Towards the Development of Computational Tools for Evaluating Phylogenetic Network Reconstruction Methods

Luay Nakhleh Jerry Sun Tandy Warnow
Dept. of Computer Sciences, U. of Texas, Austin, TX 78712
{nakhleh,jsun,tandy}@cs.utexas.edu

C. Randal Linder
Section of Integrative Biology
University of Texas
Austin, TX 78712
rlinder@mail.utexas.edu

Bernard M.E. Moret Anna Tholse
Dept. of Computer Science
University of New Mexico
Albuquerque, NM 87131
{moret,tholse}@cs.unm.edu

We report on a suite of algorithms and techniques that together provide a simulation flow for studying the topological accuracy of methods for reconstructing phylogenetic networks. We implemented those algorithms and techniques and used three phylogenetic reconstruction methods for a case study of our tools. We present the results of our experimental studies in analyzing the relative performance of these methods. Our results indicate that our simulator and our proposed measure of accuracy, the latter an extension of the widely used Robinson-Foulds measure, offer a robust platform for the evaluation of network reconstruction algorithms.

1 Introduction

Phylogenies, i.e., the evolutionary histories of groups of organisms, play a major role in representing the interrelationships among biological entities. Many methods for reconstructing such phylogenies have been proposed, but almost all of them assume that the underlying evolutionary history of a given set of species can be represented by a tree. While this model gives a satisfactory first-order approximation for many families of organisms, other families exhibit evolutionary mechanisms that cannot be represented by a tree. Processes such as hybridization and horizontal gene transfer result in networks of relationships rather than trees of relationships. Although this problem is widely appreciated, there has been comparatively little work on computational methods for estimating evolutionary networks.

A standard technique for assessing the performance of phylogenetic reconstruction methods is to use simulation studies. In such studies, a model topology (tree or network) is generated, after which a sequence is evolved (including bifurcations and non-treelike events) down the edges of the model topology according to some chosen model of sequence evolution. Finally, a phylogeny is reconstructed on the resulting set of sequences and its topology is compared to the model topology in order to assess the quality, or topological accuracy, of the reconstruction. While many simulation tools and accuracy measures are available for studying the performance of phylogenetic tree reconstruction methods, such tools and measures are lacking in the context of phylogenetic networks.

We describe a collection of such techniques and quality measures: (i) a technique for generating random phylogenetic networks and simulating the evolution of sequences on these networks, and (ii) measures to assess the topological accuracy of the reconstructed networks. We implemented those techniques and conducted a simulation study on one network reconstruction method, `SplitsTree`[1], and two tree reconstruction methods, *Neighbor-Joining* and *greedy Maximum Parsimony*. We assessed the performance of these methods on datasets generated in our simulation.

The rest of the paper is organized as follows. Section 2 provides some background on phylogenetic trees and describes the various steps involved in a simulation study. Section 3 introduces our new techniques and reviews the existing reconstruction methods used in our study. Section 4 describes our experimental setup, while Section 5 reports the results of our experiments and offers some remarks on the usefulness of our simulation flow.

2 Phylogenetic Trees

A *phylogenetic tree* on a set S of taxa is a rooted tree whose leaves are labeled by S. Such a tree represents the evolutionary history of a set of taxa, where the leaves of the phylogenetic tree correspond to the extant taxa and the internal nodes represent the (hypothetical) ancestors. Many algorithms have been designed for the inference of phylogenetic trees, mainly from biomolecular (i.e., DNA, RNA, or amino-acid) sequences[2]. Evaluating these algorithms cannot be done with real data alone, since we typically do not know with high confidence the details of the "true" evolutionary history. Thus the standard means of performance evaluation for phylogenetic reconstruction methods is the simulation study, in which "model" phylogenies are constructed using some chosen model of evolution, the "modern data" (found in the leaves of the model phylogeny) are fed to the reconstruction algorithms, and the output of the algorithms compared with the model phylogeny.

