BMC Bioinformatics

Software

PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships

Cuong Than*†, Derek Ruths† and Luay Nakhleh*†

Address: Department of Computer Science, Rice University, 6100 Main Street, MS 132, Houston, TX, USA

Email: Cuong Than* - cvthan@cs.rice.edu; Derek Ruths - druths@cs.rice.edu; Luay Nakhleh* - nakhleh@cs.rice.edu

* Corresponding authors †Equal contributors

Published: 28 July 2008

BMC Bioinformatics 2008, **9**:322 doi:10.1186/1471-2105-9-322

Received: 4 March 2008
Accepted: 28 July 2008

This article is available from: http://www.biomedcentral.com/1471-2105/9/322

Abstract

Background: Phylogenies, i.e., the evolutionary histories of groups of taxa, play a major role in representing the interrelationships among biological entities. Many software tools for reconstructing and evaluating such phylogenies have been proposed, almost all of which assume the underlying evolutionary history to be a tree. While trees give a satisfactory first-order approximation for many families of organisms, other families exhibit evolutionary mechanisms that cannot be represented by trees. Processes such as horizontal gene transfer (HGT), hybrid speciation, and interspecific recombination, collectively referred to as *reticulate evolutionary events*, result in *networks*, rather than trees, of relationships. Various software tools have been recently developed to analyze reticulate evolutionary relationships, which include SplitsTree4, LatTrans, EEEP, HorizStory, and T-REX.

Results: In this paper, we report on the PhyloNet software package, which is a suite of tools for analyzing reticulate evolutionary relationships, or *evolutionary networks*, which are rooted, directed, acyclic graphs, leaf-labeled by a set of taxa. These tools can be classified into four categories: (1) evolutionary network representation: reading/writing evolutionary networks in a newly devised compact form; (2) evolutionary network characterization: analyzing evolutionary networks in terms of three basic building blocks – trees, clusters, and tripartitions; (3) evolutionary network comparison: comparing two evolutionary networks in terms of topological dissimilarities, as well as fitness to sequence evolution under a maximum parsimony criterion; and (4) evolutionary network reconstruction: reconstructing an evolutionary network from a species tree and a set of gene trees.

Conclusion: The software package, PhyloNet, offers an array of utilities to allow for efficient and accurate analysis of evolutionary networks. The software package will help significantly in analyzing large data sets, as well as in studying the performance of evolutionary network reconstruction methods. Further, the software package supports the proposed eNewick format for compact representation of evolutionary networks, a feature that allows for efficient interoperability of evolutionary network software tools. Currently, all utilities in PhyloNet are invoked on the command line.

Background

A phylogenetic tree models the evolutionary history of a set of taxa from their most recent common ancestor. The assumptions of strict divergence and vertical inheritance render trees appropriate for modeling the evolutionary histories of several groups of species or organisms. However, when *reticulate* evolutionary events such as horizontal gene transfer or interspecific recombination occur, the evolutionary history is more appropriately modeled by an evolutionary network.

Evidence of reticulate evolution has been shown in various domains in the Tree of Life. Bacteria obtain a large proportion of their genetic diversity through the acquisition of sequences from distantly related organisms, via horizontal gene transfer (HGT) [1]. Furthermore, more evidence of widespread HGT in plants is emerging recently [2-4]. Interspecific recombination is believed to be ubiquitous among viruses [5,6], and hybrid speciation is a major evolutionary mechanisms in plants, and groups of fish and frogs [7-10]. All of these processes result in networks, rather than trees, of evolutionary relationships, even though at the gene level evolutionary histories may be treelike, as we now describe.

Figure 1 illustrates the three major events that result in networks of evolutionary relationships among species, namely horizontal gene transfer, interspecific recombination, and hybrid speciation. The tubes depict the evolutionary network of the species, within which two gene trees are shown. In each box, the two possible gene trees *T* and *T*' are shown separately, as well as the network *N* at an abstract level. In an evolutionary scenario involving horizontal transfer, certain sites (specified by a specific substring within the DNA sequence of the species into which the horizontally transferred DNA was inserted) are inherited from another species (the tree *T*' in dashed lines in Figure 1(a)), while all others are inherited from the parent (the tree *T* in solid lines in Figure 1(a)). Thus, each site evolves down one of the trees contained inside the network.

In the case of interspecific recombination, as illustrated in Figure 1(b), some genetic material is exchanged between pairs of species; in this example, species *B* and *C* exchange genetic material. The genes involved in this exchange have an evolutionary history (gene tree *T*') that is different from that of the genes that are vertically transmitted (gene tree *T*).

In the case of hybrid speciation, as illustrated in Figure 1(c), the two parents contribute equally to the genetic material of the hybrid: in *diploid* hybridization, each parent contributes a single copy of each of its chromosomes, while in *polyploid* hybridization, each parent contributes all copies of its chromosomes. Thus, each set of "orthologous sites" from all taxa has an evolutionary history that is depicted by one of the trees inside the network.

A few software tools for analyzing reticulate evolutionary relationships have been developed recently. The SplitsTree4 tool, which incorporates several algorithms that have been developed by Daniel Huson and his coworkers, is a tool for reconstructing and visualizing splits networks [11]. The tool enables constructing networks from several types of data, including sequence data, distance matrices, and sets of trees. Two major differences exist between SplitsTree4 and PhyloNet. First, SplitsTree4 constructs and analyzes splits networks, which are graphical models of incompatibility in the data, whereas PhyloNet constructs and analyzes evolutionary networks, which are rooted, directed, acyclic graphs, that represent evolutionary relationships. Second, the two tools differ in the utilities they provide, and we view them as complementary. While SplitsTree4 is mainly aimed at recon-

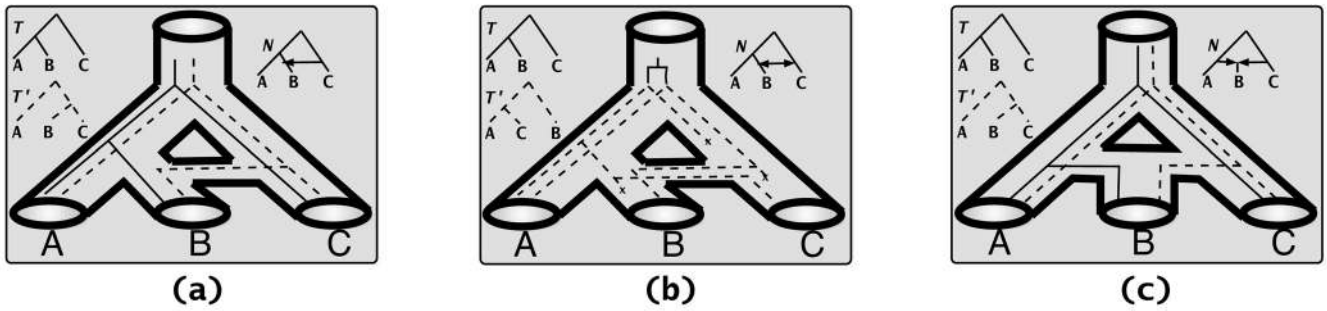

Figure 1

Evolutionary networks and gene trees. Gene trees *T* and *T*' within species networks *N*. (a) The gene whose tree is depicted with a dashed line is transferred from the genome of species *C* to that of species *B*. (b) Species *B* and *C* exchanged the two genes whose trees are *T* and *T*'. (c) Species *B* is a hybrid whose two parents are species *A* and *C*; each gene in the genome of species *B* has an evolutionary tree that is either *T* or *T*'.

structing networks, PhyloNet has several utilities for evaluating networks.

Programs such as EEEP [12], HorizStory [13], LatTrans [14], and T-REX [15] are aimed at detecting horizontal gene transfer by reconciling a pair of species/gene trees. The PhyloNet software package that we developed contains an extended implementation of the RIATA-HGT algorithm [16] with several improved algorithmic techniques for computing multiple solutions and handling non-binary trees [17]. The new version of RIATA-HGT significantly outperforms, in terms of speed, EEEP, HorizStory and LatTrans, and performs at least as well in terms of accuracy [17,18]. We have recently added a new heuristic for inferring the support of HGT moves from bootstrap values of gene tree edges. Further, we have added the capability of visualizing the networks computed by RIATA-HGT. Besides RIATA-HGT, the PhyloNet software package implements methods for comparing and characterizing evolutionary networks, which include: (1) evolutionary network representation: reading/writing evolutionary networks in a newly devised compact form; (2) evolutionary network characterization: analyzing evolutionary networks in terms of three basic building blocks – trees, clusters, and tripartitions; (3) evolutionary network comparison: comparing two evolutionary networks in terms of topological dissimilarities, as well as fitness to sequence evolution under a maximum parsimony criterion; and (4) evolutionary network reconstruction: reconstructing an evolutionary network from a species tree and a set of gene trees. Furthermore, since various evolutionary network utilities use functionalities from the phylogenetic trees domain, PhyloNet provides a set of standalone phylogenetic tree analysis tools.

Results and discussion

The evolutionary network model

In this paper, we assume the "evolutionary network" model, which was formulated independently by Moret *et al*. [19] and Baroni *et al*. [20]. We now describe the model as well as some basic definitions and notations that we will use later.

