof of Theorem 1. Since the set of tf-quarnets $\mathcal{Q}$ is induced by the network $\mathcal{N}$, we know by Lemma A.2 that the tree $\mathcal{T}^*$ (as constructed in Step A1) is equal to the blobtree of $\mathcal{N}$. The tree $\mathcal{T}_1$ (in Step A2) is a *refinement* of $\mathcal{T}^*$. That is, $\mathcal{T}^*$ can be obtained from $\mathcal{T}_1$ by contracting edges. Recall that the tree $\mathcal{T}^*$ is the unique most refined tree on $\mathcal{X}$ such that no tf-quarnet in $\mathcal{Q}$ contradicts a split of $\mathcal{T}^*$. This means that if we keep contracting the least supported split of $\mathcal{T}_1$ (see equation (4) in Step A3 of Section 4.3), we eventually end up with $\mathcal{T}^*$. Therefore, the tree $\mathcal{T}^*$ (and thus the blobtree of $\mathcal{N}$) is part of the sequence $(\mathcal{T}_1, \dots, \mathcal{T}_{n-3})$.

Next, we show that given the blobtree $\mathcal{T}^*$ of $\mathcal{N}$ (and assuming the set $\mathcal{Q}$ is induced by $\mathcal{N}$), Step B of SQUIRREL correctly constructs the network $\mathcal{N}$. We first show this is true when $\mathcal{N}$ is a sunlet network (and so $\mathcal{T}^*$ is an unresolved tree with a single internal vertex). Then, the set $\tilde{\mathcal{Q}}_v$ constructed in Step B1 is the same as the original set $\mathcal{Q}$ of tf-quarnets (up to the relabeling defined by f). From Lemma A.5 we then know that the optimal TSP-tour using the distances $D_{\tilde{\mathcal{Q}}_v}$ will correspond to the circular ordering induced by $\mathcal{N}$. Whenever the sunlet network $\mathcal{N}$ has at least five leaves, only the leaf below the reticulation will appear in every 4-cycle tf-quarnet induced by $\mathcal{N}$. On the other hand, if $\mathcal{N}$ has only four leaves it only induces one tf-quarnet: a 4-cycle with the correct leaf below its reticulation. Hence, SQUIRREL correctly picks the reticulation vertex in Step B3. Thus, in the case that $\mathcal{N}$ is a sunlet network SQUIRREL constructs $\mathcal{N}$ from $\mathcal{T}^*$.

Whenever $\mathcal{N}$ is not a sunlet network, the proof is similar. To be precise, for every internal vertex v with induced partition $Y_1 | \dots | Y_s$, let $\tilde{y}_i$ be an arbitrary leaf of Y_i. Then, up to the relabeling defined by f, the set $\tilde{\mathcal{Q}}_v$ (as constructed in Step B1) is equal to the set of tf-quarnets induced by the sunlet network $\mathcal{N}|_{\{\tilde{y}_1, \dots, \tilde{y}_s\}}$ (the restriction of $\mathcal{N}$ to $\{\tilde{y}_1, \dots, \tilde{y}_s\}$).

Using a similar argument as above, Step B3 then correctly reconstructs the cycle in the sunlet network $\mathcal{N}|_{\{\tilde{y}_1,\dots,\tilde{y}_s\}}$. From this, it follows that we replace v in $\mathcal{T}^*$ by the correct cycle.

Since the blobtree $\mathcal{T}^*$ of $\mathcal{N}$ was part of the sequence $(\mathcal{T}_1,\dots,\mathcal{T}_{n-3})$, exactly one of the candidate networks $(\mathcal{N}_1,\dots,\mathcal{N}_{n-3})$ will be our original network $\mathcal{N}$. By Frohn et al. (2024) we know that tf-quarnets are enough to encode a triangle-free semi-directed level-1 network. Hence, this will be the only network with a (weighted) tf-quarnet consistency score of 1 and so SQUIRREL returns the correct network $\mathcal{N}$. □

Whereas in Theorem 1 we showed that SQUIRREL reconstructs a network from its set of tf-quarnets, in practice, the input will not be a set of tf-quarnets coming from one unique network. In the following lemma we prove that even if the input tf-quarnets do not come from one unique network, SQUIRREL is still guaranteed to return a triangle-free semi-directed level-1 network.