2.1 Model trees

Model trees are typically taken from some underlying distribution on all rooted binary trees with n leaves; commonly used distributions include the uniform distribution and the Yule-Harding distribution[3,4]. Although we generate networks rather than trees, we have based our network generation on the widely used model of birth-death evolution, which we now briefly review in the context of tree generation.

To generate a random birth-death tree on n leaves, we view speciation and extinction events as occurring over a continuous interval. During a short time interval, Δt, since the last event, a species can split into two with probability $b(t)\Delta t$ or become extinct with probability $d(t)\Delta t$. To generate a tree with n taxa, we begin this process with a single node and continue until we have a tree with n leaves. (With some nonzero probability some processes will not produce a tree of the desired size, since all nodes could go "extinct" before n species are generated; we then repeat the process until a tree of the desired size is generated.) Under this distribution, a natural length is associated with each edge, namely the time elapsed between the speciation event that gave rise to that edge and the (speciation or extinction) event that ended that edge. Thus birth-death trees are inherently ultrametric, that is, the branch lengths are proportional to time.

2.2 Tree reconstruction methods

In our experiments, we used one network reconstruction method and two tree reconstruction methods, neighbor-joining (NJ) and greedy maximum parsimony (MP).

- Neighor-joining [5] is the most popular distance-based method. For every pair of taxa, it determines a score based on the pairwise distance matrix, then joins the pair with the smallest score, building a subtree of two leaves whose root replaces the two chosen taxa in the matrix. Pairwise distances for this new "supertaxon" are then recalculated and the entire process is repeated until only three nodes remain; these are then joined to form an unrooted binary tree.
- Maximum parsimony [6] is one of the two main optimization criteria used in phylogenetic reconstruction. Because the problem is NP-hard, we use a simple greedy heuristic [7,8], which adds taxa to the tree one at a time in some random order—the placement of each new taxon is optimized locally.

2.3 Measures of accuracy

A commonly used measure of the topological accuracy of reconstructed trees is the *Robinson-Foulds (RF)* value [9]. Every edge e in an unrooted leaf-labeled tree T defines a bipartition π_e on the leaves (deleting e cuts the tree); we set $C(T) = \{\pi_e : e \in E(T)\}$, where $E(T)$ is the set of all internal edges of T. If T is a model tree and T' is a reconstructed tree, the *false positives* are the edges of the set $C(T') - C(T)$ and the *false negatives* are those of the set $C(T) - C(T')$.

- The *false positive rate (FP)* is $(|C(T') - C(T)|)/(n-3)$.
- The *false negative rate (FN)* is $(|C(T) - C(T')|)(n-3)$.

Since $n-3$ is the number of internal edges of an unrooted binary tree on n leaves, the false positive and false negative rates are values in the range $[0, 1]$. The RF distance between T and T' is simply the average of these two rates, $\frac{\text{FN}+\text{FP}}{2}$.

3 Phylogenetic Networks

3.1 Hybridization and gene transfer

Two of the mechanisms that can result in non-tree evolution are *hybridization* and *horizontal gene transfer*.

- In hybridization, two lineages recombine to create a new species, as symbolized in Figure 1. The new species may have the same number of chromosomes as

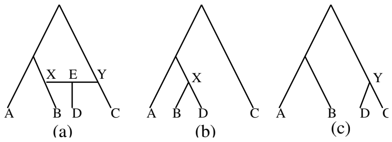


Figure 1: Hybridization: the network and its two induced trees

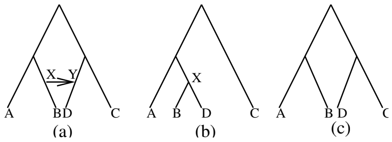


Figure 2: Horizontal transfer: the network and its two induced trees

its parent (*diploid hybridization*) or the sum of the numbers of chromosomes of its parents (*polyploid hybridization*).

