

Regular networks can be uniquely constructed from their trees

Stephen J. Willson
 Department of Mathematics
 Iowa State University
 Ames, IA 50011 USA
 swillson@iastate.edu

Abstract—A rooted acyclic digraph N with labelled leaves displays a tree T when there exists a way to select a unique parent of each hybrid vertex resulting in the tree T . Let $Tr(N)$ denote the set of all trees displayed by the network N . In general, there may be many other networks M such that $Tr(M) = Tr(N)$. A network is regular if it is isomorphic with its cover digraph. If N is regular and \mathcal{D} is a collection of trees displayed by N , this paper studies some procedures to try to reconstruct N given \mathcal{D} . If the input is $\mathcal{D} = Tr(N)$, one procedure is described which will reconstruct N . Hence if N and M are regular networks and $Tr(N) = Tr(M)$, it follows that $N = M$, proving that a regular network is uniquely determined by its displayed trees. If \mathcal{D} is a (usually very much smaller) collection of displayed trees that satisfies certain hypotheses, modifications of the procedure will still reconstruct N given \mathcal{D} .

Index Terms—Phylogeny, phylogenetic, network, tree, regular, hybrid.



1 INTRODUCTION

It has become common, for a given collection X of taxa and given a particular gene g , to use DNA to determine a phylogenetic tree T^g . The extant taxa correspond to leaves of the trees, while internal vertices correspond to ancestral species. Each arc represents a lineage (the course of a species through time) during which the population is subject to genetic change until the lineage is next involved in a speciation event. Typical methods for determining the trees include maximum likelihood, maximum parsimony, and neighbor-joining, but many other methods are also utilized.

Frequently the use of a different gene h for the same collection X of taxa results in a tree T^h that differs from T^g . Indeed, many different trees arise for different genes g but the same X . For example [20] utilized 106 orthologs common to seven species of yeast and an outgroup. The collection of 106 maximum-parsimony trees and 106 maximum-likelihood trees included more than 20 different robustly supported topologies. While [20] concatenated the data to try to achieve resolution, [12] employed consensus networks to display the incompatibilities that existed among the trees.

One hypothesis to explain the deviations of such gene trees from a single “species tree” is to assume “lineage sorting”. In this model a single species tree is seen as a kind of pipeline containing populations with significant genetic diversity; the genes actually fixate at locations that need not coincide with the speciation events in the species tree. Hence the genes do not necessarily follow the species tree. Coalescence methods are utilized in [21], [8], [22]. For example, [8] shows that the most likely gene

tree need not coincide with the species tree. Much of the resulting diversity in the gene trees, however, makes use of short branch-lengths separating some speciation events in the species tree.

Other approaches to the problem involve methods, given a number of gene trees, for finding a species tree that best explains the gene trees. For example, Hallett and Lagergren [10] and Page and Charleston [18], [19] seek the species tree with the minimum number of gene duplications and/or losses needed to reconcile the gene trees with the species tree. Alternatively [11] identifies the role of an activity parameter measuring the number of genes that are simultaneously active in the genome for use in the reconciliation. Arvestad *et al.* [1] utilize probabilistic methods and MCMC algorithms to perform the reconciliation.

Another hypothesis to explain the deviations of such gene trees from a single species tree is to assume that evolution actually occurs on networks that are not necessarily trees. Besides mutation events, these networks could include such reticulation events as hybridization or lateral gene transfer. General frameworks are discussed in [2], [3], [16], and [17].

Even if the underlying species relationships are given by a network, the evolution of an individual gene might best be described by a tree. The idea is that, at a hybridization event, some genes would be inherited from one parent species, and other genes from another parent species. Suppose, for example, the underlying species network is M in Figure 1. Species 2 is hybrid with parental species B and C . If a particular gene in 2 is inherited from B , then the correct description of the inheritance of that gene would be tree b in Figure 2. If

instead a gene in 2 is inherited from C , then the correct description for that gene would be tree c in Figure 2. Thus we would expect to see both trees b and c among the various gene trees. Trees b and c are said to be *displayed* by the network. On the other hand, tree d in Figure 2 is not displayed by M , so we would not expect a gene to evolve according to d under these assumptions.

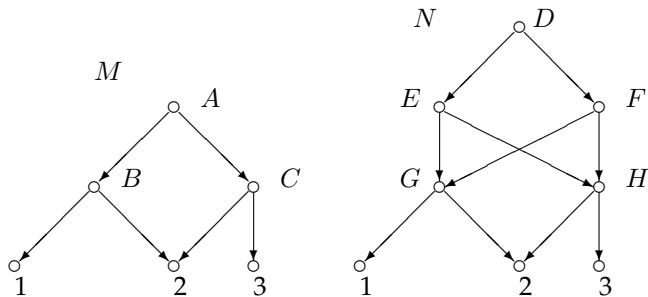


Fig. 1. Two phylogenetic networks with base-set $X = \{1, 2, 3\}$.

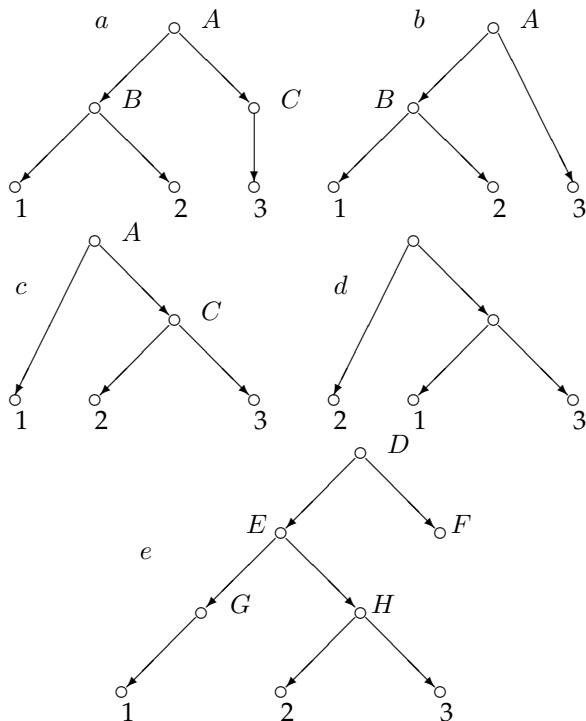


Fig. 2. Some trees related to Figure 1. Both M and N display trees b and c but not d .

The assumption that the underlying description of evolutionary history is a network rather than a tree raises the fundamental problem of reconstructing a network from data. Suppose that a collection of gene trees for the same set X of taxa is known. Can the underlying network be uniquely reconstructed?

If M is a network, let $Tr(M)$ denote the set of rooted

trees displayed by M . Figure 1 shows two distinct networks M and N such that $Tr(M) = Tr(N) = \{b, c\}$. This example represents a common situation. In general there may be many networks that display exactly the same trees.

One approach to obtain a uniquely specified network has been to seek a network that displays a collection of trees and which has the fewest hybridization events. This problem was proved to be NP-hard [5]. Various special cases with additional hypotheses on the networks have also been studied, such as [24], [9], [13], [14].

A different approach has been to make assumptions on the properties of an allowable phylogenetic network. It would be desirable to have a class of phylogenetic networks which are biologically plausible and such that there is often a uniquely determined network of this type with certain observable properties. It is commonly assumed that the networks are rooted acyclic digraphs [23], [16], [17], [15]. Restrictions that appear tractable and yield interesting results include *time consistency* [16], [7], roughly that the parents of a hybrid be contemporaneous. Others include restrictions on the children of vertices, for example *tree-child networks* [6] or *tree-sibling networks* [7]. Certain unique reconstructions for *normal networks* are given in [25].

Baroni *et al.* [3], [4] defined the notion of a *regular network*. The precise definition is given in Section 2. The basic idea is as follows: The *cluster* $cl(v)$ of a vertex v is the set of leaves which are descendants of v . In a regular network, no two distinct vertices have the same cluster. Moreover, $cl(u) \subset cl(v)$ iff there is a directed path from v to u . In Figure 1, M is regular but N is not regular since in N , $cl(D) = cl(E) = cl(F) = \{1, 2, 3\}$.

The main result of this paper is Theorem 3.1. This theorem gives a polynomial-time method which, given $Tr(N)$ for a regular network N uniquely reconstructs the network N . Corollary 3.2 asserts the consequence that if M and N are regular networks with the same set of leaves and $Tr(M) = Tr(N)$, then $M = N$. Thus the entire collection of trees displayed by a regular network uniquely determines the network.

Figure 1 shows two distinct networks M and N such that $Tr(M) = Tr(N)$, where N is not regular. It follows that, that without an assumption such as regularity, a network is not uniquely determined by the set of its displayed trees.

The proof of Theorem 3.1 is constructive. A procedure Maximal Proper Child is applied to the input $\mathcal{D} = Tr(N)$. When N is regular, the procedure outputs the network N up to isomorphism. (In fact it outputs the cover digraph of N , see [4].) An example is worked in Section 5, illustrating the procedure.

The procedure for Theorem 3.1 has input consisting of a set \mathcal{D} of trees and yields a network for any input \mathcal{D} . The complexity is polynomial-time in the input. More specifically, suppose N has v vertices and there are n leaves; suppose \mathcal{D} consists of d trees. Then the procedure for Theorem 3.1 in Cor 3.4 will be shown to take time

$O(vn^2d^2)$.

It is not likely in a real biological problem that all the trees displayed by a network are known. Moreover, the number of such trees could grow exponentially large. Indeed, if N has k hybrid vertices, each with exactly two parents, then $Tr(N)$ may have 2^k members, making the input $\mathcal{D} = Tr(N)$ very large, and making the algorithm take time $O(vn^22^{2k})$.