Let $T = (V, E)$ be a tree, where V and E are the *tree nodes* and *tree edges*, respectively, and let $L(T)$ denote the tree's leaf set. Further, let χ be a set of taxa (organisms). Then, T is a *phylogenetic tree* over χ if there is a bijection between χ and $L(T)$. Henceforth, we will identify the taxa set with the leaves they are mapped to, and let $[n] = \{1,..., n\}$ denote the set of leaf-labels. A tree T is said to be *rooted* if the set of edges E is directed and there is a single node $r \in V$ with in-degree 0. Let T be a phylogenetic tree on set χ of taxa, and let $\chi' \subseteq \chi$ be a subset of taxa; then, we denote by $T|\chi'$ the subtree with minimum number of nodes and edges that spans the leaves in χ' (in other words, $T|\chi'$ is the tree T restricted to subset χ' of its leaves).

An evolutionary (phylogenetic) network $N = (V, E)$ over the set χ of taxa is a rooted, directed, acyclic graph such that there is a bijection between χ and the set $L(N)$ of the network's leaves (see Figure 2). The set V is partitioned into two sets: V_T, the set of *tree nodes*, which are the nodes with in-degree smaller than two, and V_N, the set of *network nodes*, which are the nodes with in-degree greater than or equal to two. Similarly, the set E is partitioned into two sets: E_T, the set of *tree edges*, which are edges incident into tree nodes, and E_N, the set of *network edges*, which are the edges incident into network nodes.

For two nodes u and v in directed graph G, we say that v is reachable from u, denoted by $u \rightsquigarrow v$ if there exists a directed path from u to v in the tree G. For three nodes u, v and x in directed graph G, we write $u \rightsquigarrow^{[x]} v$ if all directed paths from u to v go through node x; $u \not\rightsquigarrow^{[x]} v$ if no directed paths from u to v go through node x; and $u \rightsquigarrow^{\{x\}} v$ if at least one directed path from u to v goes through node x and at least one directed path from u to v does not go through node x. For example, in network N_1 in Figure 2, rooted at node r_1, we have $r_1 \rightsquigarrow^{[Y]} D$, $r_1 \not\rightsquigarrow^{[Y]} E$, and $r_1 \rightsquigarrow^{[X]} D$.

Evolutionary network representation

The Newick format for representing and storing phylogenetic trees was adopted in 1986 [21], and it has been the standard for almost all phylogeny software packages ever since. This format captures an elegant correspondence

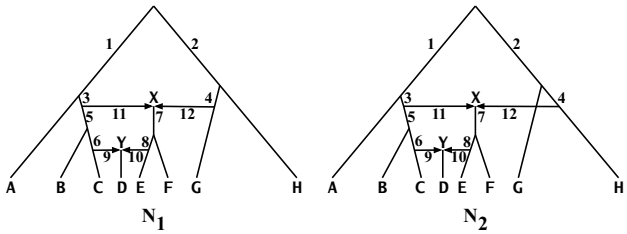

Figure 2

Sample evolutionary networks. Two evolutionary networks N_1 and N_2, each with eight leaves (labeled *A*,..., *H*) and two network nodes *X* and *Y*. Shown are the orientation of the network edges; all other edges are directed away from the root (toward the leaves) Notice that the difference between the two networks is that node *X* in N_1 has lineage *G* as one of its parents, whereas node *X* in N_2 has lineage *H* as one of its parents.

between leaf-labeled trees and matched parentheses, where the leaves are represented by their names and the internal nodes by a matched pair of parentheses that contains a list of the Newick representation of all its children. Shown in Figure 3 are three trees along with their representations in the Newick format.

Existing phylogenetic network software tools store these networks as adjacency lists of their underlying graphs, which are usually very large and necessitate translation of representations among the different tools. Morin and Moret [22] proposed a modified version of the Newick format for representing such networks. In their format, network nodes are represented by nodes labeled with #H, and those nodes are considered as two separate nodes in the normal Newick format for trees. See Figure 4 for an example. We have independently proposed a new method of *tree decomposition* of evolutionary networks, which provides the basis for a new format, *extended Newick* (or eNewick for short), and used it as a compact representation of evolutionary networks. The idea in our method is to break the network into a set of trees, and then represent the network as a collection of Newick representations of those trees. Since the eNewick format is nothing but a collection of trees in the Newick format, it follows that eNewick can represent unrooted networks. However, both in this paper as well as in the PhyloNet utilities, rooting is assumed, since different ways of rooting the same evolutionary networks may imply different evolutionary relationships.

Let $N = (V = (V_N \cup V_T); E)$ be an evolutionary network, with $|V_N| =$. We create a forest of trees as follows.

• For every $u_i \in V_N$

- Compute the set $V_i = \{v \in V : (v, u_i) \in E\}$ of u_i's parents;

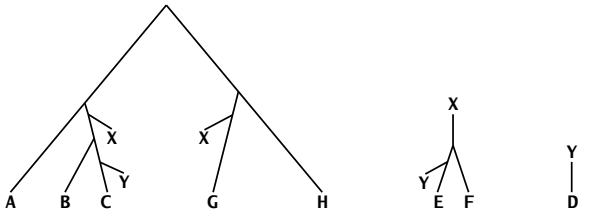

Figure 3

A modified Newick format. Three trees, *N*', *X*, and *Y*, along with their Newick representation. These three trees form the tree decomposition $\mathcal{F}$ of the evolutionary network N_1 in Figure 2. The eNewick representation of *N* is the triplet $N'; X; Y$.

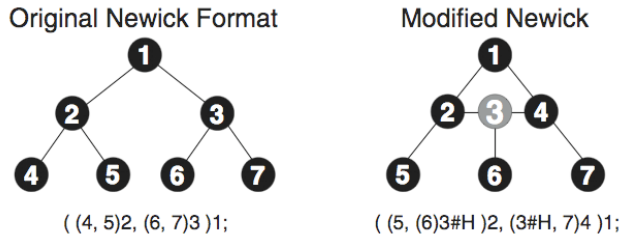

Figure 4

The eNewick format. A modified Newick format for representing evolutionary networks. The figure is taken from the paper by Morin *et. al.* [22].

- Create k new leaves, all labeled with x_i ($\{x_i\} \cap L(N) = \emptyset$);

- Delete from V the set of all edges in $V_i \times \{u_i\}$;

- Add to V the set of edges $V_i \times \{x_i\}$;

- Assign x_i as the name of the tree rooted at node u_i;

The result is a forest of trees $\mathcal{F} = \{t_1,..., t\}$ such that (1) $|L(t_i)| \geq 1$ for every $1 \leq i \leq$, (2) $\bigcup_{i=1}^{\ell} L(t_i) = L(N)$ and (3) $L(t_i) \cap L(t_j) = \emptyset$ for every $1 \leq i, j \leq$ and $i \neq j$. We call $\mathcal{F}$ the *tree decomposition* of *N*. Then, the eNewick representation of *N* is the -tuple $n(t_1);...; n(t)$, where $n(t_i)$ is the Newick representation of tree t_i. Figure 3 shows the tree decomposition and eNewick representation of the network N_1 in Figure 2.

In the case of modeling networks with horizontal gene transfer events, it is often very helpful to the biologist to know what the species tree is and what the additional set of HGT events are. Such information is "lost" in an eNewick representation, unless the representation is extended further to keep a record of the "species tree parent" of each network node. Therefore, in this case (which is the output of RIATA-HGT) we opt for the format of a species tree *T*, in Newick format, followed by a list of the HGT edges, each written as $X \rightarrow Y$, where *X* and *Y* are two nodes in *T*.

Evolutionary network characterization

As we described in the background section, an evolutionary network induces, or contains, a set of trees. We now formalize this concept and characterize networks in terms of the trees they induce. A tree *T* is induced by a network *N* if *T* is obtained from *N* as follows: (1) for each node of in-degree larger than one, remove all but one of the network edges incident into it, and (2) for every node of in-degree and out-degree 1, and whose parent is *u* and child

is v, remove the two edges incident with it, and add an edge from u to v. We denote by $\mathcal{T}(N)$ the set of all trees induced by N. Figure 5 shows the sets $\mathcal{T}(N_1)$ and $\mathcal{T}(N_2)$ for the two networks N_1 and N_2 in Figure 2. It is important to note that this set of trees is completely different from the set of trees obtained by the tree decomposition we introduced to facilitate the eNewick format. An evolutionary network N with $V_N = \{v_1,..., v\}$, such that *indegree*$(v_i) = \rho_i$, induces m trees, where $m \leq \prod_{i=1}^{\ell} \rho_i$ Given an evolutionary network N, the set $\mathcal{T}(N)$ is unique. Further, this set informs about the possible gene histories that the network reconciles.