- In horizontal gene transfer, genetic material is transferred from one lineage to another, producing a new lineage, as symbolized in Figure 2.

In these two cases, the true evolutionary history is best represented by a network, or *directed acyclic graph*, rather than by a tree.

Consider how an individual site evolves down a network. For diploid organisms, each chromosome consists of a pair of homologs. In a diploid hybridization event, the hybrid inherits one of the two homologs for each chromosome from each of its two parents. Since homologs assort at random into the gametes (sex cells), each has an equal probability of ending up in the hybrid. In polyploid hybridization, both homologs from both parents are contributed to the hybrid. Prior to hybridization, each site on the homolog has evolved in a tree-like fashion, although due to meiotic recombination (exchanges between the parental homologs during gamete production), different strings of sites may have different histories. Thus each site in the homologs of the parents of the hybrid evolved in a tree-like fashion on one of the trees contained inside (or, induced by) the network representing the hybridization event; see Figures 1(b) and 1(c). Similarly, in an evolutionary scenario involving horizontal transfer, certain sites are inherited through horizontal transfer from another species, as in Figure 2(b), while all others are inherited from the parent, as in Figure 2(c). Thus, in each of these two scenarios, *each site evolves down one of the trees induced by the network*.

3.2 Representation

Phylogenetic networks can be represented by *rooted directed acyclic graphs*, where each node (except for the root) has indegree 1 or 2. Nodes of indegree 1 are called *tree nodes*, whereas nodes of indegree 2 are called *hybrid nodes*. A hybrid node typically takes its genetic material from both of its parents, whereas a tree node takes its genetic material from its sole parent. The leaves of a network represent the extant taxa, and the internal nodes represent the hypothetical ancestral taxa. Whereas phylogenetic trees have a standard representation, the Newick format (a form of preorder traversal), no such representation exists for phylogenetic networks. We thus simply represent a network as a list of its edges, where each edge is defined by its two endpoints and its weight (the expected number of changes along that edge).

3.3 Model networks

We propose a model for generating random networks based on birth-death model for trees. A birth event in trees represents regular speciation; in networks, a birth event can be either regular speciation or hybrid speciation. We describe first the general model, then the restricted model used in our experiments.

Our model incorporates three types of events: regular speciation, hybrid speciation, and extinction. (Lateral gene transfers can easily be added as well.) Hybrid speciation events include diploid, allo-tetraploid, allo-hexaploid, allo-octaploid, and auto-tetraploid hybridization. To generate a random network, we start with one node (the root) and a pair of sequences (the two homologs of the chromosome at the root) and initiate a regular speciation event, thus creating two lineages.

Each lineage is represented by an edge in the network. An edge e is defined by its two endpoint nodes u and v. Associated with each node u is a time-stamp, $t(u)$, and associated with each edge e is a positive real number, $w(e)$, indicating the expected number of changes along that edge. Whenever a new lineage is created at node u, the sequences at node u evolve along the newly created lineage, according to a given model of evolution and to the $w(e)$ value associated with that lineage. (The model can be easily extended to support nonuniform rates among segments of the sequences by associating with each edge a collection of values, one for each segment.)

At any time t, we consider all of the lineages that exist at that time. For each such lineage l, and based on certain probabilities, either nothing happens, which means the lineage l is continued, or one of three mutually exclusive events occurs:

Extinction: Lineage l becomes extinct and a leaf u is created with stamp t.
Speciation: A node u is created with stamp t and two new lineages are started from u.
Hybridization: Let H be the set of all lineages at time t and choose a lineage $l' \in H$ to hybridize with l. (The choice of l' depends on the ploidy level and number of chromosomes of l', as well as on the evolutionary distance between l and l'.) When l and l' hybridize, the two lineages are continued and a new, third lineage arises from the hybrid speciation event. We allow each lineage to hybridize only once at each point in time.