Section 4 shows that there is a much smaller collection \mathcal{D} of trees displayed by any regular network N such that a modified procedure will reconstruct N from \mathcal{D} . In fact, suppose a is the number of arcs in N and h is the maximal number of arcs on a directed path in N . There exists \mathcal{D} with a trees (or fewer) such that N can be reconstructed from \mathcal{D} in time $O(vhn^2a^2)$. If there are k hybrid vertices each with 2 parents, this time will typically be far smaller than $O(vn^22^{2k})$. Thus N is uniquely determined by a much smaller collection of trees \mathcal{D} than $Tr(N)$. We do not present, however, a polynomial-time construction of such \mathcal{D} .

In Section 4 we also see that if N is a normal network, then a simpler and faster algorithm will reconstruct N from a suitable collection \mathcal{D} containing a trees.

A result related to Corollary 3.2 is found in [16]. Given a *reconstructible* network N , [16] gives a procedure to find a *reduced version* $R(N)$ of N . Theorem 2 of [16], in our notation, asserts that if M and N are reconstructible networks and $Tr(M) = Tr(N)$, then $R(M) = R(N)$. Thus if N is reconstructible, then $R(N)$ is uniquely determined by $Tr(N)$. For such networks the authors do not, however, give a procedure to reconstruct $R(N)$ from $Tr(N)$, analogous to our Theorem 3.1.

This situation contrasts with that in which the *generalized clusters* or *tree clusters* of a network are given instead of all the displayed trees. For a network N , a generalized or tree cluster is any cluster of any tree T displayed by N . The set of all tree clusters of N is denoted $TrCl(N)$. The paper [15] shows that the problem, given a network N and a cluster U , of deciding whether $U \in TrCl(N)$ is NP-complete. The paper [25] presents examples of distinct regular networks (indeed normal networks) M and N which have the same leaf sets and which have precisely the same tree clusters; thus $TrCl(M) = TrCl(N)$ but M and N are not isomorphic. The author therefore finds it somewhat surprising that, as shown in the current paper, the trees themselves do determine the network uniquely for a broad class of networks.

Section 6 concludes the paper with a suggestion about how these methods might be applied to real data.

2 BASICS

A *directed graph* or *digraph* $N = (V, A)$ consists of a finite set $V = V(N)$ of *vertices* and a finite set $A = A(N)$ of *arcs*, each consisting of an ordered pair (u, v) where $u \in V$, $v \in V$, $u \neq v$, interpreted as an arrow from u (the *parent*) to v (the *child*). There are no multiple arcs and no loops. A *directed path* is a sequence u_0, u_1, \dots, u_k of vertices

such that for $i = 1, \dots, k$, $(u_{i-1}, u_i) \in A$. The *length* of the path is k and the path is *trivial* if $k = 0$. The graph is *acyclic* if there is no nontrivial directed path starting and ending at the same point. Write $u \leq_N v$ or more informally $u \leq v$ in N if there is a directed path starting at u and ending at v . Write $u < v$ if $u \leq v$ and $u \neq v$. If the graph is acyclic, it is easy to see that \leq is a partial order on V .

A vertex r is a *root* of the directed acyclic graph (V, A) if, for all $v \in V$, $r \leq v$. The network is *rooted* if it has a root. Clearly there can be at most one root.

The *indegree* of vertex u is the number of $v \in V$ such that $(v, u) \in A$. The *outdegree* of u is the number of $v \in V$ such that $(u, v) \in A$. If N is rooted at r then r is the only vertex of indegree 0. A *leaf* is a vertex of outdegree 0. A *normal* (or *tree*) vertex is a vertex of indegree at most 1. A *hybrid* vertex (or *recombination vertex* or *reticulation node*) is a vertex of indegree at least 2.

Let X be a set. The cardinality of X will be denoted $|X|$. In biological terms we consider the members of X to be a specific collection of biological species. We call X the *base-set* of the directed graph $N = (V, A)$ if there is a given one-to-one relationship between X and the subset $L(N) \subseteq V$ consisting of the leaves of N . Thus we identify the leaves of N with the members of X . The interpretation of X is that its members correspond to taxa on which direct measurements may be made, while N describes a proposed evolutionary history giving rise to these taxa. The leaves correspond to extant taxa so direct measurements are possible. Typically one taxon is included which is an outgroup—an extant species clearly on a separate evolutionary track from all other taxa. Hence the root is located as the attachment vertex of the outgroup taxon.

In this paper a (*phylogenetic*) network $N = (V, A, r, X)$ is an acyclic digraph (V, A) with root r and base-set X . Two networks $N = (V, A, r, X)$ and $M = (V', A', r', X')$ are *equal*, $N = M$, iff there is a bijection $\phi : V \rightarrow V'$ such that for all $x \in X$, $\phi(x) = x$, and $(u, v) \in A$ iff $(\phi(u), \phi(v)) \in A'$.

Let $N = (V, A, r, X)$ be a phylogenetic network. Let $\mathcal{P}(X)$ denote the set of all subsets of X . For $v \in V$, define the (full) *cluster* of v in N by $cl(v, N) = \{x \in X : v \leq x\}$. It is clear that for each $v \in V$, $cl(v, N) \in \mathcal{P}(X)$. Define for each phylogenetic network N with base-set X , $cl_N : V \rightarrow \mathcal{P}(X)$ by $cl_N(v) = cl(v, N)$.

The following properties of the clusters are basic:

- (1) For $v \in V$, $cl_N(v)$ is nonempty.
- (2) If $u \leq_N v$, then $cl_N(v) \subseteq cl_N(u)$.
- (3) $cl_N(r) = X$.
- (4) If $x \in X$, then $cl_N(x) = \{x\}$.

Note that (1) follows since a maximal path must end at a leaf and every leaf lies in X . Moreover (2) follows since \leq_N is a partial order. In particular, if (u, v) is an arc of N , then $cl_N(v) \subseteq cl_N(u)$. Also, (3) follows since for each $x \in X$, we have $r \leq x$, and (4) follows since each $x \in X$ satisfies that x is a leaf.

The clusters X and $\{x\}$ for $x \in X$ are called the *trivial*

clusters since they occur in each network. Any other clusters will be called *nontrivial*.

Given the network $N = (V, A, r, X)$, we may let $\mathcal{C}(N) = \{cl_N(v) : v \in V\} \subseteq \mathcal{P}(X)$. The *cover digraph* of N is the digraph (W, E) where

- (1) $W = \mathcal{C}(N)$, and
- (2) there is an arc $(B, C) \in E$ for B and C in W iff
 - (2a) $C \subset B$; and
 - (2b) there is no $D \in W$ such that $C \subset D \subset B$.

Note the root is $X = cl_N(r)$ because for all $x \in X$, $r \leq x$. Note since the members of X are the leaves of N , it follows for each $x \in X$, $cl_N(x) = \{x\}$ so the leaves of the cover digraph are the singleton sets $\{x\}$ for $x \in X$. Hence the leaves may be identified with the members of X and the root r with X .

Baroni *et al.* [3], [4] defined a *regular* network to be a network which is isomorphic with its cover digraph. The following equivalent description is similar to that given in Theorem 4.1 of [3]. The phylogenetic network $N = (V, A, r, X)$ is *regular* provided

- (1) $cl_N : V \rightarrow \mathcal{P}(X)$ is one-to-one; and
- (2) there is an arc $(u, v) \in A$ iff $cl_N(v) \subset cl_N(u)$ and there is no $w \in V$ such that $cl_N(v) \subset cl_N(w) \subset cl_N(u)$.

Call an arc (u, v) *redundant* if there is a directed path from u to v other than along arc (u, v) . From (2) it is clear that a regular network contains no redundant arc.

Let $N = (V, A, r, X)$ be a phylogenetic network. A *parent map* for N is a map $p : V - \{r\} \rightarrow V$ such that for each $v \in V$, $v \neq r$, $p(v)$ is a parent of v , i.e., $(p(v), v) \in A$. Since the root r is unique, it is clear that if $v \in V$, $v \neq r$, then v has a parent. Note that if v is normal and $v \neq r$, then v has exactly one parent q , so all parent maps p will satisfy $p(v) = q$. When v is hybrid, however, there are at least two parents of v .

Let $Par(N)$ denote the collection of parent maps for the network N . Let $i(v, N)$ denote the indegree of v in the network N . Then the number of distinct parent maps is clearly $|Par(N)| = \prod [i(v, N) : v \in V, v \neq r]$.

For any parent map p for $N = (V, A, r, X)$ construct a new network $N_p = (V, E, r, X)$ as follows: The vertex set, root, and base-set are the same as for N . The arc set E consists of all arcs of the form $(p(v), v)$ where $v \in V$, $v \neq r$. Thus $E \subseteq A$.

Each vertex v other than r has exactly one parent in N_p ; i.e., $i(v, N_p) = 1$. Hence N_p is a rooted tree. It is quite possible that v has outdegree 1 as well as indegree 1, but such vertices are often suppressed in a rooted tree. We will therefore consider two kinds of simplification to change N_p into a rooted tree in standard form.

Type 1: Suppress a vertex with outdegree 1. More specifically, if u has outdegree 1, say via arc (u, v) , then remove u ; remove also each arc (w, u) and replace it by arc (w, v) . If u was the root of the original tree, then v becomes the root of the revised tree.

Type 2: Suppress a vertex with no directed path to a member of X . More specifically, suppose u is such a vertex. Then, delete u , for each arc (v, u) delete (v, u) , and for each arc (u, v) , delete (u, v) .

The result of performing all possible simplifications of Type 1 or Type 2 on N_p is denoted $T(N_p)$, called the *standard form* of N_p .

Given a network N , $Tr(N)$ will denote the set of all displayed trees: $Tr(N) = \{T(N_p) : p \in Par(N)\}$.

Figure 2 exhibits some rooted trees related to Figure 1. In M let the parent map p satisfy $p(2) = B$ (and trivially $p(1) = B$, $p(3) = C$, $p(B) = A$, $p(C) = A$). Then the tree M_p is given in *a*. Since C has outdegree 1 in Fig 2a, it is suppressed by Type 1, resulting in the standard form *b*, which is $T(M_p)$. Similarly if p' is the parent map with $p'(2) = C$, then Fig 2c shows $T(M_{p'})$. The tree *d* is not displayed by M since there is no parent map yielding *d*.

