Programs for Verification of Phylogenetic Networks

Andreas D.M. Gunawan[1], Bingxin Lu[2], Hon Wai Leong[2], and Louxin Zhang[1]
Departments of [1]Mathematics and [2]Computer Science, Nat'l University of Singapore, Singapore

Introduction

A tree has been used for a long time to model evolutionary history, where each internal node represents the emergence of a new species. However, tree model cannot explain reticulate evolution events such as hybridization, which drives the research on phylogenetic network (phylonet). Tree containment problem (TCP) and cluster containment problem (CCP) were formulated in order to compare a phylonet with a tree model.

Box 1 | Basic Terminologies

- A phylonet is a acyclic directed graph, in which every node other than the root has either indegree one (tree node) or outdegree one (reticulation). The leaves are bijectively labeled by a set of taxa. Example can be found in Figure 1.
- A phylogenetic tree (phylotree) is simply a phylonet without reticulation.
- A node is visible, if there is a leaf below it such that every path from the network root to the leaf passes through the node.

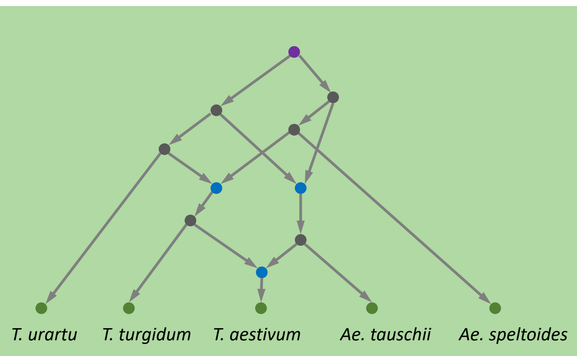

Figure 1 | Phylonet model for hybridization of *Triticum* and *Aegilops* wheat[1]. It contains a network root (purple), three reticulation nodes (blue), six internal non-root tree nodes (grey), and five leaves (green).

Phylonet decomposition

Removing all reticulations in a phylonet N yields a forest consisting of non-reticulation nodes. Each component in the forest is also called a component of N. A component is trivial, if it consists of a single leaf. A nontrivial component is exposed if every component below it is trivial. A component is visible if its root is visible. *Every component root in a reticulation-visible phylonet is visible*.

An example of phylonet decomposition can be found in Figure 2.

Formulation of two containment problems

Phylonet N displays phylotree G if N contains a subtree T that is a subdivision of G (i.e. G can be obtained from T by repeatedly removing unlabeled leaf and replacing node of indegree and outdegree one with an edge). TCP asks: *"Does a given phylonet display a given phylotree?"*

Phylonet N displays taxa set B, if there is a subtree of N in which B is the cluster of a node (i.e. set of labeled leaves below the node). CCP asks: *"Does a given phylonet display a given taxa set?"*

Illustration of TCP and CCP can be found in Figure 2.

Solving TCP and CCP

The algorithms for solving TCP or CCP is presented in Box 2. It calls the REDUCTION procedure (Box 3) as a subroutine, because the subalgorithm from [2] (Box 2, step 3) only works if the chosen exposed component is visible. Each call on REDUCTION procedure will call an extra TCP/CCP algorithm while continuing the current algorithm with a phylonet where the chosen component becomes visible. An illustration of the algorithm can be found in Figure 3.

Box 2 | Outline of TCP and CCP Algorithms

1. Decompose the phylonet into its components.
2. Choose an exposed component C.
3. If C is visible, call a subalgorithm from [2] to either deduce the answer or contract C into a leaf.
4. Else, call REDUCTION procedure (Box 3) and repeat step 3.
5. Repeat step 2 if there is another exposed component. Otherwise, conclude with positive answer.

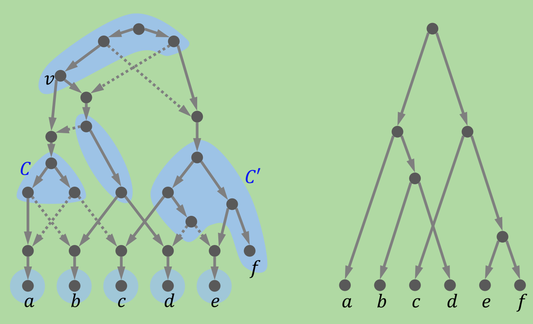

Figure 2 | Phylonet decomposition and the containment problems. The reticulation-visible phylonet on the left has nine tree node components, five of which are trivial. The components C and C' are exposed. Removing the dotted edges results in a subtree that is a subdivision of the phylotree on the right. The subtree also shows that $B = \{a, b, d\}$ is a displayed taxa set, as B is the cluster of v.