This process may generate a network with fewer than the desired number of leaves, since all lineages might go extinct before enough lineages are created; in such a case, we can repeat the process until we obtain a network of the desired size or we can conduct longitudinal studies in a population of networks of diverse sizes.

In our general model, the topology of the network and the sequences at each node are generated together, in an interdependent manner. Such dependence is important in networks, since the probability of a particular hybridization's taking place depends, among other things, on the actual genomes of the putative hybridization parents. In the generation of trees, in contrast, simulation studies generally generate a tree topology first, then use it to derive many different collections of sequences—the topology does not depend on the sequences. Our restricted model follows the outline of our general model, but assumes the same independence used in tree simulations and thus begins by generating a network and then evolves the sequences down the completed network. To generate a random network N with one hybrid, we start with a random birth-death tree T, with times associated with its nodes. Let the time at the root be 0 and that at the last generated leaf be t_l; let t_i be a uniform variate in the interval $[0, t_l]$. We identify all nodes in the tree with age not exceeding t_i, but whose children have ages exceeding t_i: these nodes are the potential parents of the hybrid. We calculate the pairwise evolutionary distances (the length of the tree paths) among all of these nodes and choose the two parents with a probability inversely proportional to the evolutionary distance between the two nodes. Hybridization thus occurs between nodes that coexist in time and is more likely between more closely related ancestral taxa. To generate p hybrid nodes, we repeat the process p times, thus allowing hybrids to hybridize again. The networks thus generated are ultrametric; for each edge, we use a uniform variate x in the range $[-\ln c, \ln c]$ and multiply the edge length by e^x to deviate the networks from ultrametricity. (We call c the *deviation factor*.)

3.4 Simulating sequence evolution on networks

We based our sequence generation on the popular Seq-Gen[10] tool, which takes a tree in the Newick format and simulates the evolution of sequences along that tree under a choice of models of evolution. Seq-Gen simulates the evolution of sequences on a tree by placing a random sequence of the desired length at the root of the tree and then evolving it down the tree using the specified model of evolution and other parameters, including scaling factor, codon-specific corrections, gamma rate heterogeneity, etc.

Our modified version, Seq-Gen2, takes a network as an input; it assumes diploid hybridization and uses networks produced by our restricted model. The main change is that Seq-Gen2 evolves a *pair* of sequences, A and B, corresponding to the two homologs of the chromosome. The two sequences A and B are evolved independently down the tree, using the specified model of evolution and the other parameters. In the case of tree nodes, both the A and B sequences are evolved in exactly the same manner as they evolve on a tree. In the case of a diploid hybrid node, the node inherits the A or B sequence from one parent and the A or B sequence from the other parent. There is no evolution on the edges between the parents of the hybrid and the node at the origin of the hybrid, since, at the scale of evolution, hybridization is essentially instantaneous. The output of Seq-Gen2 is a set of pairs of sequences at the leaves.

4 Measuring the Distance Between Two Phylogenetic Networks

We want to measure the error between a model network N_1 and an inferred network N_2; this measure, $m(N_1, N_2)$, must be nonnegative and symmetric and satisfy two properties:

- N_1 and N_2 are the same network exactly when $m(N_1, N_2) = 0$.
- If N_1 and N_2 are trees, then $m(N_1, N_2)$ reduces to the RF measure.

These are very mild requirements; in particular, they do not define a metric, since we have not included any form of triangle inequality.

We propose two measures. Our first measure, based on a graph model, seeks to extend the RF measure by viewing the networks as extensions of trees. (Removing a chosen subset of edges from the network leaves a tree; the number of such subsets, however, can be very large.) This approach leads to a sophisticated view of the relationship between two networks, but the resulting measure is too expensive to compute exactly and tends to overemphasize the importance of guessing just the right number of non-tree events. Our second measure seeks to extend the RF measure by viewing the networks in terms of partitions. Just as the RF measure counts the number of compatible bipartitions, our new measure counts the number of compatible tripartitions. This second measure is easy to define and compute, is very closely related to the RF measure, and can also support a weighting scheme.