It is easy to see that N in Fig 1 also displays *b* and *c*. Consider the parent map q for N given by $q(2) = H$, $q(G) = E$, $q(H) = E$. Then $N_q = e$ in Figure 2. We simplify *e* by suppressing *F* by Type 2 and then *D* and *G* by Type 1. Hence $T(N_q) = c$ in Figure 2. For both the networks in figure 1, we have $Tr(M) = Tr(N) = \{b, c\}$ using the notation in Figure 2.

It is quite possible that there are two distinct parent maps p and q such that $T(N_p) = T(N_q)$, even when N is regular. For example, consider Figure 3. The parent maps p and q given by $p(1) = a, p(2) = p(3) = c$ and $q(1) = b, q(2) = q(3) = c$ both yield the same tree $(1, (2, 3))$.

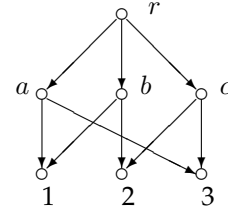


Fig. 3. A regular network with base-set $X = \{1, 2, 3\}$. There are two distinct parent maps yielding the same tree $(1, (2, 3))$ as the standard form.

3 RECONSTRUCTION OF REGULAR NETWORKS FROM THEIR TREES

Suppose \mathcal{D} is a nonempty collection of rooted trees each with the same base-set X . In this section we present a procedure called Maximal Proper Child (MPC) which constructs a phylogenetic network $MPC(\mathcal{D})$ given \mathcal{D} . The algorithm always terminates with a network.

The main theorem 3.1 asserts that if $\mathcal{D} = Tr(N)$ for some regular network N , then the output of the procedure is the cover digraph of N , hence isomorphic to N , so N has been reconstructed.

Theorem 3.1. *Suppose $N = (V, A, r, X)$ is a regular phylogenetic network. Then the output of Maximal Proper Child applied to $\mathcal{D} = Tr(N)$ is the cover digraph of N , hence isomorphic with N .*

An immediate consequence of Theorem 3.1 is Corollary 3.2, which asserts that the set of trees displayed by a regular network uniquely determines the network.

Corollary 3.2. *Suppose M and N are regular phylogenetic networks with base-set X . If $Tr(M) = Tr(N)$, then $M = N$.*

The number of displayed trees may be exponentially large in $|X|$, so the algorithm need not be polynomial-time in $|X|$. We shall see, however, that the procedure is polynomial-time in $|X||V||\mathcal{D}|$.

Figure 1 shows that Cor 3.2 fails without the assumption of regularity.

A crucial concept for the proof is the following: Let $P_n = r, P_{n-1}, \dots, P_1, P_0 = C$ be a directed path in N from the root r to C . A displayed tree T is *complete for the path* provided that there is a directed path $w_n = r, w_{n-1}, \dots, w_1, w_0$ in T such that for all $i = 0, 1, \dots, n$, we have $cl(w_i, T) = cl(P_i, N)$.

Here is an overview of the procedure: We reconstruct the network N recursively by finding the clusters of N . Initially, we have only the root cluster, which is X . At any given stage, given a cluster $C = cl(u, N)$ for some vertex $u \in V$ we are able to identify the clusters of all its children in N . To do so, by construction we will know already the clusters along a directed path $X = P_n, P_{n-1}, \dots, P_0 = C$ from X to C . The “proper trees” for C , denoted $ProperTr(C)$, will consist of all input trees T which are complete for any such directed path ending at C . We list the clusters for the children of C in all the proper trees for C . Among these we consider the set of “maximal proper children,” denoted $MaxProperCh(C)$, consisting of the clusters U for children of C in a proper tree, such that there is no other cluster W which is a child of C in some proper tree and for which $U \subset W \subset C$. We show that these maximal children are necessarily the clusters of children of C in N , and all clusters of the children of C in N arise in this manner. Hence the children of C are precisely the members of $MaxProperCh(C)$. We insert the members of $MaxProperCh(C)$ into the set of vertices of our reconstruction, together with arcs from C to each such vertex; then we continue recursively.

An example of the procedure will be given in Section 4.

The following is a precise more formal description:

Algorithm Maximal Proper Child.

Input: \mathcal{D} is a nonempty collection of rooted trees each with the base-set X .

Output: A regular phylogenetic network M with base-set X .

Procedure. We construct a sequence M_0, M_1, \dots of directed graphs where $M_k = (V_k, A_k)$. Each member of V_k is a nonempty subset of X , and $V_0 \subseteq V_1 \subseteq V_2 \subseteq \dots$.

1. Initially $M_0 = (V_0, A_0)$ with $V_0 = \{X\}$ and $A_0 = \emptyset$. Thus M_0 has a single vertex which is the cluster X . This vertex is not checked off.

Recursively perform the following step 2:

2. Suppose $M_k = (V_k, A_k)$ is known and some vertex $U \in V_k$ is not checked off.

2a. If V_k contains a singleton set $C = \{a\}$ which is not checked off, then $M_{k+1} = M_k$ except that $\{a\}$ has been checked off.

2b. If V_k contains a doubleton set $C = \{a, b\}$ which is not checked off, then $V_{k+1} := V_k \cup \{\{a\}, \{b\}\}$, and $A_{k+1} = A_k \cup \{(\{a, b\}, \{a\}), (\{a, b\}, \{b\})\}$. In M_{k+1} check off all members of V_k that were already checked off and in addition check off $\{a, b\}, \{a\}$, and $\{b\}$ but nothing else. This thus adjoins the two singletons $\{a\}$ and $\{b\}$.

2c. Suppose neither 2a nor 2b applies. Suppose $C \in V_k$ has not been checked off.

2c1. Let $ProperTr(C) = \{T \in \mathcal{D} : C \text{ is a cluster of } T \text{ and there is a directed path } X = P_n, P_{n-1}, \dots, P_1, P_0 \text{ in } T \text{ such that } cl(P_0, T) = C \text{ and for each } i, cl(P_i, T) \text{ is a vertex of } M_k \text{ and each arc } (P_i, P_{i-1}) \text{ is an arc of } M_k\}$. Call it the set of proper trees for C .

2c2. Let $ProperCh(C) = \{D : \text{for some } T \in ProperTr(C), \text{ the vertex } c \text{ of } T \text{ such that } cl(c, T) = C \text{ has a child } d \text{ such that } D = cl(d, T)\}$. Essentially it is the set of children of C in any proper tree for C .

2c3. Let $MaxProperCh(C) = \{D \in ProperCh(C) : \text{there is no } D' \text{ in } ProperCh(C) \text{ such that } D \subset D' \subset C\}$ (strict inclusions). It is the set of maximal children of C from any proper tree for C .

2c4. For each $D \in MaxProperCh(C)$, adjoin to M the vertex D (if it is not already present) and the arc (C, D) . More explicitly define $V_{k+1} = V_k \cup \{D : D \in MaxProperCh(C)\}$. Define $A_{k+1} = A_k \cup \{(D, C) : D \in MaxProperCh(C)\}$. In M_{k+1} check off all vertices checked off in M_k and also check off C but nothing else. Note that it is possible that D is already present in V_k , but that this construction may still introduce a new arc incoming to D .

3. The procedure terminates with M_n such that every member of V_n has been checked off. Return M_n .

It is clear that the procedure always terminates, whether or not $\mathcal{D} = Tr(N)$. This is because X is a finite set, so $\mathcal{P}(X)$ is finite and there can only be finitely many vertices. At the end of 2a, 2b, or 2c an additional vertex is checked off. Hence after finitely many steps all vertices must be checked off.

Moreover, whenever a new vertex D is added in step 2b or 2c, $ProperTr(D)$ is nonempty. This is trivially true if D arose as a singleton set in 2b. If D arose in 2c, then there exists a parent C of D and $T \in ProperTr(C)$. Hence T also lies in $ProperTr(D)$. Thus when 2c is applied to C containing at least three members of X , it identifies a child D of C which is a nonempty proper subset of C . It follows that when the procedure terminates, each singleton set $\{x\}$ is in V_n .

An example is given in Section 5.

We now turn to the proof of Theorem 3.1. The first step is a lemma which shows the existence of displayed trees which are complete for a given directed path in N starting at the root.

Lemma 3.3. *Let C be a vertex of N . Let $P_n = r, P_{n-1}, \dots, P_1, P_0 = C$ be a directed path in N from the root r to C . There exists a tree T displayed by N which is complete for the path.*

Proof: We find a tree T as follows: The parent map p which yields T is selected by

(0) If W is normal, $W \neq r$, then $p(W)$ is the unique parent of W .

(1) If H is hybrid and $C < H$, choose a parent $p(H)$ of H such that $C \leq p(H)$ in N .

(2) Suppose $n \geq 1$. If H is hybrid and $P_1 < H$, but it is false that $C < H$, choose $p(H)$ such that $P_1 \leq p(H)$ in N .

(3) Suppose $n \geq 2$. If H is hybrid and $P_2 < H$ but it is false that $P_1 < H$ (hence also false that $C < H$), then select $p(H)$ such that $P_2 \leq p(H)$ in N .

(k) In general, if $n \geq k$, H is hybrid, and $P_k < H$ but it is false that $P_{k-1} < H$, select $p(H)$ such that $P_k \leq p(H)$ in N .

Since $P_n = r$, it follows that for each hybrid H , $p(H)$ will be defined.

I claim that $cl(C, N_p) = cl(C, N)$. Clearly $cl(C, N_p) \subseteq cl(C, N)$. Conversely, suppose W is a vertex of N and $C \leq W$ in N . I will show that $C \leq W$ in N_p . It suffices to show that whenever $C < W$ in N , then there exists a parent P of W in N_p such that $C \leq P$ in N . The result is immediate if W has a unique parent P in N because since $C < W$ it follows $C \leq P$. If, instead, W is hybrid, then by assumption $p(W)$ is a parent of W in N_p and by (1) $C \leq p(W)$ in N . This proves that $C \leq W$ in N_p if $C \leq W$ in N . Now, if $x \in cl(C, N)$ the choice $W = x$ shows, since $C \leq x$ in N , that $C \leq x$ in N_p , whence $x \in cl(C, N_p)$. Thus $cl(C, N_p) = cl(C, N)$.