In addition to characterizing evolutionary networks by the set of trees they induce, we consider a *cluster*-based characterization. This view of evolutionary networks is very important for understanding the relationships among the "evolutionary perspective" of evolutionary networks and the "cluster, or splits, perspective", which is adopted in various methods [23,24]. Let $T = (V, E)$ be a phylogenetic tree on set χ of taxa and rooted at node r. Each edge $e = (u, v) \in E$ induces a *cluster* of taxa, denoted c_e, which is the set $\{x \in \chi : r \rightsquigarrow^{[v]} x\}$. The (nontrivial) clusters of tree T is the set $C(T) = \{c_e : e \text{ is an internal edge in } E\}$. The topology of a tree is a compact graphical representation of its clusters, where the root of the clade that corresponds to cluster $c_{e'}$ lies on the path from the root of the tree to the root of the clade that corresponds to cluster c_e if and only if $c_e \subseteq c_{e'}$. Hence, clusters play an important role in phylogenetic tree characterization and reconstruction. A straightforward way to extend this concept to evolutionary networks is to define the set of clusters of evolutionary network N as $C(N) = \bigcup_{T \in \mathcal{T}(N)} C(T)$. The clusters of the two networks N_1 and N_2 in Figure 2 are listed in Table 1.

In this form of cluster-based characterization, clusters are unweighted; equivalently, all clusters are weighted equally. One option of weighting the clusters is by considering the fraction of trees in which it appears. In other words, the weight of a cluster c_e can be computed as

$$w(c_e) = \frac{|\{T \in \mathcal{T}(N) : c_e \in C(T)\}|}{|\mathcal{T}(N)|}.$$

This weighting scheme informs not only about the clusters of taxa that the network represents, but also how many gene trees in the input share each cluster. It is important to note here that this weighting of a cluster should not be confused with, or used in lieu of, support values of clusters, since a cluster may appear in only one gene tree and have a high support (e.g., by having a high bootstrap value on the edge that defines it) whereas a poorly supported cluster may appear in several trees.

Nakhleh and colleagues have recently introduced a new characterization of evolutionary networks based on the *tripartitions* of their edges [19]. Let $e = (u, v)$ be an edge in

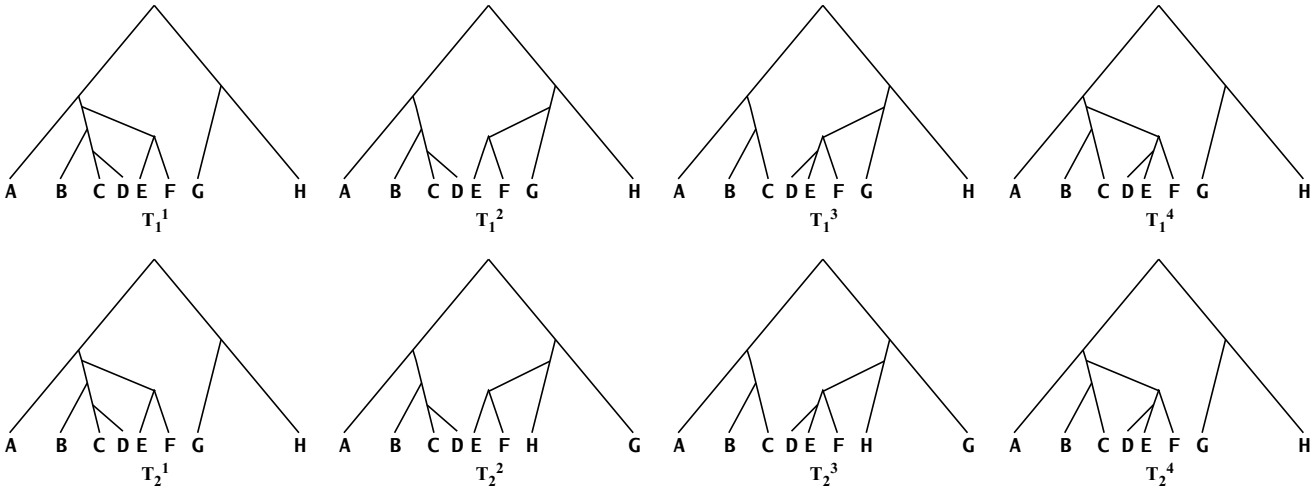

Figure 5

Trees within networks. The sets $\mathcal{T}(N_1) = \{T_1^1, T_1^2, T_1^3, T_1^4\}$ and $\mathcal{T}(N_2) = \{T_2^1, T_2^2, T_2^3, T_2^4\}$ of all eight trees induced by the two networks N_1 and N_2, respectively, in Figure 2.

Table 1: The clusters of the two networks in Figure 2.

Network N_1	Network N_2
{*B*, *C*}	{*B*, *C*}
{*C*, *D*}	{*C*, *D*}
{*B*, *C*, *D*}	{*B*, *C*, *D*}
{*D*, *E*}	{*D*, *E*}
{*E*, *F*}	{*E*, *F*}
{*D*, *E*, *F*}	{*D*, *E*, *F*}
{*B*, *C*, *D*, *E*, *F*}	{*B*, *C*, *D*, *E*, *F*}
{*A*, *B*, *C*}	{*A*, *B*, *C*}
{*A*, *B*, *C*, *D*}	{*A*, *B*, *C*, *D*}
{*A*, *B*, *C*, *D*, *E*, *F*}	{*A*, *B*, *C*, *D*, *E*, *F*}
{**E**, **F**, **G**}	{**E**, **F**, **H**}
{**D**, **E**, **F**, **G**}	{**D**, **E**, **F**, **H**}
{*G*, *H*}	{*G*, *H*}
{*D*, *E*, *F*, *G*, *H*}	{*D*, *E*, *F*, *G*, *H*}

A table of the (nontrivial) clusters of the two networks N_1 and N_2 in Figure 2, denoted by $\mathcal{C}(N_1)$ and $\mathcal{C}(N_2)$, respectively, in the text. Highlighted are rows corresponding to clusters that differ between the two networks.

an evolutionary network on set χ of taxa and rooted at node r. We define three disjoint sets $A_e = \{x \in \chi : r \rightsquigarrow^{[v]} x\}$, $B_e = \{x \in \chi : r \rightsquigarrow^{[v]} x\}$, and $C_e = \{x \in \chi : r \not\rightsquigarrow^{[v]} x\}$. Then, the tripartition induced by edge e, denoted θ_e, is the triplet $A_e; B_e; C_e$. Roughly speaking, the tripartition induced by an edge is the three sets of taxa reachable from the root only through that edge (A_e), reachable through that edge but not exclusively (B_e), and not reachable through that edge (C_e). The set of (nontrivial) tripartitions induced by a evolutionary network N, denoted by $\theta(N)$, is $\{\theta_e : e \text{ is an internal edge in } E\}$. The tripartitions of the two networks N_1 and N_2 in Figure 2 are listed in Table 2.

Tripartition-based characterization of an evolutionary network helps to identify clades across which no genetic transfer occurred. If $A_e = X$ and $B_e = \emptyset$ for an edge $e = (u, v)$, this implies that the clade rooted at node v has set X of leaves, and there does not exist any exchange or transfer of genetic material between any organism in X and another organism that is not in X. Equivalently, an evolutionary network can be partitioned into a collection $\{N_1, N_2, \ldots, N_k\}$ of evolutionary networks that result from N by deleting every edge e for which $B_e = \emptyset$. Such a partition informs about the "locality" of reticulation events: each event in N is local to one of the k components in $\{N_1, N_2, \ldots, N_k\}$. Further, this partition implies that each of the trees in $\mathcal{T}(N)$ has k clades that have the sets $\{L(N_1), L(N_2), \ldots, L(N_k)\}$ of leaves.

Evolutionary network comparison

Researchers are often interested in quantifying the similarities and differences between two phylogenies reconstructed either from two different sources of data or from two different reconstruction methods. Such a quantification provides insights into agreements and disagreements among analyses, confidence values for different parts of the phylogenies, and metrics for comparing the performance of phylogenetic reconstruction methods. In the context of phylogenetic trees, this quantification is most commonly done based on one of two criteria:

• *Topological differences*. The topologies, or shapes, of two phylogenetic trees are compared, and their differences are quantified. Several measures have been introduced to quantify topological differences and similarities between a pair of trees, such as the Robinson-Foulds measure and the SPR distance; see [25,26] for a description of several such measures.

Table 2: The tripartitions of the two networks in Figure 2.

Edge Label	Network N_1	Network N_2
1	{*A*, *B*, *C*}, {*D*, *E*, *F*}, {*G*, *H*}	{*A*, *B*, *C*}, {*D*, *E*, *F*}, {*G*, *H*}
2	{*G*, *H*}, {*D*, *E*, *F*}, {*A*, *B*, *C*}	{*G*, *H*}, {*D*, *E*, *F*}, {*A*, *B*, *C*}
3	{*B*, *C*}, {*D*, *E*, *F*}, {*A*, *G*, *H*}	{*B*, *C*}, {*D*, *E*, *F*}, {*A*, *G*, *H*}
4	{**G**}, {**D**, **E**, **F**}, {**A**, **B**, **C**, **H**}	{**H**}, {**D**, **E**, **F**}, {**A**, **B**, **C**, **G**}
5	{*B*, *C*}, {*D*}, {*A*, *E*, *F*, *G*, *H*}	{*B*, *C*}, {*D*}, {*A*, *E*, *F*, *G*, *H*}
6	{*C*}, {*D*}, {*A*, *B*, *E*, *F*, *G*, *H*}	{*C*}, {*D*}, {*A*, *B*, *E*, *F*, *G*, *H*}
7	{*E*, *F*}, {*D*}, {*A*, *B*, *C*, *G*, *H*}	{*E*, *F*}, {*D*}, {*A*, *B*, *C*, *G*, *H*}
8	{*E*}, {*D*}, {*A*, *B*, *C*, *F*, *G*, *H*}	{*E*}, {*D*}, {*A*, *B*, *C*, *F*, *G*, *H*}
9	{*D*}, {}, {*A*, *B*, *C*, *E*, *F*, *G*, *H*}	{*D*}, {}, {*A*, *B*, *C*, *E*, *F*, *G*, *H*}
10	{*D*}, {}, {*A*, *B*, *C*, *E*, *F*, *G*, *H*}	{*D*}, {}, {*A*, *B*, *C*, *E*, *F*, *G*, *H*}
11	{*E*, *F*}, {*D*}, {*A*, *B*, *C*, *G*, *H*}	{*E*, *F*}, {*D*}, {*A*, *B*, *C*, *G*, *H*}
12	{*E*, *F*}, {*D*}, {*A*, *B*, *C*, *G*, *H*}	{*E*, *F*}, {*D*}, {*A*, *B*, *C*, *G*, *H*}

A table of the (nontrivial) tripartitions of the two networks N_1 and N_2 in Figure 2, denoted by $\theta(N_1)$ and $\theta(N_2)$, respectively, in the text. Highlighted are rows corresponding to tripartitions that differ between the two networks.