Let N_1 have p hybrid nodes and N_2 have q hybrid nodes. Then N_1 induces a set $\mathcal{T}_1$ of at most 2^p trees and N_2 induces a set $\mathcal{T}_2$ of at most 2^q trees. Figure 1 shows a network with one hybrid node and its two induced trees. Define the complete bipartite graph $G^{N_1,N_2} = (\mathcal{T}_1 \cup \mathcal{T}_2, E)$ (which has an edge $e = (u, v)$ between every $u \in \mathcal{T}_1$ and $v \in \mathcal{T}_2$) and assign to each edge $e = \{u, v\}$ a weight $w(e)$, the RF value between the two trees that correspond to nodes u and v. Our first measure can now be defined.

Definition 1 *The error rate between N_1 and N_2 is the weight of the minimum-weight edge-cover of G^{N_1,N_2}.*

The minimum-weight edge cover of a graph $G = (V, E)$ is a subset $E' \subseteq E$ such that E' covers V and the sum of edge weights, $\sum_{w(e)\in E'} w(e)$, is minimum. This measure clearly satisfies our two requirements. Although the minimum-weight edge-cover problem is solvable in polynomial time, the size of the bipartite graph is exponential in the number of hybrid nodes, so that computing the error rate may require exponential time. In practice, because hybridization is a rare event, the true network will have relatively few hybrid nodes, so that good reconstructions can be evaluated quickly. More damaging is the fact that this measure places a very strong emphasis on reconstructing networks that have the right number of hybridization events—small errors in that number dominate even very large errors in the choice of other edges.

This shortcoming leads us to design a measure based directly on partitions. Let N be a network, leaf-labeled by a set of taxa S, and e be an edge in N, where u is the source of e, and v is the target of e (i.e., u is a parent of v). Each edge $e \in N$ induces a tripartition of S, defined by the sets

- $A(e) = \{s \in S : s$ is reachable from the root of N only via $v\}$.
- $B(e) = \{s \in S : s$ is reachable from the root of N via at least one path passing through v and one path not passing through $v\}$.
- $C(e) = \{s \in S : s$ is not reachable from the root of N via $v\}$.

For each edge e, the three sets $A(e)$, $B(e)$ and $C(e)$ are weighted; the weight of an element in $A(e)$ or $C(e)$ is 0 (or any fixed constant) and the weight of an element $s \in B(e)$ is the maximum number of hybrid nodes on a path from v to s, where v is the target of edge e. Two weighted sets S_1 and S_2 are interchangeable, denoted by $S_1 \equiv S_2$, whenever they contain the same elements and each element has the same weight in both sets. Two edges e_1 and e_2 are *compatible*, denoted by $e_1 \equiv e_2$, whenever we have $A(e_1) \equiv A(e_2)$ and $B(e_1) \equiv B(e_2)$ and $C(e_1) \equiv C(e_2)$.

We define the *false negative rate* (FN) and *false positive rate* (FP) between two networks N_1 and N_2 as follows.

- $FN(N_1, N_2) = |\{e_1 \in N_1 : \nexists e_2 \in N_2 \; s.t. \; e_1 \equiv e_2\}|/|N_1|$.
- $FP(N_1, N_2) = |\{e_2 \in N_2 : \nexists e_1 \in N_1 \; s.t. \; e_1 \equiv e_2\}|/|N_2|$.

Definition 2 *The error rate between N_1 and N_2 is the average of FN(N_1, N_2) and FP(N_1, N_2).*
This measure clearly satisfies our two conditions and is computable in time polynomial in the size of the two networks.

