A New Linear-time Heuristic Algorithm for Computing the Parsimony Score of Phylogenetic Networks: Theoretical Bounds and Empirical Performance[⋆]

Guohua Jin[1], Luay Nakhleh[1], Sagi Snir[2], and Tamir Tuller[3]

[1] Department of Computer Science, Rice University, Houston, TX 77005, USA,
{jin,nakhleh}@cs.rice.edu
[2] Department of Mathematics, University of California, Berkeley, CA 94720, USA,
ssagi@math.berkeley.edu
[3] School of Computer Science, Tel Aviv University, Tel Aviv, Israel,
tamirtul@post.tau.ac.il

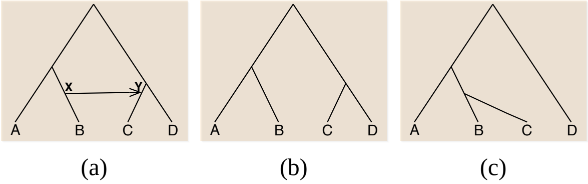
Abstract. Phylogenies play a major role in representing the interrelationships among biological entities. Many methods for reconstructing and studying such phylogenies have been proposed, almost all of which assume that the underlying history of a given set of species can be represented by a binary tree. Although many biological processes can be effectively modeled and summarized in this fashion, others cannot: recombination, hybrid speciation, and horizontal gene transfer result in *networks*, rather than trees, of relationships.

In a series of papers, we have extended the maximum parsimony (MP) criterion to phylogenetic networks, demonstrated its appropriateness, and established the intractability of the problem of scoring the parsimony of a phylogenetic network. In this work we show the hardness of approximation for the general case of the problem, devise a very fast (linear-time) heuristic algorithm for it, and implement it on simulated as well as biological data.

1 Introduction

Phylogenetic networks are a special class of *directed acyclic graphs* (DAGs) that models evolutionary histories when trees are inappropriate, such as in the cases of horizontal gene transfer (HGT) and hybrid speciation [26, 30, 27]. Fig. 1(a) illustrates a phylogenetic network on four species with a single HGT event. In horizontal gene transfer (HGT), genetic material is transferred from one lineage to another, as in Fig. 1(a). In an evolutionary scenario involving horizontal transfer, certain sites (specified by a specific substring within the DNA sequence of the species into which the horizontally transferred DNA was inserted) are inherited through horizontal transfer from another species (as in Figure 1(c)), while all others are inherited from the parent (as in Figure 1(b)). Thus, *each site evolves down one of the trees induced by (or, contained in) the network*. Similar scenarios arise in the cases of other reticulate evolution events (such as hybrid speciation and interspecific recombination).

[⋆] The authors appear in alphabetical order.



(a) (b) (c)

Fig. 1. (a) A phylogenetic network with a single HGT event from X to Y. (b) The underlying organismal (species) tree. (c) The tree of a horizontally transferred gene.

HGT plays a major role in bacterial genome diversification (e.g., see [7, 8, 19, 20]), and is a significant mechanism by which bacteria develop resistance to antibiotics (e.g., see [9]). Therefore, in order to reconstruct and analyze evolutionary histories of these groups of species, as well as to reconstruct the prokaryotic branch of the Tree of Life, developing accurate criteria for reconstructing and evaluating phylogenetic networks and efficient algorithms for inference based on these criteria is imperative. A large number of publications have been introduced in recent years about various aspects of phylogenetic networks; e.g., see [12, 30, 32, 11, 17, 18, 1, 31] for a sample of such papers in the last two years, and [26, 27] for detailed surveys.

In this work, we consider the *maximum parsimony* (MP) criterion, which has been in wide use for phylogenetic tree inference and evaluation. Roughly speaking, inference based on this criterion seeks the tree that minimizes the amount of evolution (in terms of number of mutations). In 1990, Jotun Hein proposed using this criterion for inferring the evolution of sequences subject to recombination. Recently, Nakhleh *et. al.* formulated the parsimony criterion for evaluating and inferring general phylogenetic networks [31], and we have recently demonstrated its appropriateness on both simulated and biological datasets [21, 22]. Applying the parsimony criterion for phylogenetic networks involves solving the *big* and the *small* parsimony problems, referred to as the **FTMPPN** and **PSPN** problems, respectively, in [31]. In [21] the small problem (scoring the parsimony of a given network) was proved to be NP-hard and a heuristic algorithm was devised. A recent work by Nguyen *et. al.* [33] provided a hardness result for a related, yet different, version of the small parsimony problem.