Suppose $n \geq 1$. I now claim that $cl(P_1, N_p) = cl(P_1, N)$. It is immediate that $cl(P_1, N_p) \subseteq cl(P_1, N)$. For the converse, suppose $x \in cl(P_1, N)$. Suppose W is a vertex of N and $P_1 \leq W$ in N . I show that $P_1 \leq W$ in N_p . It suffices to show that if $P_1 < W$ in N , then there exists a parent P of W in N_p such that $P_1 \leq P$ in N . If $C < W$ in N , then from above there exists a parent P of W in N_p such that $C \leq P$ in N , whence $P_1 \leq C \leq P$ in N . Hence we may assume that $C \not\leq W$ in N . If W is normal, then its unique parent P must satisfy that $P \leq W$ in N_p (since arcs to normal vertices remain in N_p) whence $P_1 \leq P$ in N . If instead W is hybrid, then since $P_1 < W$ but $C \not\leq W$ it follows from (2) that $p(W)$ satisfies $P_1 \leq p(W)$ in N . This proves that $P_1 \leq W$ in N_p if $P_1 \leq W$ in N . Now, if $x \in cl(P_1, N)$ the choice $W = x$ shows, since $P_1 \leq x$ in N , that $P_1 \leq x$ in N_p , whence $x \in cl(P_1, N_p)$. Thus $cl(P_1, N_p) = cl(P_1, N)$.

The argument can be iterated to show that for $i = 0, \dots, n$, $cl(P_i, N_p) = cl(P_i, N)$.

Let $T = T(N_p)$ be the standard form of N_p obtained by suppressing vertices of outdegree 1 and vertices with no directed paths to any member of X . By regularity of N , the sets $cl(P_i, N)$ are distinct for $i = 0, \dots, n$. Hence the sets $cl(P_i, N_p)$ are distinct for $i = 0, \dots, n$. I claim

that P_n, \dots, P_0 are vertices of T .

Note first that there exists a directed path in N of maximal length (number of arcs) starting at $P_0 = C$. The path must end at some leaf which consists of a member $x \in X$ since X contains all the leaves. From (0) and (1) it follows that there is a path in N_p from C to x as well; otherwise some vertex W on that path would satisfy that $C < W$ in N so some parent P of W satisfies $C \leq P$, but $p(W)$ satisfies that $C \not\leq p(W)$, contradicting (0) or (1). Hence there is a directed path in N_p from C to x , whence also a directed path from each P_i to x . It follows that no P_i is suppressed because there is no path to a member of X .

Moreover, for $i = 1, \dots, n$, P_i is a vertex of T ; otherwise P_i would have outdegree 1 in N_p whence $cl(P_i, N_p) = cl(P_{i-1}, N_p)$, whence $cl(P_i, N) = cl(P_{i-1}, N)$. Moreover, I claim that $C = P_0$ is a vertex of T . The claim is immediate if C is a leaf. If C is not a leaf then C has children D_1, D_2, \dots, D_k in N , with $k \geq 2$. By regularity $cl(D_j, N)$ is a proper subset of $cl(C, N)$. If C were not a vertex of T , it would have outdegree 1 in N_p . Assume its child in N_p is D_1 . Then $cl(C, N) = cl(C, N_p) = cl(D_1, N_p) \subseteq cl(D_1, N) \subset cl(C, N)$, a contradiction.

It follows that in T there is a directed path $P_n = X, P_{n-1}, \dots, P_1, P_0 = C$ such that for $i = 0, \dots, n$, $cl(P_i, T) = cl(P_i, N)$. \square

We now prove Theorem 3.1.

Proof: Let N be a regular network and $\mathcal{D} = Tr(N)$. Let the sequence of networks obtained from MPC be denoted M_0, M_1, \dots, M_n where $M_i = (V_i, A_i)$ has the set V_i of vertices and the set A_i of arcs. Initially $V_0 = \{X\}$ and $A_0 = \emptyset$.

The proof will be by induction. The i -th inductive hypothesis H_i is that

(1) For each vertex U of M_i there exists a vertex U' of N such that $U = cl(U', N)$.

(2) For each arc (U, W) of M_i , (U', W') is an arc of N .

(3) For each vertex U of M_i that has at least one child in M_i , for every child Y of U' in N , there exists a vertex W of M_i such that W is a child of U in M_i and $W' = Y$.

H_0 is trivially true since X is the only vertex of M_0 and X' is the root of N .

Claim 1. Assume H_j and the procedure has not terminated. We show H_{j+1} .

If 2a or 2b applies, then Claim 1 is immediate. Hence we assume that 2c applies and there is a vertex C of M_j containing at least three points which has not been checked off. Compute $ProperTr(C)$ and $MaxProperCh(C)$ as above. By H_j , there exists vertex C' of N such that $C = cl(C', N)$. It suffices to show that (a) for each child Y of C' in N , $D := cl(Y, N)$ lies in $MaxProperCh(C)$; and

(b) each member of $MaxProperCh(C)$ consists of a cluster D for which there exists a child E of C' in N such that $D = cl(E, N)$.

We first prove (a):

Claim 1a. Let Y be a child of C' in N . Then $D = cl(Y, N)$ is a member of $MaxProperCh(C)$.

Since C is a vertex in M_j , there exists by H_j a directed path $r = P_n, P_{n-1}, P_{n-2}, \dots, P_1, P_0 = C'$ in N from r to C' such that for $i = 0, \dots, n$, $cl(P_i, N)$ is a vertex of M and for $i = 1, \dots, n$, $(cl(P_i, N), cl(P_{i-1}, N))$ is an arc of M . (This is because C occurred in M as a child of some vertex, which occurred in M as a child of some other vertex, etc.)

By Lemma 3.3, since Y is a child of C' in N , there exists a tree T in $Tr(N)$ that contains the directed path $r = Q_n, Q_{n-1}, \dots, Q_0, Q_{-1}$ for which $cl(Q_i, T) = cl(P_i, N)$, $cl(Q_0, T) = cl(C', N) = C$ and $cl(Q_{-1}, T) = cl(Y, N) = D$. By H_j , $T \in ProperTr(C)$, so it follows that $D = cl(Y, N) \in ProperCh(C)$.

I claim that $D \in MaxProperCh(C)$. Otherwise, there exists a tree \hat{T} in $ProperTr(C)$ with vertex \hat{C} such that $C = cl(\hat{C}, \hat{T})$, \hat{C} has child \hat{D} in \hat{T} , and $D \subset cl(\hat{D}, \hat{T}) \subset C = cl(\hat{C}, \hat{T})$. Let $r = \hat{P}_m, \hat{P}_{m-1}, \dots, \hat{P}_0 = \hat{C}$ be the directed path from the root r to \hat{C} in \hat{T} . By construction, for $i = 0, \dots, m$, $cl(\hat{P}_i, \hat{T})$ is a member of M and for $i = 1, \dots, m$, each arc $(cl(\hat{P}_i, \hat{T}), cl(\hat{P}_{i-1}, \hat{T}))$ is an arc in M_j . By H_j , for each i , $cl(\hat{P}_i, \hat{T})$ is a cluster of N ; i.e., there exists vertex Q_i in N such that $cl(Q_i, N) = cl(\hat{P}_i, \hat{T})$ and (Q_i, Q_{i-1}) is an arc of N . In particular, by regularity of N , $Q_0 = C'$.

Let \hat{p} be the parent map that yields \hat{T} (i.e., $T(N_{\hat{p}}) = \hat{T}$). Note for $0 \leq i \leq m$ that \hat{P}_i is a vertex of both \hat{T} and $N_{\hat{p}}$. Then $cl(Q_i, N) = cl(\hat{P}_i, \hat{T}) = cl(\hat{P}_i, N_{\hat{p}}) \subseteq cl(\hat{P}_i, N)$. Since N is regular it follows that $\hat{P}_i \leq Q_i$ in N . Since the arcs of $N_{\hat{p}}$ form a subset of the arcs of N , it follows from $\hat{P}_{i+1} \leq \hat{P}_i$ in $N_{\hat{p}}$ that $\hat{P}_{i+1} \leq \hat{P}_i$ in N as well for $i = 0, \dots, m-1$.

Since $cl(\hat{P}_m, N_{\hat{p}}) = X = cl(Q_m, N)$ it is clear that $\hat{P}_m = Q_m$.

Now $cl(Q_{m-1}, N) = cl(\hat{P}_{m-1}, N_{\hat{p}}) \subseteq cl(\hat{P}_{m-1}, N) \subseteq cl(\hat{P}_m, N) = cl(Q_m, N)$ [since $\hat{P}_m \leq \hat{P}_{m-1}$ in N]. By regularity of N it follows that $Q_m \leq \hat{P}_{m-1} \leq Q_{m-1}$ in N . The arc (Q_m, Q_{m-1}) of N is not redundant, so it follows that either $\hat{P}_{m-1} = Q_m$ or $\hat{P}_{m-1} = Q_{m-1}$. But $\hat{P}_{m-1} \neq \hat{P}_m = Q_m$, so we see that $\hat{P}_{m-1} = Q_{m-1}$.