5 Experimental Settings

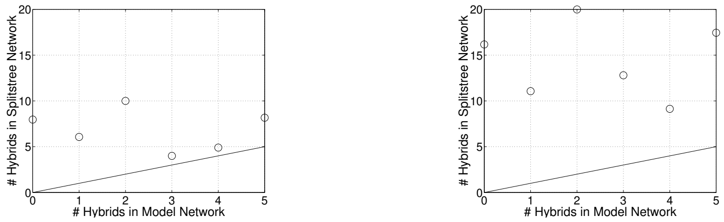
In order to obtain statistically robust results[11,12], we used 30 *runs*, each composed of a number of *trials* (a trial is a single comparison), computed a mean outcome of each run, and studied the mean and standard deviation of these runs. The standard deviation of the mean outcomes in our studies was generally negligible for NJ and MP (except for very small numbers of taxa), and only rarely larger for `SplitsTree`. We graphed the average of the mean outcomes for the runs, but omitted the standard deviation from the figures for clarity.

We ran our studies on random networks generated using the technique described earlier. These networks had diameter 2; in order to obtain networks with other diameters, we scaled the edge lengths by factors of 0.01, 0.05, 0.1, 0.5, 1, and 2, producing networks with diameters of 0.02, 0.1, 0.2, 1, 2, and 4, respectively. To deviate the networks from ultrametricity, we used a deviation factor of 4. We generated networks with 0, 1, 2, 3, 4, and 5 hybrids, for 10, 20, 40 and 80 leaves—one network for each combination of diameter, number of taxa, and number of hybrids. We then evolved sequences on these networks using the K2P+Gamma[13] model of evolution (we chose $\alpha = 1$ for the shape parameter and set the transition/transversion ratio to 2). We used a fixed factor[14] of 1 for distance correction and used sequence lengths of 25, 50, 100, 250, and 500. We used our Seq-Gen2 to evolve DNA sequences down the network under the K2P+Gamma model of evolution. The datasets we generated consisted of pairs of sequences, as described; however, the phylogenetic methods under study take datasets with only a single sequence per taxon. Therefore, from each sequence dataset that we generated, we created two sets: one that consisted of the A sequence of each taxon, and one that consisted of concatenation of the A and B sequences of each taxon. Thus, the effective sequence lengths that we looked at were 25, 50, 100, 250, and 500, when the A sequences were used, and 50, 100, 200, 500, and 1000, when the concatenation of A and B was used. We used PAUP*[15] for the greedy MP method and the `SplitsTree` software package[1].

6 Results and Discussion

We present two sets of results: one set compares the output of `SplitsTree` with the true network in terms of the number of hybrid nodes created, while the second compares the topological error rate of the three methods. Our first finding is that `SplitsTree` tends to reconstruct more hybrid nodes than are present in the true network, as illustrated in Figure 3 (the correct answer is on the oblique line). The effect decreases as the number of true hybrids increases—indeed, the worst-case instances for `SplitsTree` are trees.

Our second set of figures compares the FN and FP error rates (as defined in Section 4) of `SplitsTree`, NJ, and MP as a function of the number of true hybrids (Figure 4), the sequence length (Figure 5), the number of taxa (Figure 6), and the scaling factor (Figure 7). The two tree reconstruction methods are nearly indistinguishable throughout our experiments (with a slight advantage of MP over NJ in some



(a) single sequence (length 500) *(b) concatenated sequences (length 1000)*

Figure 3: The number of hybrids in the output of `SplitsTree` vs. the true number of hybrids, for 80 taxa at a scaling of 0.5.

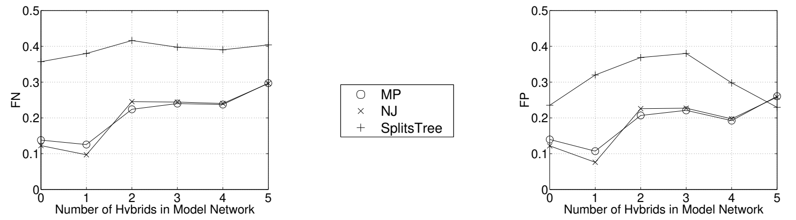


Figure 4: The FN and FP error rates of the three methods as a function of the number of hybrids in the model network. Scaling=0.5, concatenated sequences, 40 taxa, and sequence length=1000.