In this paper we devise a very fast (linear-time) heuristic algorithm, with very good empirical performance, for the PSPN problem. Further, we show that for a restricted, yet realistic, class of phylogenetic networks, our algorithm gives a polynomial time 3-approximation for the problem. Moreover, we show that although the theoretical approximation ratio is not very promising, the algorithm does give very good results in practice compared to the exact algorithm.

2 Parsimony of Phylogenetic Networks

Preliminaries and Definitions Let $T = (V, E)$ be a tree, where V and E are the *tree nodes* and *tree edges*, respectively, and let $\mathcal{L}(T)$ denote its leaf set. Further, let $\mathcal{X}$ be a set of taxa (species). Then, T is a phylogenetic tree over $\mathcal{X}$ if there is a bijection between $\mathcal{X}$ and $\mathcal{L}(T)$. Henceforth, we will identify the taxa set with the leaves they are mapped to, and let $[n] = \{1, .., n\}$ denote the set of leaf-labels. A tree T is said to be *rooted* if the set of edges E is directed and there is a single distinguished internal vertex r with in-degree 0. We denote by T_v the subtree rooted at v induced by the tree edges. A function $\lambda : [n] \rightarrow \{0, 1, .., \Sigma - 1\}$ is called a *state assignment function* over the alphabet Σ for T. We say that function $\hat{\lambda} : V(T) \rightarrow \{0, 1, .., \Sigma - 1\}$ is an extension of λ on T if it agrees with λ on the leaves of T. In a similar way, we define a function $\lambda^k : [n] \longmapsto \{0, 1, .., \Sigma - 1\}^k$ (in applications of the methodology, k corresponds to the sequence length) and an extension $\hat{\lambda}^k : V(T) \longmapsto \{0, 1, .., \Sigma - 1\}^k$. The latter function is called a *labeling* of T. We write $\hat{\lambda}^k(v) = s$ to denote that sequence s is the label of the vertex v. The ith *site* is an n-tuple where the jth coordinate is the state of the ith site of species (leaf) j.

Given a labeling $\hat{\lambda}^k$, let $d_e(\hat{\lambda}^k)$ denote the Hamming distance between the two sequences labeling the two endpoints of the edge $e \in E(T)$.

A phylogenetic network $N = N(T) = (V', E')$ over the taxa set $\mathcal{X}$ is derived from $T = (V, E)$ by adding a set H of edges to T, where each edge $h \in H$ is added as follows: (1) split an edge $e \in E$ by adding new node, v_e; (2) split an edge $e' \in E$ by adding new node, $v_{e'}$; (3) finally, add a directed *reticulation edge* from v_e to $v_{e'}$. It is important to note that the resulting network must be acyclic [30].

We extend the notion of T_v to networks as follows. For a network N and a node $v \in V(N)$, let N_v be the graph induced by all the nodes reachable from v. Finally, we denote by $\mathcal{T}(N)$ the set of all trees contained inside network N. Each such tree is obtained by the following two steps: (1) for each node of in-degree 2, remove one of the incoming edges, and then (2) for every node x of in-degree and out-degree 1, whose parent is u and child is v, remove node x and its two adjacent edges, and add a new edge from u to v.

Further, Phylogenetic networks must satisfy additional temporal constraints [30]. First, N should be acyclic (genetic material flows only forward in time). Second, N should satisfy additional temporal constraints, so as to reflect the biological fact that the donor and recipient of a horizontally transferred gene must co-exist in time. Since at the scale of evolution HGT events are instantaneous in time, a reticulation edge between two points dictates that they correspond to the same chronological time. This in turn implies that if x and y are the two endpoints of an HGT edge and their time-stamp is t, then there cannot be an HGT edge between a node z at time $t' < t$ and a node w at time $t'' > t$. Note that this condition is not guaranteed by the acyclicity condition[4]. See [30] for a formal description of the temporal constraints on phylogenetic networks.

2.1 Parsimony of Phylogenetic Networks

We begin by reviewing the parsimony criterion for phylogenetic trees.

Problem 1. Parsimony Score of Phylogenetic Trees (PSPT)

[4] It is important to note that, while acyclicity must be satisfied by all phylogenetic networks, the other temporal constraints may be violated, due to extinction or incomplete taxon sampling, for example.