• *Fitness to sequence evolution*. When two phylogenies are reconstructed from the same sequence data set, it is common to compare them in terms of how well they model the evolution of the sequences. The most commonly used criteria for measuring such fitness are maximum parsimony, maximum likelihood, and the Bayesian posterior probability; see [25] for a detailed discussion of all three criteria.

In this section, we report on the capabilities in PhyloNet for comparing two evolutionary networks in terms of their topological differences and similarities, as well as in terms of their fitness to sequence evolution based on the maximum parsimony criterion.

For quantifying the dissimilarity between two evolutionary network topologies N_1 and N_2, we want a measure $m(\cdot, \cdot)$ that satisfies three conditions:

Identity: $m(N_1, N_2) = 0$ if and only if N_1 and N_2 are *equivalent*;

Symmetry: $m(N_1, N_2) = m(N_2, N_1)$; and

Triangle inequality: $m(N_1, N_3) + m(N_3, N_2) \geq m(N_1, N_2)$ for any evolutionary network N_3.

This issue of evolutionary network equivalence was discussed in [19]. The three characterizations of evolutionary networks that we described above induce three measures which we now define. Let N_1 and N_2 be two evolutionary networks on the same set X of leaves; we define the three measures as follows.

Tree-based comparison

Let $\mathcal{T}(N_1)$ and $\mathcal{T}(N_2)$ be the two sets of all trees induced by the two networks, and let $d(\cdot, \cdot)$ be a distance metric on trees (see [26] for examples of such metrics). The idea is to compare the two networks based on how similar their corresponding sets of trees are. We formalize this as follows. Construct a weighted complete bipartite graph $G(U_1, U_2, E)$, where $|U_i| = |\mathcal{T}(N_i)|$, and there are two bijections $f_i : U_i \to \mathcal{T}(N_i)$ for $i = 1, 2$. The weight of an edge $e = (u, v) \in E$ for $u \in U_1$ and $v \in U_2$, $w(e) = d(f_1(u), f_2(v))$. Then, the tree-based measure $m^{tree}(N_1, N_2)$ is defined as the weight of a minimum-weight edge cover of G divided by, the number of the edges in the cover. In its current implementation, PhyloNet uses the Robinson-Foulds distance measure [27] for d. The tree-based measure was first introduced by Nakhleh *et al.* [28]. An illustration of tree-based comparison of the two networks N_1 and N_2 in Figure 2 is given in Figure 6. Shown on the left of Figure 6 is the bipartite graph G built from the sets $\mathcal{T}(N_1)$ and $\mathcal{T}(N_2)$ of trees induced by the two networks; these two sets are shown in Figure 5. The weight of each edge connecting two nodes in G is the RF distance between the two trees corresponding to these two nodes. These weights can be normalized by the number of internal edges in the trees. Since each of the eight trees has six internal edges, the weight of each edge in G can be divided by six to normalize it.

Shown on the right of Figure 6 is the minimum-weight edge cover of G, which is the set of edges that satisfies two conditions: (1) each node in G must be the endpoint of at least one edge in the set, and (2) the sum of the weights of the edges in the set is minimum among all sets of edges satisfying condition 1. In this case, the four edges shown are a cover, since each node in G is "covered" by at least one edge (here, each node is covered by exactly one edge). Further, it is of minimum weight, which equals 2, since a simple inspection yields that every other cover has a weight larger than 2. Since the cover has four edges in it, we have $m^{tree}(N_1, N_2) = (0 + 0 + 1/6 + 1/6)/4 = 1/12$. If we use the raw RF values, then $m^{tree}(N_1, N_2) = (0 + 0 + 1 + 1)/4 = 1/2$.

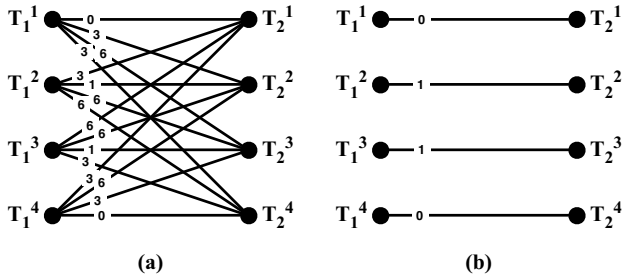

Figure 6

Tree-based comparison of networks. Illustration of the tree-based network comparison measure. (a) The weighted bipartite graph G that is constructed from the two networks N_1 and N_2 in Figure 2. On the left are four nodes that correspond to the four trees in $\mathcal{T}(N_1)$ and on the right are four nodes that correspond to the four trees in $\mathcal{T}(N_2)$. The weight of an edge between T_1^i and T_2^j is the values of the Robinson-Foulds (RF) distance between the two trees, which is computed as the number of clusters present in one but not both of the trees, divided by 2. (b) The edges that comprise the minimum-weight edge cover of the bipartite graph G. The weight of this cover is 2, which is the sum of the weights of the edges in the cover; therefore, $m^{tree}(N_1, N_2) = 2$.

Cluster-based comparison

Let $C_1 = C(N_1)$ and $C_2 = C(N_2)$ be the two sets of all clusters induced by the two networks. We define the measure based on these two sets to be

$$m^{cluster}(N_1, N_2) = \left(\frac{|C_1 - C_2|}{|C_1|} + \frac{|C_2 - C_1|}{|C_2|} \right) / 2.$$

The rationale behind this measure is that it is the sum of the ratios of clusters present in one but not both networks. The cluster-based measure was first introduced by Nakhleh *et al*. [29]. The sets $C_1 = C(N_1)$ and $C_2 = C(N_2)$ of the two networks N_1 and N_2 in Figure 2 are listed in Table 1, with $|C_1| = |C_2| = 14$. Since $|C_1 - C_2| = |C_2 - C_1| = 2$ (the two highlighted clusters in Table 1), we have $m^{cluster}(N_1, N_2) = 1/7$. A similar weighting scheme to that described in the previous section can be used to incorporate the fraction of trees in which a cluster appears into the measure calculation.

Tripartition-based comparison

Let $\theta_1 = \theta(N_1)$ and $\theta_2 = \theta(N_2)$ be the two sets of all tripartitions induced by the two networks. We define the measure based on these two sets to be

$$m^{tripartition}(N_1, N_2) = \left(\frac{|\theta_1 - \theta_2|}{|\theta_1|} + \frac{|\theta_2 - \theta_1|}{|\theta_2|} \right) / 2.$$

This measure views the two networks in terms of the sets of edges they define (where an edge is in a 1-1 correspondence with a tripartition) and computes the sum of the ratios of edges present in one but not both networks. The tripartition-based measure was devised by Moret *et al*. [19]. The sets $\theta_1 = \theta(N_1)$ and $\theta_2 = \theta(N_2)$ of the two networks N_1 and N_2 in Figure 2 are listed in Table 2, with $|\theta_1| = |\theta_2| = 12$. Since $|\theta_1 - \theta_2| = |\theta_2 - \theta_1| = 1$ (the highlighted tripartition in Table 2), we have $m^{tripartition}(N_1, N_2) = 1/12$.

Which measure to use?

Several distance measures, such as the Robinson-Fould measure and the Subtree Prune and Regraft (SPR) distance, have been introduced over the years to quantify the difference between the topologies of a pair of phylogenetic trees; e.g., see [25,26] for description of many of these measures. Even though these measures may compute different distance values on the same pair of trees, there has been no consensus as to which measure should be used in general [30]. It may be the case that the Robinson-Foulds measure is more commonly used than the others, but this may be a mere reflection of its very low time requirements as compared to the other, more compute-intensive, measures.