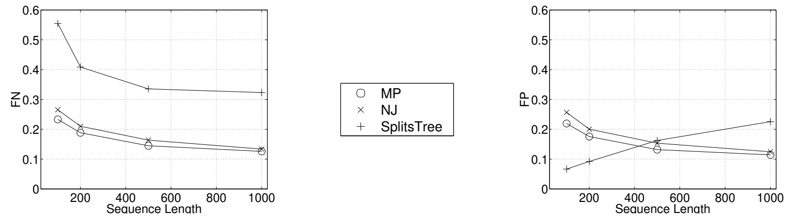


Figure 5: The FN and FP error rates of the three methods as a function of the sequence length. Scaling=0.5, concatenated sequences, 80 taxa, and 1 hybrid in the model networks.

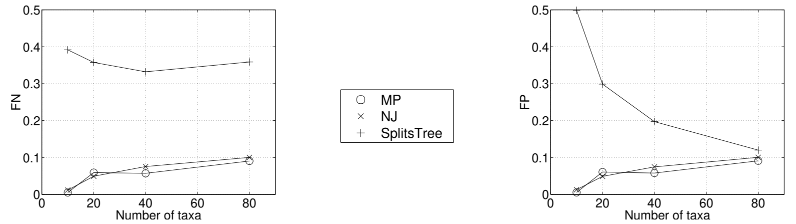


Figure 6: The FN and FP error rates of the three methods as a function of the number of taxa. Scaling=0.1, concatenated sequences, sequence length=1000, and no hybrids in the model networks.

cases). As expected from the results of many previous experimental investigations, their error rate grows slowly with the number of taxa, decreases slowly with increasing sequence length, and grows very slowly with increasing scaling factor. Since these methods never create hybrid nodes, their error rate as a function of the number of true hybrids must grow linearly in p, where p is the number of true hybrids, a growth clearly demonstrated in Figure 4.

The accuracy of `SplitsTree` suffers from its overestimates of the number of hybrid nodes: every hybrid not present in the true network affects all edges from which the hybrid is reachable. Thus Figure 4 confirms our findings from Figure 3: as the number of true hybrid nodes increases, `SplitsTree` infers a more accurate

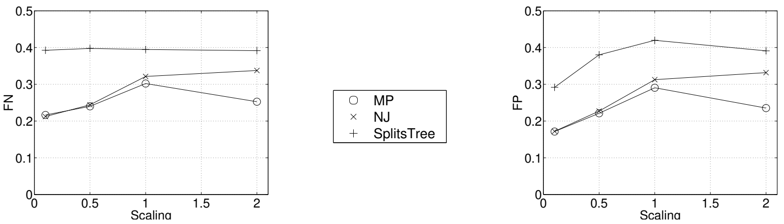


Figure 7: The FN and FP error rates of the three methods as a function of the scaling. Concatenated sequences, sequence length=1000, 40 taxa, and 3 hybrids in the model networks.

number of such nodes and its error rate does not noticeably increase with the number of hybrids. (Note that it is the FP rate that varies; the FN rate appears unaffected by the number of hybrids.) So far, our measure of topological accuracy behaves as desired. However, its behavior for `SplitsTree` in Figures 5 and 6 is, at first sight, surprising: one might expect the same trends as for the two tree reconstruction algorithms. Once again, however, this behavior accurately reflects the characteristics of the reconstructions. The mild increase of the FP rate in Figure 5 is due to the increasing number of incompatible splits (the decision criterion used by `SplitsTree`) due to an increasing number of characters. The sharp decrease in the FP rate in Figure 6 is due to the normalization by the increased number of taxa and the fact that the error is mostly attributable to excess hybrid nodes, whose number does not strongly depend on the number of taxa.

Overall, our proposed (second) measure of topological accuracy gives results that follow qualitative expectations, neither over- nor underemphasizing the importance of hybrid events.