Similarly $cl(Q_{m-2}, N) = cl(\hat{P}_{m-2}, N_{\hat{p}}) \subseteq cl(\hat{P}_{m-2}, N) \subseteq cl(\hat{P}_{m-1}, N) = cl(Q_{m-1}, N)$ [since $\hat{P}_{m-1} \leq \hat{P}_{m-2}$ in N]. By regularity of N it follows that $Q_{m-1} \leq \hat{P}_{m-2} \leq Q_{m-2}$ in N . The arc (Q_{m-1}, Q_{m-2}) of N is not redundant since N is regular, so it follows that either $\hat{P}_{m-2} = Q_{m-1}$ or $\hat{P}_{m-2} = Q_{m-2}$. But $\hat{P}_{m-2} \neq \hat{P}_{m-1} = Q_{m-1}$, so we see that $\hat{P}_{m-2} = Q_{m-2}$.

In like manner we see that $\hat{P}_i = Q_i$ for $i = m-3, m-4, \dots, 0$.

It follows that $\hat{C} = \hat{P}_0 = Q_0 = C'$. Since Y is a child of C' in N we know $cl(Y, N) = D \subset cl(\hat{D}, \hat{T}) = cl(\hat{D}, N_{\hat{p}}) \subseteq cl(\hat{D}, N) \subseteq cl(\hat{P}_0, N) = cl(C', N)$. It follows that $C' \leq \hat{D} \leq Y$ in N . Since the arc (C', Y) is nonredundant, either $\hat{D} = C'$ or $\hat{D} = Y$. But $\hat{D} \neq C'$ since \hat{D} is a child of $\hat{C} = C'$. It follows that $\hat{D} = Y$. Hence $D = cl(Y, N) \subset cl(\hat{D}, \hat{T}) \subseteq cl(\hat{D}, N) = cl(Y, N)$,

which is impossible. This contradiction proves that $D = cl(Y, N) \in MaxProperCh(C)$.

Next we prove (b):

Claim 1b. Each member D of $MaxProperCh(C)$ satisfies that there exists a child E of C' in N such that $D = cl(E, N)$.

Let D be a member of $MaxProperCh(C)$. Thus there exists a tree \hat{T} in $ProperTr(C)$ with vertex \hat{C} such that $C = cl(\hat{C}, \hat{T})$ and \hat{C} has child \hat{D} in \hat{T} such that $cl(\hat{D}, \hat{T}) = D$. Let $r = \hat{P}_m, \hat{P}_{m-1}, \dots, \hat{P}_0 = \hat{C}$ be the directed path from r to \hat{C} in \hat{T} . By construction, for $i = 0, \dots, m$, $cl(\hat{P}_i, \hat{T})$ is a vertex of M_j and for $i = 1, \dots, m$, each arc $(cl(\hat{P}_i, \hat{T}), cl(\hat{P}_{i-1}, \hat{T}))$ is an arc in M_j . By H_j , for i such that $0 \leq i \leq m$, there exists a vertex Q_i of N such that $cl(\hat{P}_i, \hat{T}) = cl(Q_i, N)$, and for $1 \leq i \leq m$, (Q_i, Q_{i-1}) is an arc of N . In particular, by regularity of N , $Q_0 = C'$.

As in the proof of Claim 1a, we see that $\hat{P}_i = Q_i$ for $i = m, m-1, \dots, 0$ and $\hat{C} = \hat{P}_0 = Q_0 = C'$. Since \hat{D} is a child of \hat{C} in $N_{\hat{p}}$, it follows that $\hat{C} \leq \hat{D}$ in $N_{\hat{p}}$, whence $\hat{C} \leq \hat{D}$ in N . Since $\hat{D} \neq \hat{C}$, there exists a child E of $\hat{C} = C'$ in N such that $E \leq \hat{D}$. Hence $D = cl(\hat{D}, \hat{T}) = cl(\hat{D}, N_{\hat{p}}) \subseteq cl(E, N)$. By Claim 1a, $cl(E, N) \in MaxProperCh(C)$. Hence D is not in $MaxProperCh(C)$ unless $D = cl(E, N)$, proving Claim 1b.

This completes the proof of Claim 1.

We now complete the proof of Theorem 3.1.

We saw above that the procedure terminates, say with M_n . By Claim 1, H_n will be true. In fact, each vertex W of N has been represented in M_n in the sense that $cl(W, N) \in V_n$. To see this, note that there is a directed path $P_0 = r, P_1, \dots, P_k = W$ in N since r is the root of N . Since $X' = r$, by Claim 1a it follows that $cl(P_1, N)$ is a member of $MaxProperCh(X)$, whence by construction $cl(P_1, N) \in V_n$, $cl(P_1, N)' = P_1$, and $(cl(r, N), cl(P_1, N))$ in A_n . Since $cl(P_1, N)' = P_1$ and P_2 is a child of P_1 in N , by Claim 1a again it follows that $cl(P_2, N) \in V_n$, $cl(P_2, N)' = P_2$, and $(cl(P_1, N), cl(P_2, N)) \in A_n$. Repeating the argument we ultimately obtain that $cl(P_k, N) = cl(W, N)$ in V_n .

Since every arc in N lies on some directed path in N starting at r hence occurs as some arc (P_i, P_{i+1}) using the notation above, the same argument shows that the arc corresponds to the arc $(cl(P_i, N), cl(P_{i+1}, N)) \in A_n$. Thus every vertex and arc of N has a corresponding vertex and arc in M_n .

There remains only to show that M_n has no additional vertices or arcs. By Claim 1b every vertex which is added at any stage has the form $cl(E, N)$ for some vertex E of N . Hence M_n has no additional vertices. By claim 1a, every arc in M_n corresponds to an arc in N .

Note that the argument has shown that when $U = cl(U', N)$, then the children of U in M_n will be precisely the clusters $cl(C', N)$ where C' is a child of U' in N . This means that M_n is the cover digraph of N .

This completes the proof. \square

Corollary 3.4. Suppose $N = (V, A, r, X)$ is a regular

phylogenetic network. Let v be the number of vertices of N and $n = |X|$ be the number of leaves. Suppose \mathcal{D} consists of d trees. Then Maximal Proper Child takes time $O(vn^2d^2)$.

Proof: Note that the algorithm proceeds by identifying all the children of a cluster W , hence in v steps. For each cluster W , one finds the proper trees for the cluster W as follows: Suppose that the constructed network so far is M_k and one seeks the children of W . For each tree $T \in \mathcal{D}$, one checks whether W is a cluster of T . If not, then T is ignored; if so, then check whether the parental cluster U to W in T is a parent of W in M_k . If not, then T is ignored. If U is in M_k and U is the root, then T is a proper tree for W . If not, then let Y denote the parent of U in T , and check whether Y is a parent of U in M_k , rejecting T if Y is not a parent of U in M_k . Repeat the procedure until either T has been rejected or the root has been reached, in which case T is a proper tree for the original W . The children of W in such a tree can be found in time $O(n)$. Hence $ProperCh(W)$ can be computed in time $O(nd)$. Accordingly, the number of clusters in $ProperCh(W)$ is at most $O(nd)$.

If a collection C of k subsets of X is given, pairwise comparison of each member of C with every other member lets one tell in time $O(k^2)$ which subsets are maximal. Since $ProperCh(W)$ has $k = O(nd)$ members, it follows that $MaxProperCh(W)$ can be computed in time $k^2 = O(n^2d^2)$.

Since the procedure is to be repeated for v vertices, the total time to find all the vertices is $O(vn^2d^2)$. \square

4 EXTENSIONS

Theorem 3.1 need not be true without the assumption that \mathcal{D} includes all the trees displayed by N . It is easy to find examples in which N is not reconstructed if some trees are missing from \mathcal{D} . On the other hand, it is also easy to find examples in which $\mathcal{D} \neq Tr(N)$ but still $MPC(\mathcal{D}) = N$. What is important is that enough of the ‘‘right’’ trees lie in \mathcal{D} . Roughly speaking, the ‘‘right’’ trees are those that are complete for directed paths starting at the root.

The next results show that there is in general a collection of trees \mathcal{D} much smaller than $Tr(N)$ such that modified versions of the procedure will output N . Indeed the number of trees in \mathcal{D} can be chosen to be the number of arcs in N provided that the trees have certain properties.

Define the *height* of a network N to be the maximal number of arcs on a directed path from the root to a vertex of N .

Theorem 4.1. *Let $N = (V, A, r, X)$ be a regular network with leaf set X . Let \mathcal{D} consist of a collection of displayed trees such that, for every arc $(u, v) \in A$ there exists a directed path $r = v_0, v_1, \dots, v_{k-1} = u, v_k = v$ in N and a tree $T \in \mathcal{D}$ such that T is complete for the path v_0, \dots, v_k . Let $d = |\mathcal{D}|$, $v = |V|$, $n = |X|$, and let h be the height of N .*

There is a modified procedure which reconstructs N from \mathcal{D} in time $O(vhn^2d^2) \leq O(vn^3d^2)$.

Proof: The procedure is the same as Maximal Proper Child except that we add a step to pick the order in which the vertices are considered.

The difficulty we must address is the following: Suppose we consider a vertex U in M_k that has not been checked off. Some parent P of U in M_k will have been identified. Note, however, that not all parents of U may have been identified yet, so maybe the parent Q of U is not yet identified. Suppose W is a child of U . It is possible that \mathcal{D} may contain a tree complete for a path containing the successive vertices Q, U, W but not any path involving the successive vertices P, U, W , so the procedure could not identify W as a child of U .

To remedy the difficulty we arrange the order in which we consider vertices so that we check off a vertex U only when all its ancestors in N have already been identified.