Regarding the three measures for comparing networks, a scenario analogous to that in phylogenetic trees arises here: each measure gives a different quantification of the dissimilarity between two networks based on one of the three ways to characterize a given network. As shown in the examples above, some or all of these measures may compute the same value for a given pair of networks, but that may not always be the case. Tree-based comparison of networks can be viewed as a method to quantify how similar, or dissimilar, two networks are in terms of their quality as a summary of a collection of trees. In some cases, even though two networks "look different," they may be identical in terms of the trees they induce – this is the issue of indistinguishability of networks from a collection of trees that Nakhleh and colleagues discussed in [19]. In such a case, using the tree-based comparison, or equivalently the cluster-based comparison, is most appropriate. However, if the similarity/dissimilarity of two networks means something close to an *isomorphism*, then the tripartition-based measure is more appropriate. However, it is important to note that none of the three measures described here is a metric on the general space of all evolutionary networks labeled by a given set of taxa.

A practical distinction among the three measures can be derived based on the methods used to infer the evolutionary history of the set of species under study. Methods such as SplitsTree [23] and NeighborNet [24] represent the evolutionary history as a set of splits, or clusters, hence making it more natural to use cluster-based comparison to study their performance. Methods such as RIATA-HGT [16] and LatTrans [14] compute evolutionary networks that are rooted, directed, acylic graphs, where internal nodes have an evolutionary implication in terms of ancestry. For these two methods, all three measures are appropriate. When the evolutionary history of a set of species is represented as a collection of its constituent gene trees, the tree-based measure is most appropriate.

Finally, a clear distinction can be made among the methods in terms of computational requirements. In their current implementations, the tripartition-based measure is very fast in practice, taking time that is polynomial in the size of the two networks. On the other hand, the tree- and cluster-based measures are much slower, taking time that is exponential in the number of network nodes in the two networks (since these measures compute explicitly all trees inside each of the two networks). In light of recent complexity results that we obtained [31], it is very likely that no polynomial-time algorithms exist for computing the tree- and cluster-based measures in general.

Parsimony of evolutionary networks

Nakhleh and colleagues have recently formalized a maximum parsimony (MP) criterion for evolutionary net-

works [32] and demonstrated its utility in reconstructing evolutionary networks on both biological and synthetic data sets [33]. In this section, we describe a PhyloNet utility that allows for comparing two evolutionary networks in terms of their fitness to the evolution of a sequence data set, based on the MP criterion. We first begin with a brief review of the MP criterion, based on the exposition in [32].

The relationship between an evolutionary network and its constituent trees, as described in the background section, is the basis for the MP extension to evolutionary networks.

Definition 1 *The Hamming distance between two equal-length sequences x and y, denoted by $H(x, y)$, is the number of positions j such that $x_j \neq y_j$.*

Given a fully-labeled tree T, i.e., a tree in which each node v is labeled by a sequence s_v over some alphabet Σ, we define the Hamming distance of an edge $e \in E(T)$, denoted by $H(e)$, to be $H(s_u, s_v)$, where u and v are the two endpoints of e. We now define the parsimony score of a tree T.

Definition 2 *The parsimony score of a fully-labeled tree T, is $\Sigma_{e \in E(T)} H(e)$. Given a set S of sequences, a maximum parsimony tree for S is a tree leaf-labeled by S and assigned labels for the internal nodes, of minimum parsimony score.*

The parsimony definitions can be extended in a straightforward manner to incorporate different site substitution matrices, where different substitutions do not necessarily contribute equally to the parsimony score, by simply modifying the formula $H(x, y)$ to reflect the weights. Let Σ be the set of states that the two sequences x and y can take (e.g., $\Sigma = \{A, C, T, G\}$ for DNA sequences), and W the site substitution matrix such that $W[\sigma_1, \sigma_2]$ is the weight of replacing σ_1 by σ_2, for every $\sigma_1, \sigma_2 \in \Sigma$. In particular, the *identity* site substitution matrix satisfies $W[\sigma_1, \sigma_2] = 0$ when $\sigma_1 = \sigma_2$, and $W[\sigma_1, \sigma_2] = 1$ otherwise. The weighted Hamming distance between two sequence is $H(x, y) = \Sigma_{1 \leq i \leq k} W(x_i, y_i)$, where k is the length of the sequences x and y. The rest of the definitions are identical to the simple Hamming distance case. As described above, the evolutionary history of a single (non-recombining) gene is modeled by one of the trees contained inside the evolutionary network of the species containing that gene. Therefore the evolutionary history of a site s is also modeled by a tree contained inside the evolutionary network. A natural way to extend the tree-based parsimony score to fit a dataset that evolved on a network is to define the parsimony score for each site as the minimum parsimony score of that site over all trees contained inside the network.

Definition 3 *([32]) The parsimony score of a network N leaf-labeled by a set S of taxa, is*

$$NCost(N, S) := \sum_{s_i \in S} (\min_{T \in \mathcal{T}(N)} TCost(T, s_i))$$

where $TCost(T, s_i)$ is the parsimony score of site s_i on tree T.

Notice that as usually large segments of DNA, rather than single sites, evolve together, Definition 3 can be extended easily to reflect this fact, by partitioning the sequences S into non-overlapping blocks b_i of sites, rather than sites s_i, and replacing s_i by b_i in Definition 3. This extension may be very significant if, for example, the evolutionary history of a gene includes some recombination events, and hence that evolutionary history is not a single tree. In this case, the recombination breakpoint can be detected by experimenting with different block sizes.

The MP utility in PhyloNet allows the user to specify two evolutionary networks (either or both of which can be a tree) N_1 and N_2 and a sequence data set S, and computes the parsimony scores $NCost(N_1, S)$ and $NCost(N_2, S)$. The user can then compare the two scores and evaluate the fitness of the networks to the data set S based on the difference in the scores. Further, the utility allows the user, for example, to evaluate the significance of each network edge in a network N by comparing the parsimony scores of two different versions of N that contain different subsets of the network edges in N.

Reconstructing evolutionary networks from species/gene trees

Assuming incongruence among gene and species trees is the result of HGT events only, the *Phylogeny-based HGT Reconstruction Problem*, or HGT Reconstruction Problem for short, is defined as follows:

Problem 1 *(HGT Reconstruction Problem)*

Input: *A species tree ST and a set $\mathcal{T} = \{T_1, ..., T_p\}$ of gene trees.*

Output: *An evolutionary network N, obtained by adding a minimal set of edges Ξ to T, such that N contains every tree $T_i \in \mathcal{T}$*

The minimization criterion is a reflection of Occam's razor: in the absence of any additional biological knowledge, HGT events should be used sparingly to explain data features otherwise explainable under a tree model. The problem of finding a minimum-cardinality set of HGT

events whose occurrence on species tree *ST* would give rise to the gene trees in set $\mathcal{T}$ is computationally hard [34]. In [16], Nakhleh *et al*. introduced an accurate, polynomial-time heuristic, RIATA-HGT, for solving the HGT Reconstruction Problem for a pair of species and gene trees (in other words, RIATA-HGT currently handles the case where $|\mathcal{T}| = 1$). In a nutshell, the method computes the maximum agreement subtree [35] of the species tree and each of the gene trees, and adds HGT edges to connect all subtrees that do not appear in the maximum agreement subtree. Theoretically, RIATA-HGT may not compute the minimum-cardinality set of HGT events; nonetheless, experimental results show very good empirical performance on synthetic as well as biological data [16].

Computing multiple solutions and the graphical output

RIATA-HGT was designed originally to compute a single solution to the problem, and was mainly aimed at binary trees. Later, Than and Nakhleh [17] extended the method to compute multiple solutions and to handle non-binary trees. These two features are very significant: the former allows biologists to explore multiple potential HGT scenarios, whereas the latter allows for analyzing trees in which some edges were contracted due to inaccuracies (see [36] for example). We have conducted an experimental study to compare the performance of RIATA-HGT with LatTrans [18]. Although RIATA-HGT and LatTrans [14] have almost the same performance in terms of the number of HGT solutions and the solution size, the former runs much faster than the latter.

For a compact representation of multiple solutions, we introduce four terms:

• An *event*: this is a single HGT edge, written in the form of $X \rightarrow Y$, where X and Y are two nodes in the species tree.

• A *subsolution*: this is an *atomic* set of events, which forms a part of an overall solution. In other words, either all or none of the events of a subsolution are taken in a solution.

• A *component*: a set of components and/or subsolutions. Two components at the same level of decomposition are independent, in that an element of each component is needed to form a solution.

• A *solution*: the union of a single element from each component at the highest level.

To illustrate these concepts, consider species tree $(((a, b), c), (d, (e, f)))$ and the gene tree $(((a, c), b), ((d, f), e))$. Observe, that each HGT event required to reconcile the two trees has both endpoints in the subtree $((a, b), c)$ or both endpoints in the subtree $(d, (e, f))$, and no HGT event has endpoints in both subtrees. In this case, RIATA-HGT divides the pair of trees into two pairs:

• Pair 1: $((a, b), c)$ and $((a, c), b)$

• Pair 2: $(d, (e, f))$ and $((d, f), e)$,

and solves the HGT Reconstruction Problem on each of the two pairs *independently*. The set of solutions of each pair is a component. Notice that for each pair there are three possible ways to reconcile them; each such way is a called a subsolution. Each subsolution is a set of events, which in this case is only one event. Figure 7(a) shows the screen captures of two graphical outputs that correspond to two solutions on this pair of trees. Notice that if a component can be further divided into independent components, RIATA-HGT would do so, which will result in components at different levels, with the largest components being at the highest level.