Lemma A.6. *Let $\mathcal{Q}$ be a dense set of weighted tf-quarnets on $\mathcal{X}$, then* SQUIRREL *applied to $\mathcal{Q}$ returns a triangle-free semi-directed level-1 network on $\mathcal{X}$.*

Proof. Step A (Algorithm 1) of SQUIRREL relies on algorithms from Berry and Gascuel (2000) and Grünewald et al. (2009). It follows from those papers that Algorithm 1 always returns a sequence of phylogenetic trees on $\mathcal{X}$. To prove the lemma, we now need to show that Step B of SQUIRREL (Algorithm 2) returns a triangle-free semi-directed level-1 network on $\mathcal{X}$ for any phylogenetic tree $\mathcal{T}$ on $\mathcal{X}$. Since Algorithm 2 only alters the tree $\mathcal{T}$ by replacing internal vertices of degree at least 4 by cycles with reticulations, the resulting network will always be level-1 and triangle-free. It remains to prove that the resulting network is a *valid* semi-directed network (i.e. a network that has a valid root-location, or equivalently, a network with no two reticulations oriented towards each other or towards the optional outgroup).

Clearly, when the first high-degree node is replaced by a cycle in the first iteration of the algorithm, there always is a location for the reticulation vertex that results in a partial network with a valid root-location. We can thus inductively assume that at the start of the other iterations, when we want to replace some internal vertex v by a cycle, the partial network constructed in the previous iteration has a valid root-location at some edge (or in case of a specified outgroup, the specific edge incident to the outgroup). Hence, the partial network can be rooted at this edge, forming a non-binary rooted phylogenetic network. We can then expand the high-degree node in this directed acyclic graph that corresponds to v in such a way that the resulting rooted phylogenetic network remains a directed acyclic graph. Thus, when we replace v by a cycle in the partially constructed semi-directed network, there is a location for the reticulation such that the partial network remains valid, which proves the lemma. □

B Random network generator

In this section we provide a concise description of the algorithm that was used to generate the random semi-directed level-1 networks in Section 2.1. The algorithm takes as input a number of leaves n and a number of reticulations $r \geq 1$. In the special case that $r = 0$, we simply generate a random network with $r = 1$ reticulation and randomly turn it into a phylogenetic tree by deleting one of the reticulation edges (and suppressing the resulting degree-2 vertex). We first outline the general structure of the algorithm, while in the last paragraph we explain how the reticulation number is enforced. We emphasize that this algorithm is able to generate any semi-directed level-1 network on n leaves and with k reticulations, up to the labeling of the leaves.

The algorithm starts by building a random (possibly non-binary) phylogenetic tree on n leaves. In particular, it first generates a random tree on n vertices (by constructing a random spanning tree on a complete graph with n vertices), after which it attaches one leaf to every vertex. By suppressing all degree-2 vertices this results in a (possibly non-binary) tree with n leaves. Such a tree is transformed into a triangle-free semi-directed level-1 network by replacing every internal vertex of degree at least 4 by a random cycle. Finally, a random vertex of the cycle is assigned as a reticulation (while ensuring the resulting semi-directed network has a valid root location).

To enforce the correct reticulation number r in the resulting network, we ensure that the initial tree has exactly r internal vertices of high degree (i.e. a degree of at least 4). To this end, we iteratively adjust the tree until it satisfies this condition. In particular, if our initial tree has more than r high degree internal vertices, we contract the unique path between two random high degree internal vertices (without another high degree vertex on this path). This decreases the number of internal vertices with high degree by exactly one. On the other hand, if our initial tree has less than r of these high-degree internal vertices, we contract a random edge between two degree-3 nodes to create a new high degree vertex. This process is repeated until the tree has exactly r high-degree internal vertices.