Box 3 | REDUCTION procedure

1. Choose a leaf ℓ below C and define the followings:
   $R = \text{parent}(\ell) \cup \{\text{ reticulation } r \mid \text{child}(r) \in R\ \}$,
   $E = \{\text{ edge } (u, v) \mid u \text{ is a tree node and } v \in R\ \}$, and
   $E_C = \{\text{ edge } (u, v) \mid\ u \in C\ \}$.
2. Call a new CCP algorithm using $N - E_C$ as input.
3. Continue the current algorithm using $N\ -(E \backslash E_C)$ as input.

Finding a tight bound for the running time is not easy, but for a restricted input, we have the following theorem (see [3,4]).

Theorem: *The algorithm runs in $O\left(2^{0.694 \cdot r} poly(|N|)\right)$ time if the input phylonet is bicombining and reduced.*

In contrast, a naïve algorithm considers every possible subtree of N, and thus runs in $O\left(2^r \cdot \text{poly}(|N|)\right)$ time.

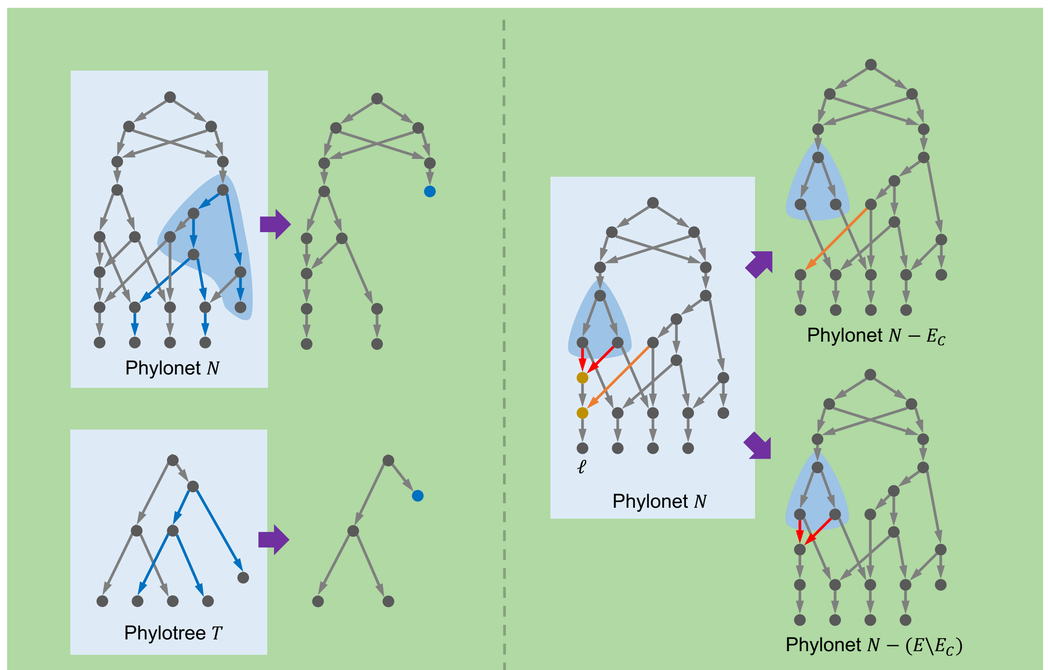

Figure 3 | Exposed component contraction in TCP. Let C be the highlighted exposed component. **(left)** If C is visible, then we run a subalgorithm from [2], which finds a maximal subtree of G displayed below the component (blue edges), and contracts the exposed component (along with a subtree of the input phylotree) into a single leaf. **(right)** Otherwise, we call the REDUCTION procedure. Choose a leaf ℓ, define the reticulation set R (yellow nodes), edge set E (red and orange edges) and E_C (red edges), call a new TCP algorithm with input $N - E_C$ and the current phylotree, and continue the current algorithm with $N - (E \backslash E_C)$.

References

1. Marcussen, Thomas, et al. *Science* 345.6194 (2014): 1250092.
2. Gunawan, Andreas DM, Bhaskar DasGupta, and Louxin Zhang. *Information and Computation* 252 (2017): 161-175.
3. Gunawan, Andreas DM, Bingxin Lu, and Louxin Zhang. *Bioinformatics* 32.17 (2016): i503-i510.
4. Lu, Bingxin, Louxin Zhang, and Hon Wai Leong. *BMC Genomics* 18.2 (2017): 111.

Funding

The Singapore MOE 2014-T2-1-155 and a Tier 1 grant R-146-000-238-114.

Download information

TCP algorithm is available on:
http://phylnet.univ-mlv.fr/tools/treeContainment.php
CCP algorithm is available on:
https://github.com/icelu/PlyloNetwork