The compact representation of RIATA-HGT's output in terms of subsolutions and components is especially helpful when the number of solutions is large. RIATA-HGT also has an option to display all complete solutions. RIATA-HGT enumerates all complete solutions that are compactly represented as described in the preceding paragraphs. Each solution, which is a set of HGT events, along with the species tree defines an evolutionary network, which RIATA-HGT displays in the eNewick format. For example, for the trees $(((a, b), c), (d, (e, f)))$ and $(((a, c), b), ((d, f), e))$, RIATA-HGT outputs 9 different networks in the eNewick format, if RIATA-HGT's option for displaying complete solutions is on. Figure 7(b) shows the corresponding eNewick representations.

From the multiple comparisons between a species and a set of trees, RIATA-HGT offers a (strict) consensus network. For each pair of species tree and gene tree, RIATA-HGT computes a set of HGT events for reconciling them. To obtain the consensus network, RIATA-HGT retains only HGT events that appear in every set of solutions for every pair of species tree and gene tree. Those events are then added to the species to build the consensus network.

We note here that while offering a simple summary of solutions, this way of computing consensus networks may not be appropriate in general; work is under way to address this issue more properly.

Finally, RIATA-HGT may report '[time violation?]' next to an inferred HGT $X \rightarrow Y$. If this is the case, this indicates that node X lies on the path from Y to the root of the species tree. Theoretically, this indicates that two nodes that do not co-exist in time, X and Y in this case, shared genetic

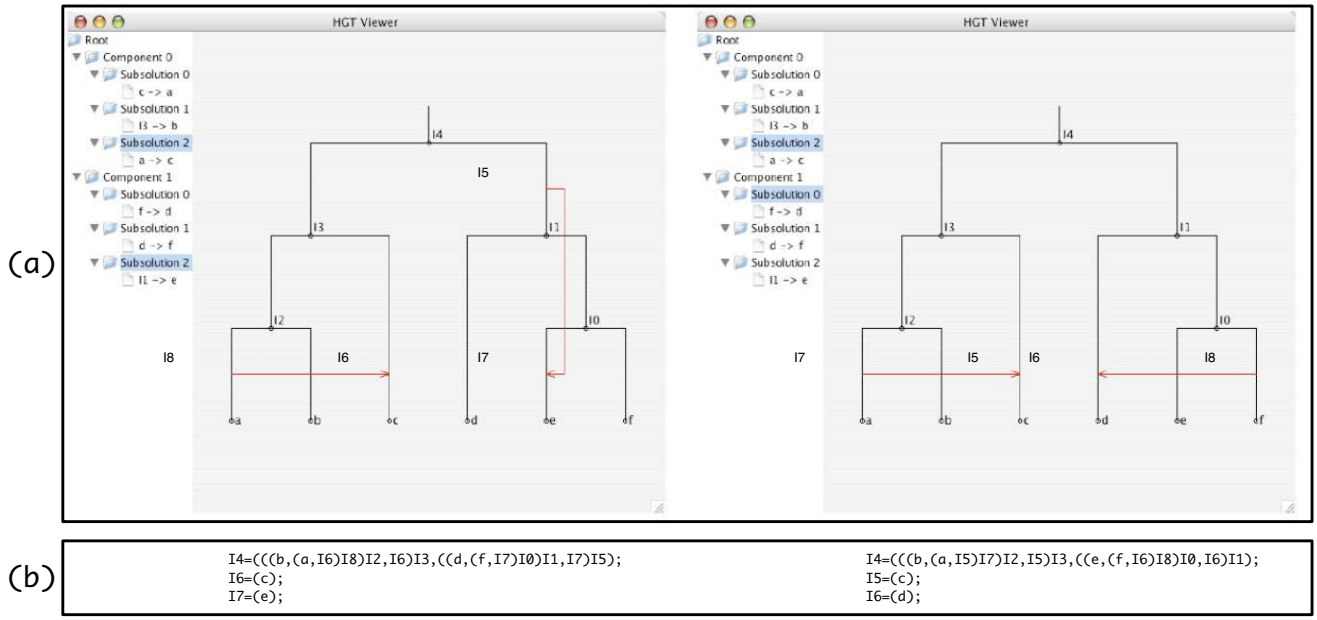

Figure 7
Screenshot of the graphical output of RIATA-HGT. (a) Screen captures of the graphical output of RIATA-HGT on the pair of trees (((*a*, *b*), *c*), (*d*, (*e*, *f*))) and (((*a*, *c*), *b*), ((*d*, *f*), *e*)). (b) The eNewick representations of the two selected networks.

material, and hence the warning of 'time violation.' However, this may be the case simply due to incomplete taxon sampling, as discussed in [19]. Therefore, the warning is issued in this case so as to alert that user that this inferred HGT edge is worth further inspection.

Assessing the support of HGT edges

In [37] we proposed a method for assessing the support of HGT edges. Roughly speaking, the support value of HGT edge $X \rightarrow Y$ in the species tree, where $Y'$is the sibling of Y, is derived from the bootstrap values of the gene tree branches that separate the clade under Y from the clade under Y'. The rationale behind the idea is that if Y' and Y are well separated in the gene trees (i.e., some branches in the path from Y to Y' have high bootstrap values), HGT is necessary to move Y away from Y'). For example, the support of HGT edge $X \rightarrow Y$ in Figure 8(a) is calculated based on the bootstrap values of the branches separating B from A in the gene tree, and it is 80 (which is the maximum bootstrap value of all edges on the path separating A and B in the gene tree). More technical details can be found in [37].

Than *et al*. [37] have studied the reliability of this method for assessing the support of HGT edges on various data sets from [38]. In this paper, we illustrate the output of RIATA-HGT on a pair of species/gene trees from [38], as shown in Figure 9. The output of RIATA-HGT on this pair of trees is shown in Figure 10. RIATA-HGT computed four solutions, each of which has nine HGT edges. To allow for a compact representation of the solutions, they are divided into two components (which are computed automatically by RIATA-HGT), and each solution is formed by taking one subsolution from each component. HGT edges for the solutions are divided into two components, which means that a complete solution is formed by taking one solution from each component. Each component is labeled by the name of the internal node that is the root of the clade corresponding to that component. In the case of the solutions presented in Figure 10, each solution contains nine HGT edges, eight of which form a single subso-

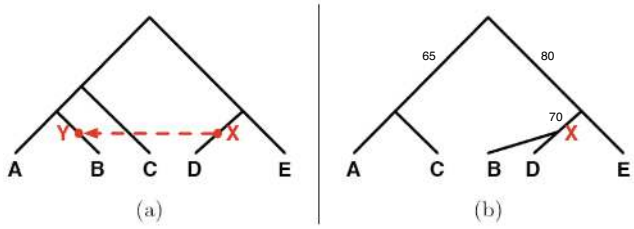

Figure 8
An illustration of computing the support value of an HGT edge. An illustration of computing the support value of an HGT edge. In this case, the support of HGT edge $X \rightarrow Y$ added on the species tree (a), is calculated based on the bootstrap of the branches that separate Y (or B) from A in the gene tree (b). The value for the event $X \rightarrow Y$ is 80.

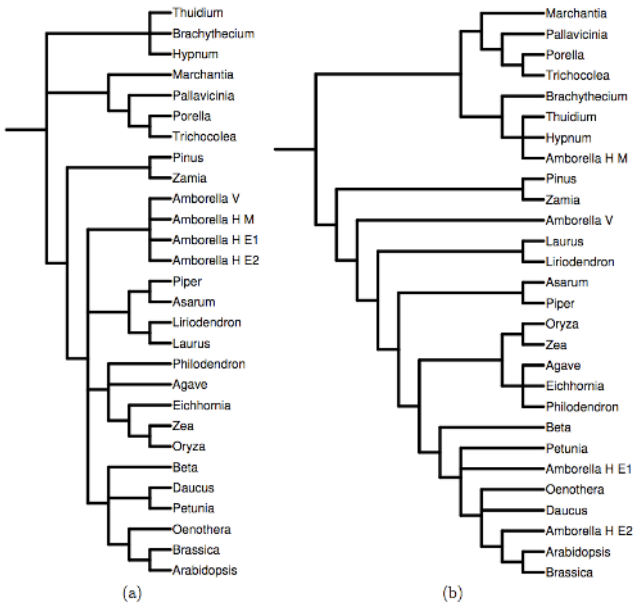

Figure 9

The cox2 trees. The species tree (a): (((Pallavicinia, (Porella, Trichocolea)I15)I16, Marchantia)I17, (Thuidium, Brachythecium, Hypnum)I14, (((Amborella_V, Amborella_H_M, Amborella_H_E1, Amborella_H_E2)I9, ((Eichhornia, (Zea, Oryza)I6)I7, Philodendron, Agave)I8, ((Daucus, Petunia)I4, Beta, (Oenothera, (Brassica, Arabidopsis)I2)I3)I5, ((Piper, Asarum)I10, (Liriodendron, Laurus)I11)I1)I12, (Pinus, Zamia)I13)I0)I18; and gene tree (b): (((((((Petunia, Amborella_H_E1, (((Arabidopsis, Brassica)I2:99.0, Amborella_H_E2), Oenothera, Daucus)), Beta):73.0, ((Agave, Eichhornia, Philodendron), (Oryza, Zea)I6:100.0)), (Asarum, Piper)I10), (Laurus, Liriodendron)I11), Amborella_V), (Pinus, Zamia)I13:72.0), (((Thuidium, Hypnum, Amborella_H_M):91.0, Brachythecium):98.0, (Marchantia, ((Porella, Trichocolea):97.0, Pallavicinia))I17)); for gene *cox2*. Bootstrap values for the branches in the gene tree that are greater than 50.0% are included in the tree Newick representation. The species tree branches do not have bootstrap values.