Suppose U is a vertex of M_k which has not been checked off (and is not a singleton or a doubleton). We classify each tree $T \in \mathcal{D}$ as follows:

- (1) T is *irrelevant for U* if U is not a cluster of T .
- (2) T is *proper for U* if U is a cluster of T , say $U = cl(v, T)$, and the path $r = v_0, v_1, \dots, v_j = v$ in T (which is uniquely determined since T is a tree) satisfies that $cl(r, T), cl(v_1, T), \dots, cl(v_j, T)$ is a directed path in M_k from $cl(r, T)$ to U .
- (3) T is *certified improper for U* if U is a cluster of T , say $U = cl(v, T)$, and the path $r = v_0, v_1, \dots, v_j = v$ in T satisfies that for some i such that $0 \leq i < j$, $cl(r, T), cl(v_1, T), \dots, cl(v_i, T)$ is a directed path in M_k , $cl(v_i, T)$ has been checked off, but $cl(v_{i+1}, T)$ is not a child of $cl(v_i, T)$ in M_k . In this case, note T is definitely not proper.
- (4) T is *undetermined for U* if U is a cluster of T , say $U = cl(v, T)$, and the path $r = v_0, v_1, \dots, v_j = v$ in T satisfies that for some i such that $0 \leq i < j$, $cl(r, T), cl(v_1, T), \dots, cl(v_i, T)$ is a directed path in M_k , but $cl(v_i, T)$ has not been checked off. In this case note that it is possible that $cl(v_{i+1}, T)$ is the cluster of a child of the vertex corresponding to $cl(v_i, T)$ in N , but it is also possible that it is not such a child.

If there are no trees in \mathcal{D} that are undetermined for U , then proceed as in the algorithm Maximal Proper Child to identify the children of U , form M_{k+1} , and check off U . (Identify the children of U in all proper trees; the children of U in M_{k+1} will be the maximal proper children.) For each child W of U in N , by hypothesis \mathcal{D} contains a tree T complete for a path from r to W through U . This tree T will be proper for U with child W . Hence the argument for Theorem 3.1 shows that all the children of U will be identified by the procedure.

If there is an undetermined tree T for U , then we reject U , do not check it off, and consider another vertex. More specifically for some undetermined tree T for U , consider a path $r = v_0, v_1, \dots, v_j = U$ in T that satisfies that for some i such that $0 \leq i < j$, $cl(r, T), cl(v_1, T), \dots, cl(v_i, T)$ is a directed path in M_k , but $cl(v_i, T)$ has not been checked off. Let $W := cl(v_i, T)$. We replace U by W and try again, classifying each tree in \mathcal{D} for W . If there is an

undetermined tree for W , we reject W and have another candidate. Since each new candidate is a strict ancestor of the previous candidate, the procedure must ultimately find a vertex not yet checked off such that \mathcal{D} contains no undetermined tree for the vertex. This vertex will then be checked off by the procedure.

If h is the height of N , then clearly there are at most h vertices that are rejected as above before a vertex is found that can be checked off. Thus each step may take time $O(hn^2d^2)$ instead of taking time $O(n^2d^2)$.

It follows that N can be reconstructed in time $O(vhn^2d^2)$. Since N is regular, any path from the origin must remove at each step at least one member of X , whence $h \leq n$. It follows that N can be reconstructed in time $O(vn^3d^2)$. \square

Let a be the number of arcs in N . There clearly exists \mathcal{D} as described in Theorem 4.1 such that $d = a$ since we may choose one such tree for each arc (u, v) . Of course, it is possible that one tree satisfies the conditions simultaneously for two arcs, so for such a collection $d \leq a$.

Corollary 4.2. *Suppose N is regular and has a arcs. There exists \mathcal{D} with $d \leq a$ trees such that N can be reconstructed by the method of Theorem 4.1 from \mathcal{D} in time $O(vhn^2a^2)$ or $O(vn^3a^2)$.*

A network $N = (V, A, r, X)$ is *normal* provided that N is regular and in addition each vertex that is not a leaf has a normal child (i.e., a child that is not hybrid). An equivalent definition is given in [25]. It follows easily that starting at every vertex v there exists a directed path $v = v_0, v_1, \dots, v_k$ such that v_k is a leaf, whence $v_k \in X$, and such that the vertex v_i is normal for $i > 0$. Such a path will be called a *normal path* from v to $v_k \in X$. (If v is a leaf, then the trivial path at v satisfies the condition.)

Since N is regular, it contains no redundant arcs. Moreover, note that if $x \in X$ is a leaf then there is a unique maximal normal path (possibly trivial) leading to x , since each vertex on the path (being normal) other than the starting vertex of the maximal such path has a unique parent. It follows that if there is a normal path from v to $x \in X$ and $w \leq x$, then either $v \leq w$ (and w lies on the normal path from v to x) or $w \leq v$.

We will see that a normal network can be reconstructed using the following simpler procedure called Maximal Child.

Algorithm Maximal Child.

Input: \mathcal{D} is a nonempty collection of rooted trees each with the base-set X .

Output: A regular phylogenetic network M with base-set X .

Procedure. We construct a sequence S_0, S_1, \dots of nonempty subsets of X , each member of which either is or is not checked off.

1. Initially $S_0 = \{X\}$. Thus S_0 has a single vertex which is the cluster X . This vertex is not checked off.

Recursively perform the following step 2:

2. Suppose S_k is known and some member $U \in S_k$ is not checked off.

2a. If U is a singleton set $U = \{a\}$ which is not checked off, then $S_{k+1} = S_k$ except that U has been checked off.

2b. Suppose 2a does not apply. Suppose $U \in S_k$ has not been checked off. Let $Tr(U) = \{T \in \mathcal{D} : U \text{ is a cluster of } T\}$ be the set of input trees containing U .

Let $Ch(U) = \{C : \text{for some } T \in Tr(U), u \text{ is a vertex of } T \text{ such that } U = cl(u), c \text{ is a child of } u \text{ in } T, \text{ and } C = cl(c, T)\}$. Essentially $Ch(U)$ is the set of all clusters of all children of U in any tree containing the cluster U . Let $MaxCh(U) = \{C \in Ch(U) : \text{there is no } C' \text{ in } Ch(U) \text{ such that } C \subset C' \subset U\}$ (strict inclusions). This is the set of maximal children of U .

For each $C \in MaxCh(U)$, adjoin to S_k the set C (if it is not already present). More explicitly define $S_{k+1} = S_k \cup \{C : C \in MaxCh(U)\}$. Check off U but do not check off any newly added $C \in MaxCh(U)$.

3. The procedure terminates with S_n such that every member of S_n has been checked off.

4. Return the Hasse graph of S_n . More explicitly return the directed graph with vertex set S_n and with an arc (U, W) precisely when U and W are members of S_n such that $W \subset U$ and there is no $Z \in S_n$ such that $W \subset Z \subset U$.

Theorem 4.3. *Suppose $N = (V, A, r, X)$ is a normal network. Suppose the input \mathcal{D} is a collection of trees displayed by N . Suppose for every arc (u, v) of N there exists a tree $T \in \mathcal{D}$ with the arc (u', v') such that $cl(u', T) = cl(u, N)$ and $cl(v', T) = cl(v, N)$. Then the procedure Maximal Child outputs the cover digraph of N .*

The heart of the proof is the following lemma:

Lemma 4.4. *Assume $N = (V, A, r, X)$ is normal and $u \in V$. Suppose $U = cl(u, N)$. Suppose T is a tree displayed by N and there exists a vertex u' of T such that $cl(u', T) = U$. Let c' be a child of u' in T . Then there exists a child c of u in N such that $cl(c', T) \subseteq cl(c, N)$.*

Proof: If u is a leaf of N , then $u \in X$ and $U = \{u\}$. Hence $cl(u', T) = \{u\}$ and u' has no child, so the claim is moot. We may then assume u is not a leaf of N . Hence u has at least two children c_1 and c_2 in N (If there were only one child c_1 then $cl(c_1, N) = cl(u, N)$, contradicting regularity.) By normality choose a normal path from c_1 to $x_1 \in X$ and from c_2 to $x_2 \in X$. Then $\{x_1, x_2\} \subseteq U$. Moreover $x_1 \neq x_2$ since if $x_1 = x_2$ then either $c_1 \leq c_2$ or $c_2 \leq c_1$ by following the (unique) normal path to x_1 , which is impossible since then there would be a path from c_1 to c_2 , contradicting nonredundancy of the arc (u, c_2) , or else a path from c_2 to c_1 , contradicting nonredundancy of the arc (u, c_1) .

Let $T = Tr(N_p)$ for the parent map p . We may regard u' as a vertex of N_p and hence of N . Then $cl(u, N) = U = cl(u', T) = cl(u', N_p) \subseteq cl(u', N)$. By regularity of N it follows that $u' \leq u$ in N . Hence there is a directed path in N from u' to u . Indeed, there is a directed path in

N_p from u' to u . To see this, consider the normal path in N from u to some $x \in X$. Then $x \in cl(u, N)$, and since the path is normal, all its arcs are arcs of N_p , whence $x \in cl(u, N_p)$. Since $cl(u, N_p) \subseteq cl(u, N) = cl(u', T)$, it follows $x \in cl(u', T) = cl(u', N_p)$ whence there is a path in N_p from u' to x , which must coincide in part with the normal path from u to x .

I claim that each vertex on the path in N_p from u' to u (other than u' and u themselves) has indegree 1 and outdegree 1 in N_p . Since N_p has no hybrid vertices, each such vertex has indegree 1. If a vertex w had outdegree greater than 1, then w would have a child c_2 in N_p distinct from the child c_1 of w along the directed path in N_p from u' to u . Since c_2 is a vertex of N , there is a normal path in N from c_2 to some $y \in X$. By the definition of N_p each arc in the normal path must lie in N_p , whence $y \in cl(c_2, N_p)$. It follows that $y \in cl(u', N_p) = U = cl(u, N)$. Hence there is a directed path in N from u to y . Since there is a normal path from c_2 to y , either u lies on this normal path or else there is a path in N from u to c_2 . In the former case, since the path lies in N_p as well, we would have N_p containing a hybrid vertex so that the path from c_2 can reach u , contradicting that N_p is a tree. But in the latter case, we either have a path in N from u to w (contradicting that N is acyclic) or else we have a path in N from u to another parent of c_2 other than w . In this latter case the arc (w, c_2) would be redundant since there would be an alternative path from w to u to c_2 . Hence neither case can occur, proving the claim.