7 Conclusions

We have presented a suite of tools that forms the basis of a simulation flow for the study of network reconstruction methods. The paucity of tools in this area, coupled with the universal recognition that tree models are too limited in many areas of the Tree of Life, makes the development of measures, algorithms, and tools an urgent task in phylogenetic research. While our test suite (generators and measures) is tailored for network reconstruction, our results at this point do not allow us to conclude much about network reconstruction methods—we could only test one existing method (others we tried would not run properly) and our range of experimental data in this study was somewhat limited. Yet, our test suite shows that it is possible to devise and use simulations for the assessment of reconstruction methods for phylogenetic networks. Our error measure has the advantage that it does not handle independently tree errors and network characteristics, avoiding the pitfall of having to assign relative weights or priorities to the two; if it does appear to favor trees slightly over networks, that is but a reflection of Occam's razor: hybridization events should be used only sparingly to explain data features otherwise explainable under a tree model.

8 Acknowledgments

This work is supported by National Science Foundation grants ACI 00-81404 (Moret), DEB 01-20709 (Linder, Moret, and Warnow), EIA 01-13095 (Moret), EIA 01-13654 (Warnow), EIA 01-21377 (Moret), and EIA 01-21680 (Linder and Warnow), and by the David and Lucile Packard Foundation (Warnow).

1. D.H. Huson. SplitsTree: A program for analyzing and visualizing evolutionary data. *Bioinformatics*, 14(1):68–73, 1998.
2. D.L. Swofford, G.J. Olsen, P.J. Waddell, and D.M. Hillis. Phylogenetic inference. In D. M. Hillis, B. K. Mable, and C. Moritz, editors, *Molecular Systematics*, pages 407–514. Sinauer Assoc., 1996.
3. G.L. Yule. A mathematical theory of evolution based on the conclusions of Dr. J.C. Willis, FRS. *Phil. Trans. Royal Society*, 213:21–87, 1924.
4. E.F. Harding. The probabilities of rooted tree shapes generated by random bifurcation. *Adv. Appl. Prob.*, 3:44–77, 1971.
5. N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406–425, 1987.
6. W.M. Fitch. On the problem of discovering the most parsimonious tree. *Am. Nat.*, 111:223–257, 1977.
7. O. Bininda-Emonds, S. Brady, J. Kim, and M. Sanderson. Scaling of accuracy in extremely large phylogenetic trees. In *Proc. 6th Pacific Symposium on Biocomputing (PSB01)*, pages 547–557. World Scientific, 2001.
8. L. Nakhleh, B.M.E. Moret, U. Roshan, K. St. John, J. Sun, and T. Warnow. The accuracy of fast phylogenetic methods for large datasets. In *Proc. 7th Pacific Symposium on Biocomputing (PSB02)*, pages 211–222. World Scientific, 2002.
9. D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.
10. A. Rambaut and N.C. Grassly. Seq-gen: An application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comp. Appl. Biosci.*, 13:235–238, 1997.
11. C. McGeoch. Analyzing algorithms by simulation: Variance reduction techniques and simulation speedups. *ACM Comp. Surveys*, 24:195–212, 1992.
12. B.M.E. Moret and H.D. Shapiro. Algorithms and experiments: The new (and the old) methodology. *J. Univ. Comp. Sci.*, 7(5):434–446, 2001.
13. M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, 16:111–120, 1980.
14. D. Huson, K.A. Smith, and T. Warnow. Correcting large distances for phylogenetic reconstruction. In *Proc. 3rd Workshop on Algorithms Engineering (WAE99)*, volume 1668 of *Lecture Notes in Computer Science*, pages 271–285. Springer-Verlag, 1999.
15. D.L. Swofford. PAUP*: Phylogenetic analysis using parsimony (and other methods), 1996. Sinauer Associates, Underland, Massachusetts, Version 4.0.