lution in Component I18 and the ninth is the only edge in the only subsolution in Component I8. The value in parentheses next to each HGT edge is its support value computed from the bootstrap values of the gene tree branches (Figure 9(b)). Bergthorsson *et. al*. [38] reported three HGTs involving *Amborella*: one HGT with donor being a species in the *Moss* group (species *Brachythecium*, *Hypnum*, and *Thuidium*, under the internal node I14 in the species tree) and the other two with donors being species in the *Eudicot* group (species *Arabidopsis*, *Beta*, *Brassica*, *Daucus*, *Petunia*, and *Oenothera*, under the internal node I5 in the species tree). The HGT from *Moss* has high SH support value [39]. RIATA-HGT finds this event, I14 → *Amborella_H_M*, with bootstrap value 98.0%. The other two HGT events from *Eudicot* do not have significant SH support values. RIATA-HGT also finds these events, I5 → *Amborella_H_E1* and I2 → *Amborella_H_E2*. However, their support values are 73.0%, much smaller than that of the event from *Moss*. In addition to these three HGT edges, RIATA-HGT identified six more edges, four out of which had support values smaller than 50.0% (RIATA-HGT does not display support values that are smaller than 50.0%). The HGT edge from Component I8, which is shared among all four solutions, has support value of 100.0%. This edge was not reported in [38]. A similar situation arises with the HGT edge I5 → I8, which is part of the three solutions that contain subsolutions 1, 3, and 4 from Component I18: the HGT edge has support of 100.0%, and was not reported in [38], which may be a reflection that the authors focused only on HGT events involving *Amborella*. The ninth HGT edge in Subsolution2 of Component I18 has support value smaller than 50.0%.

Other utilities

As evident from the description of the methods above, there are fundamental correlations between phylogenetic trees and networks. Hence, many of the evolutionary network utilities use functionalities from the phylogenetic trees domain, which we have implemented and provide as standalone tools in PhyloNet:

• A tool for computing the maximum agreement subtree (MAST) of a pair of trees using the algorithm of Steel and Warnow [35]. We also extended the algorithm so that it computes *all* MASTs of a pair of tree, and this feature is implemented as well.

• A tool for computing the Robinson-Foulds distance measure between two phylogenetic trees [27].

• A tool for computing the *last common ancestor* (lca) of a set of nodes in a phylogenetic tree.

Additionally, PhyloNet provides an implementation of the parsimony-based method RECOMP of Ruths and Nakhleh [40,41] for detecting interspecific recombination in a sequence alignment.

Implementation

A major goal for the PhyloNet package was to make its functionality platform-independent and accessible both to end users for data analysis and to researchers designing new computational methods and techniques. In order to encompass as many platforms as possible, PhyloNet was implemented in Java. As a result, any system with the Java 2 Platform (Version 5.0 or higher) installed can run PhyloNet.

```
There are 2 component(s), which account(s) for 4 solution(s), each of size 9
-----------------------------------------------------------------------------------------------------------
Component I18:
Subsolution1:
        I14 -> Amborella_H_M (98.0)
        I5 -> I11
        I14 -> I17
        I5 -> Amborella_H_E1 (73.0)
        I3 -> Daucus
        I2 -> Amborella_H_E2 (73.0)
        I5 -> I8 (100.0)
        I5 -> I10
        [refine nodes: I14, I5]
Subsolution2:
        I5 -> Amborella_H_E1 (73.0)
        I14 -> Amborella_H_M (98.0)
        I12 -> I10  [time violation?]
        I14 -> I17
        I3 -> Daucus
        I12 -> Amborella_V  [time violation?]
        I2 -> Amborella_H_E2 (73.0)
        I12 -> I11  [time violation?]
        [refine nodes: I5, I14]
Subsolution3:
        I12 -> I11  [time violation?]
        I2 -> Amborella_H_E2 (73.0)
        I5 -> Amborella_H_E1 (73.0)
        I14 -> I17
        I12 -> Amborella_V  [time violation?]
        I3 -> Daucus
        I5 -> I8 (100.0)
        I14 -> Amborella_H_M (98.0)
        [refine nodes: I5, I14]
Subsolution4:
        I5 -> Amborella_H_E1 (73.0)
        I2 -> Amborella_H_E2 (73.0)
        I14 -> Amborella_H_M (98.0)
        I3 -> Daucus
        I12 -> Amborella_V  [time violation?]
        I5 -> I10
        I5 -> I8 (100.0)
        I14 -> I17
        [refine nodes: I5, I14]
-----------------------------------------------------------------------------------------------------------
        Component I8:
        Subsolution1:
                I8 -> I6 (100.0) [time violation?]
***********************************************************************************************************
```

Figure 10

An example of RIATA-HGT output. The output of RIATA-HGT on the species tree and *cox2* gene tree in Figure 9. RIATA-HGT finds 4 solutions, summarized in terms of two components, so that each solution is the union of exactly one subsolution from each component.

PhyloNet can be used in two ways, depending on how the functionality needs to be accessed. A command-line interface exposes all of PhyloNet's tools on a Unix or DOS command-line. Each command accepts input from and writes output to text files. This allows PhyloNet's functionality to be used for manual data analysis or integrated into scripts for performing larger-scale processing. Additionally, a rich and thoroughly documented object model allows the incorporation of any of PhyloNet's functionality into existing Java programs. Also bundled are various programmatic utilities that represent trees, networks, and that read and write these various data structures to and from files.

The command line interface

PhyloNet has a consistent and easy-to-use command line interface. A detailed discussion of this interface and all available options is available in the documentation that accompanies a download of the tool. Here we provide a brief overview of the design of the command-line tool and the tools that can be accessed. Table 3 lists all the commands that are currently available from the command-line. Each of these commands accepts a set of parameters as command-line arguments. All trees, networks, sequences, and other major data structures are read in either from standard in or from text files. Similarly all results can be written either to standard out or to a desired text file. All trees are read and written in Newick format. Networks are read and written in eNewick format. These design features allow the easy use of PhyloNet for manual data analysis or as a tool used within a larger scripted automated analysis.

With the exception of the RECOMP tool, all the functionality of PhyloNet is independent of other third party tools. Because RECOMP must compute many trees using Maximum Parsimony trees, this tool requires that PAUP* [42] be installed on the local system. To run a tool in PhyloNet, invoke the executable *.jar* file downloaded from the PhyloNet project homepage:

java -jar phylonet.jar charnet -i net.in -m tree

Here phylonet.jar is the executable jar downloaded from the project homepage (the fle is assumed to be in the current directory where the user invokes this command), charnet is the name of the tool that decomposes the network contained in file net.in into a set of trees. The reference manual included with the executable jar provides very detailed instructions regarding how to run each tool in the PhyloNet package.

Programmatic interface

Many phylogenetic methods comprise critical, but intermediate, steps in much larger methods. As a result, there is also a need for the functionality in PhyloNet to be available for incorporation into larger programs. As a result, all of PhyloNet's functionality is exposed through an extensive set of Java classes. Each tool is contained within its own Java class and exposes a carefully constructed set of public methods that will be preserved and maintained even as PhyloNet grows. This modular design allows for the easy addition functionality in the future without effecting existing programs that use PhyloNet as a programmatic library. In addition to exposing a consistent API, PhyloNet also provides implementations of the most common phylogenetic data structures: trees and networks. Utility classes are also included that read and write these data structures to and from files. These classes can accelerate not only incorporation of PhyloNet's functionality, but also the development of new phylogenetic functionality within other applications. As PhyloNet grows, programmatic interfaces will be added to provide access to new functionality and tools. Detailed documentation of these libraries is available in JavaDoc form on the PhyloNet website.

Conclusion

Analyzing and understanding reticulate evolutionary relationships have been hindered by the lack of software tools for conducting these tasks. The proposed software package, PhyloNet, offers an array of utilities to allow for efficient and accurate analysis of such evolutionary relationships. These utilities allow for representing networks in a compact way, characterizing networks in terms of basic building blocks and comparing them based on these characterizations, comparing networks in terms of their fitness to the evolution of a given data set of sequences under the maximum parsimony model, and reconstructing networks from species/gene trees.

The software package will help significantly in analyzing large data sets, as well as in studying the performance of evolutionary network reconstruction methods. Further, the software package offers the novel eNewick format for compact representation of evolutionary networks, a feature that allows for efficient interoperability of evolutionary network software tools.

Table 3: List of tools and their description.