By a similar argument, if $u \neq u'$, then u' has outdegree 1 in N_p . It then follows that the children of u' in T must be descendants of u in N_p . Each child of u' in T is therefore a descendent of some child of u in N . The lemma follows. \square

We may now prove Theorem 4.3.

Proof: The proof of the theorem is inductive. Suppose u is a vertex of N and $U = cl(u, N)$ has been found by the procedure. (This is trivially true for the initial situation where $u = r$ and $U = X$.) Suppose v is a child of u in N . By the assumption on \mathcal{D} there exists a tree T in \mathcal{D} with vertices u' and v' , and arc (u', v') such that $cl(u', T) = cl(u, N)$ and $cl(v', T) = cl(v, N)$. Hence $T \in Tr(U)$, and $cl(v, N) \in Ch(U)$. If w is another child of u in N distinct from v , then, since N is regular, $cl(w, N) \neq cl(v, N)$; moreover, since neither v nor w is a descendent of the other, it is impossible that either $cl(w, N) \subset cl(v, N)$ or $cl(v, N) \subset cl(w, N)$. By Lemma 4.4, for every tree T'' containing a vertex u'' satisfying $cl(u'', T'') = U$, and for every child c'' of u'' in T'' , there is a child c of u in N such that $cl(c'', T) \subseteq cl(c, N)$. Hence the members of $MaxCh(U)$ are precisely the clusters $cl(c, N)$ for c a child of u in N . \square

Suppose that the normal network N satisfies that there are v vertices, $|X| = n$, and \mathcal{D} contains d trees. Then Maximal Child will terminate after time $O(vn^2d^2)$, by an argument similar to that for Corollary 3.4. There exists \mathcal{D} satisfying the hypotheses of Theorem 4.3 such that

$d \leq a$, the number of arcs in N . Given such \mathcal{D} we see that N can be reconstructed in time $O(vn^2a^2)$. In fact, it may be seen that in a normal network $v = O(n^2)$ and $a = O(n^2)$, see [26]. Hence given \mathcal{D} satisfying the hypotheses of Theorem 4.3 and with a members, it follows that N can be reconstructed in time $O(n^2n^2(n^2)^2) = O(n^8)$.

5 AN EXAMPLE OF THE RECONSTRUCTION

In this section we illustrate the procedure Maximal Proper Child, used in Theorem 3.1. Let N be the network given in Figure 4. The base-set is $X = \{1, 2, 3, 4, 5, 6\}$. The clusters satisfy $cl(A) = X$, $cl(B) = \{1, 2, 3, 4, 6\}$, $cl(C) = \{5, 6\}$, $cl(D) = \{1, 2, 3, 6\}$, $cl(E) = \{2, 3\}$, $cl(F) = \{1, 2, 6\}$, $cl(G) = \{1, 2, 3\}$, and $cl(i) = \{i\}$ for $1 \leq i \leq 6$. An inspection shows that N is regular.

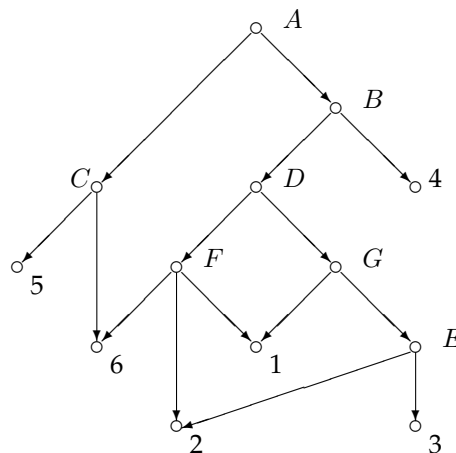


Fig. 4. A regular network N with $X = \{1, 2, 3, 4, 5, 6\}$ which will be reconstructed from its trees.

There are three hybrid vertices 1, 2, 6, each with indegree 2. Hence there are 8 parent maps. Here I will list the displayed trees by telling the parent map and the nontrivial clusters of each:

T_1 : $p(1) = G, p(2) = E, p(6) = F$. Clusters $\{2, 3\}, \{1, 2, 3\}, \{1, 2, 3, 6\}, \{1, 2, 3, 4, 6\}$.

T_2 : $p(1) = G, p(2) = E, p(6) = C$. Clusters $\{2, 3\}, \{1, 2, 3\}, \{1, 2, 3, 4\}, \{5, 6\}$.

T_3 : $p(1) = G, p(2) = F, p(6) = F$. Clusters $\{1, 3\}, \{2, 6\}, \{1, 2, 3, 6\}, \{1, 2, 3, 4, 6\}$.

T_4 : $p(1) = G, p(2) = F, p(6) = C$. Clusters $\{1, 3\}, \{1, 2, 3\}, \{1, 2, 3, 4\}, \{5, 6\}$.

T_5 : $p(1) = F, p(2) = E, p(6) = F$. Clusters $\{2, 3\}, \{1, 6\}, \{1, 2, 3, 6\}, \{1, 2, 3, 4, 6\}$.

T_6 : $p(1) = F, p(2) = E, p(6) = C$. Clusters $\{2, 3\}, \{1, 2, 3\}, \{1, 2, 3, 4\}, \{5, 6\}$.

T_7 : $p(1) = F, p(2) = F, p(6) = F$. Clusters $\{1, 2, 6\}, \{1, 2, 3, 6\}, \{1, 2, 3, 4, 6\}$.

T_8 : $p(1) = F, p(2) = F, p(6) = C$. Clusters $\{1, 2\}, \{1, 2, 3\}, \{1, 2, 3, 4\}, \{5, 6\}$.

We now perform procedure Maximal Proper Child with input $\mathcal{D} = Tr(N)$. Let $M_k = (V_k, A_k)$. Initially $V_0 = \{X\}$. The proper children of

X are the children of X in any proper tree. All the trees are proper trees for X . Hence $ProperCh(X) = \{\{1, 2, 3, 4, 6\}, \{5\}, \{1, 2, 3, 4\}, \{5, 6\}\}$. The maximal proper children are the maximal members of $ProperCh(X)$. Hence $MaxProperCh(X) = \{\{1, 2, 3, 4, 6\}, \{5, 6\}\}$. These are adjoined to M_0 as children of X . Hence $M_1 = (V_1, A_1)$ has $V_1 = \{X, \{1, 2, 3, 4, 6\}, \{5, 6\}\}$ and has arcs $(X, \{1, 2, 3, 4, 6\})$ and $(X, \{5, 6\})$.

Let $C = \{5, 6\}$ in V_1 . By 2b, the children will be $\{5\}$ and $\{6\}$. Hence M_2 has $V_2 = \{X, \{1, 2, 3, 4, 6\}, \{5, 6\}, \{5\}, \{6\}\}$ and the arcs are those of M_1 together with $(\{5, 6\}, \{5\})$ and $(\{5, 6\}, \{6\})$.

Let $C = \{1, 2, 3, 4, 6\}$. The proper trees must contain both X and $\{1, 2, 3, 4, 6\}$. Hence $ProperTr(C) = \{T_1, T_3, T_5, T_7\}$. The proper children of C are the children of C in one of the proper trees. Hence $ProperCh(C) = \{\{1, 2, 3, 6\}, \{4\}\}$. In this case all proper children are maximal. Hence M_3 has $V_3 = V_2 \cup \{\{1, 2, 3, 6\}, \{4\}\}$ and suitable arcs are also added.

Let $C = \{1, 2, 3, 6\}$. A proper tree must contain C , some parent of C hence $\{1, 2, 3, 4, 6\}$, and X . Thus $ProperTr(C) = \{T_1, T_3, T_5, T_7\}$. The proper children are the children of C in any of these proper trees, so $ProperCh(C) = \{\{1, 2, 3\}, \{6\}, \{1, 3\}, \{2, 6\}, \{1, 6\}, \{2, 3\}, \{1, 2, 6\}, \{3\}\}$. Then $MaxProperCh(C) = \{\{1, 2, 3\}, \{1, 2, 6\}\}$. These are adjoined, so $V_4 = V_3 \cup \{\{1, 2, 3\}, \{1, 2, 6\}\}$ and arcs are inserted so that these are the children in M_4 of $\{1, 2, 3, 6\}$.

Let $C = \{1, 2, 3\}$. A proper tree must contain $\{1, 2, 3\}$, $\{1, 2, 3, 6\}$, $\{1, 2, 3, 4, 6\}$, and X . Hence $ProperTr(C) = \{T_1, T_6\}$. Then $ProperCh(C) = \{\{1\}, \{2, 3\}\} = MaxProperCh(C)$. Now $V_5 = V_4 \cup \{\{1\}, \{2, 3\}\}$.

Let $C = \{1, 2, 6\}$. A proper tree must contain $\{1, 2, 6\}$, $\{1, 2, 3, 6\}$, $\{1, 2, 3, 4, 6\}$, and X . Hence $ProperTr(C) = \{T_7\}$. It follows that $ProperCh(C) = \{\{1\}, \{2\}, \{6\}\} = MaxProperCh(C)$. Now $V_6 = V_5 \cup \{\{1\}, \{2\}, \{6\}\}$. Note that $\{6\}$ was already in V_5 , but it is at this stage that we obtain the arc $(\{1, 2, 6\}, \{6\})$.

Let $C = \{2, 3\}$. By 2b the children will be $\{2\}$ and $\{3\}$. Hence $V_7 = V_6 \cup \{\{2\}, \{3\}\}$.

The procedure terminates now with M_7 . Note that V_7 now consists of exactly the sets $cl(U, N)$ where U is a vertex of N . Similarly the arcs of M_7 consist exactly of the arcs $(cl(U, N), cl(W, N))$ such that (U, W) is an arc of N . Thus M_7 is isomorphic with N ; indeed, it is the cover digraph of N .

It is natural to wish that the identification of the children could be simplified, for example by using the procedure Maximal Child. This alternative approach, however, fails on this example. Note that N is not normal, so Theorem 4.3 does not apply. If we did not insist on proper trees, then $\{1\}$ is not a maximal child of $\{1, 2, 3\}$ since T_8 contains $\{1, 2, 3\}$ with the child $\{1, 2\}$. Maximal Proper Child works since T_8 is not a proper tree for $\{1, 2, 3\}$ because the parent of $\{1, 2, 3\}$ in T_8 is $\{1, 2, 3, 4\}$ which had not been identified as a cluster in N .

The input $\mathcal{D} = \{T_1, T_2, T_7\}$ satisfies the hypotheses of Theorem 4.1, so the Procedure of Theorem 4.1 reconstructs N from this smaller \mathcal{D} . In fact, Maximal Proper Child also reconstructs N from this \mathcal{D} as well. On the other hand, if $\mathcal{D} = \{T_1, T_2, T_3\}$, then N is not reconstructed.

6 DISCUSSION

The main result in this paper is that, if $N = (V, A, r, X)$ is a regular network, then the polynomial-time procedure Maximal Proper Child will reconstruct N from the collection $Tr(N)$ of all trees displayed by N . Theorem 4.1 shows that not all trees in $Tr(N)$ need to be input, but only some trees that satisfy certain conditions.

In a given applied situation, however, a biologist probably has available only a comparatively small collection of gene trees for various genes. Even if the given gene trees are all displayed by the relevant network as assumed in this paper, it is unlikely that all the trees displayed by the network are represented in the data. One would not know in advance whether the collection of data trees satisfies the hypotheses of these theorems.

There are additional complications. Besides the factors discussed in this paper, other factors could give rise to variation in the gene trees. For example, lineage sorting, as seen in coalescent models [8], [21], [22] may also be present.

One could still, however, apply the algorithms to the collection of gene trees and obtain a network N . If the network N is consistent with the data, this fact can support the hypothesis that N tells the phylogeny of the relevant species.

For example, Rokas *et al.* [20] analyzed a set of 106 yeast genes from total database with 127,026 nucleotide sites. There were 7 yeast genomes, genus *Saccharomyces*, and one outgroup from genus *Candida*. They concatenated the aligned genes and obtained a tree with 100% bootstrap support at each internal vertex.

Using their data set, we may instead compute the 106 maximum-likelihood trees. There are 19 distinct trees, which we may list in order of the frequency of occurrence. Tree 1 occurs 45 times as a gene tree, tree 2 occurs 19 times as a gene tree, tree 3 occurs 8 times as a gene tree, and all other trees occur at most 5 times as a gene tree.

Suppose that as input to the algorithm Maximal Proper Child or Maximal Child we use the most common gene trees—e.g., trees 1, 2, and 3 in the list above. We obtain the network N in Figure 5.

In fact each of the three input trees is displayed by N , and the fourth displayed tree is one of the trees that occurs exactly once as an observed gene tree. These facts are internal evidence that the network N is consistent with these data. Moreover, network N bears an intriguing close resemblance to the consensus network found by Holland *et al.* [12], Fig. 1c, for the maximum parsimony trees for the same dataset.

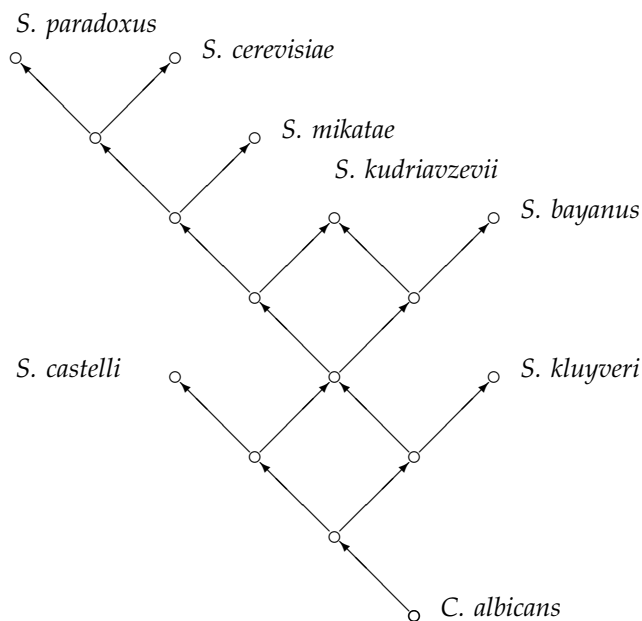


Fig. 5. The normal network which results from utilizing Maximal Child with input the three most common gene trees for the data of [20].

If, on the other hand, we also include tree 4 that occurred 5 times as a gene tree, then the resulting network M displays 16 trees of which only 7 are observed, undermining confidence that M could be correct.

Thus the methods of this paper could be applied to real data to yield networks such as N that are candidates for the phylogeny and eliminate networks such as M .

Acknowledgments

I wish to thank Mike Steel and Vincent Moulton for helpful discussions. I also wish to thank the anonymous referees for improvements of an earlier version of the paper and for additional references.

REFERENCES

- [1] L. Arvestad, A.-C. Berglund, J. Lagergren, and B. Sennblad, 2004. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. RECOMB '04, 326-335.
- [2] H.-J. Bandelt and A. Dress, 1992. Split decomposition: a new and useful approach to phylogenetic analysis of distance data. Molecular Phylogenetics and Evolution 1, 242-252.
- [3] M. Baroni, C. Semple, and M. Steel, 2004. A framework for representing reticulate evolution. Annals of Combinatorics 8, 391-408.
- [4] M. Baroni and M. Steel, 2006. Accumulation phylogenies. Annals of Combinatorics 10, 19-30.
- [5] M. Bordewich and C. Semple, 2007. Computing the minimum number of hybridization events for a consistent evolutionary history. Discrete Applied Mathematics 155, 914-928.
- [6] G. Cardona, F. Rossalló, and G. Valiente, 2007. Comparison of tree-child phylogenetic networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics, preprint, Dec. 2007, doi:10.1109/TCBB.2007.70270.
- [7] G. Cardona, L. Mercè, F. Rossalló, and G. Valiente, 2008. A distance metric for a class of tree-sibling phylogenetic networks. Bioinformatics 24(13), 1481-1488.

- [8] J. Degnan and N. Rosenberg, 2006. Discordance of species trees with their most likely gene trees. PLoS Genetics 2(5)e68, 762-768.
- [9] D. Gusfield, S. Eddhu, and C. Langley, 2004. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. Journal of Bioinformatics and Computational Biology 2, 173-213.
- [10] M. Hallett and J. Lagergren, 2000. New algorithms for the duplication-loss model. RECOMB 2000, 138-146.
- [11] M. Hallett and J. Lagergren, 2001. Efficient algorithms for lateral gene transfer problems. RECOMB 2001, 149-56.
- [12] B. Holland, K. Huber, V. Moulton, and P. Lockhart, 2004. Using consensus networks to visualize contradictory evidence for species phylogeny. Molecular Biology and Evolution 21(7), 1459-1461.
- [13] D. Huson, T. Klöpper, P. Lockhart, and M. Steel, 2005. Reconstruction of reticulate networks from gene trees. In S. Miyano et al. (eds.) RECOMB 2005, LNBI 3500, 233-249.
- [14] D. Huson and T. Klöpper, 2007. Beyond galled trees—decomposition and computation of galled networks. In T. Speed and H. Huang (eds.): RECOMB 2007, LNBI 4453, 211-225.
- [15] I. Kanj, L. Nakhleh, C. Than, and G. Xia, 2008. Seeing the trees and their branches in the network is hard. Theoretical Computer Science 401, 153-164.
- [16] B. Moret, L. Nakhleh, T. Warnow, C. R. Linder, A. Tholse, A. Padolina, J. Sun, and R. Timme, 2004. Phylogenetic networks: modeling, reconstructibility, and accuracy. IEEE Transactions on Computational Biology and Bioinformatics 1, 13-23.
- [17] L. Nakhleh, T. Warnow, C.R. Linder, and K.S. John, 2005. Reconstructing reticulate evolution in species: Theory and practice. J. Comput. Biol 12, 796-811.
- [18] R.D.M. Page and M.A. Charleston, 1997a. Reconciled trees and incongruent gene and species trees. In Mathematical hierarchies and biology, edited by B. Mirkin, F.R. McMorris, F.S. Roberts, and A. Rzhetsky, 57-70. Providence, R.I.: American Mathematical Society.
- [19] R.D.M. Page and M.A. Charleston, 1997b. From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem. Molecular Phylogenetics and Evolution 7, 231-240.
- [20] A. Rokas, B. Williams, N. King, and S. Carroll, 2003. Genome-scale approaches to resolving incongruence in molecular phylogenies. Nature 425, 798-804.
- [21] N. A. Rosenberg, 2002. The probability of topological concordance of gene trees and species trees. Theoretical Population Biology 61, 225-247.
- [22] N. A. Rosenberg, 2007. Counting coalescent histories. Journal of Computational Biology 14(3), 360-377.
- [23] K. Strimmer and V. Moulton, 2000. Likelihood analysis of phylogenetic networks using directed graph models. Molecular Biology and Evolution 17, 875-881.
- [24] L. Wang, K. Zhang, and L. Zhang, 2001. Perfect phylogenetic networks with recombination. Journal of Computational Biology 8, 69-78.
- [25] S.J. Willson, 2008. Reconstruction of certain phylogenetic networks from the genomes at their leaves. Journal of Theoretical Biology 252, 338-349.
- [26] S.J. Willson, 2009. Properties of normal phylogenetic networks. To appear in Bulletin of Mathematical Biology.

PLACE
PHOTO
HERE

Stephen Willson Stephen J. Willson received his A.B. in Mathematics from Harvard in 1968. In 1973 he received his Ph.D. in Mathematics from the University of Michigan in Ann Arbor. His dissertation was in algebraic topology under the supervision of A.G. Wasserman.

He went to Iowa State University in Ames, Iowa in 1973, where he is currently Janson Professor of Mathematics. His research interests include phylogenetics, fractals, and game theory. His hobbies include classical piano, choral singing, bird-watching, bicycling, and kayaking.