Tool name	Purpose
charnet	Computing clusters, trees and tripartitions in a network
cmpnets	Computing the distance between two networks
lca	Finding the last common ancestor of a set of nodes
mast	Computing the maximum agreement subtree
netpars	Scoring the parsimony of sequences on a pair of networks
riatahgt	Reconstructing HGT events from a pair of species/gene trees
rf	Computing the Robinson-Foulds tree measure

A table of the tools currently implemented in PhyloNet. With the exception of the three phylogenetic trees tools lca, mast, and rf, all the other tools are for analyzing reticulate evolutionary relationships.

Availability and requirements

1. **Project name:** PhyloNet | Phylogenetic Networks Toolkit.

2. **Project home page:** http://bioinfo.cs.rice.edu/phylonet/index.html.

3. **Operating system:** Platform independent.

4. **Programming language:** Java.

5. **Other requirements:** Java 1.5, PAUP* (for some applications).

6. **License:** GNU GPL.

7. **Any restrictions to use by non-academics:** The GNU GPL license applies.

Authors' contributions

All authors contributed equally to the work described in this manuscript.

Acknowledgements

The authors would like to acknowledge the very helpful comments from three anonymous reviewers which helped improve the manuscript, as well as the software tool, significantly. This work is supported in part by the Department of Energy grant DE-FG02-06ER25734, the National Science Foundation grant CCF-0622037, the George R. Brown School of Engineering Roy E. Campbell Faculty Development Award, and the Department of Computer Science at Rice University.

References

1. Ochman H, Lawrence J, Groisman E: **Lateral gene transfer and the nature of bacterial innovation.** *Nature* 2000, **405(6784):**299-304.
2. Bergthorsson U, Adams K, Thomason B, Palmer J: **Widespread horizontal transfer of mitochondrial genes in flowering plants.** *Nature* 2003, **424:**197-201.
3. Bergthorsson U, Richardson A, Young G, Goertzen L, Palmer J: **Massive horizontal transfer of mitochondrial genes from diverse land plant donors to basal angiosperm Amborella.** *Proc Natl Acad Sci U S A* 2004, **101(51):**17747-17752.
4. Mower J, Stefanovic S, Young G, Palmer J: **Gene transfer from parasitic to host plants.** *Nature* 2004, **432:**165-166.
5. Posada D, Crandall K, Holmes E: **Recombination in Evolutionary Genomics.** *Annu Rev Genet* 2002, **36:**75-97.
6. Posada D, Crandall K: **The effect of recombination on the accuracy of phylogeny estimation.** *J Mol Evol* 2002, **54(3):**396-402.
7. Ellstrand N, Whitkus R, Rieseberg L: **Distribution of spontaneous plant hybrids.** *Proc Natl Acad Sci U S A* 1996, **93(10):**5090-5093.
8. Buerkle C, Morris R, Asmussen M, Rieseberg L: **The likelihood of homoploid hybrid speciation.** *Heredity* 2000, **84(4):**441-451.
9. Rieseberg L, Baird S, Gardner K: **Hybridization, introgression, and linkage evolution.** *Plant Molecular Biology* 2000, **42:**205-224.
10. Linder C, Rieseberg L: **Reconstructing patterns of reticulate evolution in plants.** *Am J Bot* 2004, **91:**1700-1708.
11. Huson DH, Bryant D: **Application of phylogenetic networks in evolutionary studies.** *Mol Biol Evol* 2006, **23(2):**254-267.
12. Beiko R, Hamilton N: **Phylogenetic identification of lateral genetic transfer events.** *BMC Evolutionary Biology* 2006, **6:**.
13. MacLeod D, Charlebois R, Doolittle F, Bapteste E: **Deduction of probable events of lateral gene transfer through comparison of phylogenetic trees by recursive consolidation and rearrangement.** *BMC Evol Biol* 2005, **5(1):**27.
14. Hallett M, Lagergren J: **Efficient algorithms for lateral gene transfer problems.** In *Proc 5th Ann Int'l Conf Comput Mol Biol (RECOMB01)* New York: ACM Press; 2001:149-156.
15. Makarenkov V: **T-REX: Reconstructing and visualizing phylogenetic trees and reticulation networks.** *Bioinformatics* 2001, **17(7):**664-668.
16. Nakhleh L, Ruths D, Wang L: **RIATA-HGT: A Fast and accurate heuristic for reconstrucing horizontal gene transfer.** *Proceedings of the Eleventh International Computing and Combinatorics Conference (COCOON 05)* 2005:84-93. [LNCS #3595]
17. Than C, Nakhleh L: **SPR-based tree reconciliation: Non-binary trees and multiple solutions.** *Proceedings of the Sixth Asia Pacific Bioinformatics Conference (APBC)* 2008:251-260.
18. Than C, Ruths D, Innan H, Nakhleh L: **Confounding factors in HGT detection: Statistical error, coalescent effects, and multiple solutions.** *Journal of Computational Biology* 2007, **14:**517-535.
19. Moret B, Nakhleh L, Warnow T, Linder C, Tholse A, Padolina A, Sun J, Timme R: **Phylogenetic networks: modeling, reconstructibility, and accuracy.** *IEEE/ACM Trans Comput Biol Bioinform* 2004, **1(1):**13-23.
20. Baroni M, Semple C, Steel M: **A framework for representing reticulate evolution.** *Annals of Combinatorics* 2004, **8(4):**391-408.
21. Felsenstein J: **The Newick Tree Format.** 1986 [http://evolution.genetics.washington.edu/phylip/newicktree.html].
22. Morin MM, Moret BME: **NETGEN: generating phylogenetic networks with diploid hybrids.** *Bioinformatics* 2006, **22(15):**1921-1923.
23. Huson D: **SplitsTree: a program for analyzing and visualizing evolutionary data.** *Bioinformatics* 1998, **14:**68-73.
24. Bryant D, Moulton V: **NeighborNet: An agglomerative method for the construction of planar phylogenetic networks.** *Proc 2nd Int'l Workshop Algorithms in Bioinformatics (WABI02), Volume 2452 of Lecture Notes in Computer Science, Springer-Verlag* 2002:375-391.
25. Felsenstein J: *Inferring Phylogenies* Sunderland, MA: Sinauer Associates, Inc; 2003.
26. Semple C, Steel M: *Phylogenetics* Oxford Lecture Series in Mathematics and its Applications 24, Oxford University Press; 2003.
27. Robinson D, Foulds L: **Comparison of phylogenetic trees.** *Math Biosciences* 1981, **53:**131-147.
28. Nakhleh L, Sun J, Warnow T, Linder R, Moret B, Tholse A: **Towards the development of computational tools for evaluating phylogenetic network reconstruction methods.** *Proc 8th Pacific Symp on Biocomputing (PSB03), World Scientific Pub* 2003:315-326.
29. Nakhleh L, Warnow T, Linder C: **Reconstructing reticulate evolution in species-theory and practice.** *Proc 8th Ann Int'l Conf Comput Mol Biol (RECOMB04)* 2004:337-346.
30. Penny D, Hendy MD: **The use of tree comparison metrics.** *Systematic Zoology* 1985, **34:**75-82.
31. Kanj I, Nakhleh L, Than C, Xia G: **Seeing the trees and their branches in the network is hard.** *Theoretical Computer Science* 2008, **401:**153-164.
32. Nakhleh L, Jin G, Zhao F, Mellor-Crummey J: **Reconstructing phylogenetic networks using maximum parsimony.** *Proc IEEE Comput Syst Bioinform Conf* 2005:93-102.
33. Jin G, Nakhleh L, Snir S, Tuller T: **Inferring Phylogenetic Networks by the Maximum Parsimony Criterion: A Case Study.** *Mol Biol Evol* 2007, **24:**324-337.
34. Bordewich M, Semple C: **On the computational complexity of the rooted subtree prune and regraft distance.** *Annals of Combinatorics* 2005, **8(4):**409-423.
35. Steel M, Warnow T: **Kaikoura tree theorems: computing the maximum agreement subtree.** *Information Processing Letters* 1993, **48:**77-82.
36. Ruths D, Nakhleh L: **Techniques for Assessing Phylogenetic Branch Support: A Performance Study.** *Proceedings of the 4th Asia Pacific Bioinformatics Conference* 2006:187-196.
37. Than C, Jin G, Nakhleh L: **Integrating sequence and topology for efficient and accurate detection of horizontal gene transfer.** *Proceedings of the Sixth RECOM Comparative Genomics Satellite Workshop* 2008.
38. Bergthorsson U, Richardson AO, Young GJ, Goertzen LR, Palmer JD: **Massive horizontal transfer of mitochondrial genes from diverse land plant donors to the basal angiosperm Amborella.** *Proc Natl Acad Sci USA* 2004, **101(51):**17747-17752.

39. Shimodaira H, Hasegawa M: **Multiple comparisons of log-likelihoods with applications to phylogenetic inference.** *Molecular Biology and Evolution* 1999, **16:**1114-1116.
40. Ruths D, Nakhleh L: **Recombination and phylogeny: effects and detection.** *Int J Bioinform Res Appl* 2005, **1(2):**202-212.
41. Ruths D, Nakhleh L: **RECOMP: A Parsimony-based Method for Detecting Recombination.** *Proceedings of the 4th Asia Pacific Bioinformatics Conference* 2006:59-68.
42. Swofford DL: **PAUP*: Phylogenetic Analysis Using Parsimony (and Other Methods).** In *Version 4.0* Sinauer Associates, Underland, Massachusetts; 1996.