Input: A 3-tuple (S, T, λ^k), where T is a phylogenetic tree and λ^k is the labeling of $\mathcal{L}(T)$ by the sequences in S.
Output: The extension $\hat{\lambda}^k$ that minimizes the expression $\sum_{e \in E(T)} d_e(\hat{\lambda}^k)$.

We define the parsimony score for (S, T, λ^k), $pars(S, T, \lambda^k)$, as the value of this sum, and $pars(S, T, \lambda^k, i)$ as the value of this sum for site i only. In other words, $pars(S, T, \lambda^k) = \sum_{1 \leq i \leq k} pars(S, T, \lambda^k, i)$. It is easy to see that the optimal value is obtained by optimal solutions for every site $1 \leq i \leq k$. Problem 1 has a polynomial time dynamic programming type algorithm originally devised by Fitch [10] and later extended by Sankoff [36]. The algorithm finds an optimal assignment (i.e., $\hat{\lambda}^k$) for each site separately.

Since Fitch's algorithm is a basic building block in this paper, we hereby describe it. As mentioned above, the input to the problem is a tree T and a single character $C = \lambda^1$. The algorithm finds the optimal assignment to internal nodes of T, in two phases: (1) assigning values to internal nodes in a bottom-up fashion, and (2) eliminating the values determined in the previous phase in a top-down fashion. Specifically, phase (1) proceeds as follows: for a node v with children v_1 and v_2 whose values $A(v_1)$ and $A(v_2)$ have been determined,

$$A(v) = \begin{cases} A(v_1) \cap A(v_2) & \text{if } A(v_1) \cap A(v_2) \neq \emptyset \\ A(v_1) \cup A(v_2) & \text{otherwise.} \end{cases}$$

Phase (2) proceeds as follows: for a node v whose parent $f(v)$ has already been processed:

$$B(v) = \begin{cases} \sigma \in A(v) \cap A(f(v)) & \text{if } A(v) \cap A(f(v)) \neq \emptyset \\ \sigma \in A(v) & \text{otherwise.} \end{cases}$$

The algorithm above applies only to binary trees. Nonetheless, a straightforward extension to arbitrary k-degree trees can be easily achieved. We now prove a lemma that will be useful later.

Lemma 1. *Let T be a tree and C a single character over the alphabet Σ. Let x be the number of internal nodes v s.t. $|A(v)| > 1$ by applying Fitch's algorithm on (T, C). Then x is less than twice S^*—the parsimony score of T over C.*

Proof. We prove the lemma by induction on l, the length of the path from root r to the closest leaf. Obviously, we are interested only in cases where $|A(r)| > 1$ in the first phase. For $l = 1$, T is a cherry[5] with two leaves v_1 and v_2 with $A(v_1) \cap A(v_2) = \emptyset$ and the lemma follows. Assume correctness for $l = k$ and we prove for $l = k + 1$. We divide the proof into two cases:

- $A(v_1) \cap A(v_2) = \emptyset$: There must be additional mutation from v and the lemma follows.

[5] A *cherry* is a rooted tree with three nodes: the root, and two leaves which are children of the root.

- $|A(v_1)| > 1$ and $|A(v_2)| > 1$: In this case there might be no mutation from v to either of his children (e.g. $A(v_1) = \{A, C, G\}$ and $A(v_2) = \{A, G\}$). Let x_1 and x_2 be the number of nodes w in T_{v_1} and in T_{v_2} resp. with $|A(w)| > 1$, and S_1^* and S_2^* the optimal scores for T_{v_1} and T_{v_2} resp. It is clear that $S^* = S_1^* + S_2^*$, however by the assumption, $x = x_1 + x_2 + 1 < x_1 + 1 + x_2 + 1 \leq 2(S_1^* + S_2^*)$ and the assumption follows.

Problem 1 was extended to phylogenetic networks in [14, 15, 31], and its quality as a criterion for reconstructing and evaluating networks was established on both synthetic and biological data in a series of papers [31, 21, 22]

Definition 1. *Parsimony Score of Phylogenetic Networks (PSPN)*

> **Input:** *A 3-tuple (S, N, λ^k), where N is a phylogenetic network and λ^k is the labeling of $\mathcal{L}(N)$ by the sequences in S.*
> **Output:** *The extension $\hat{\lambda}^k$ that minimizes the expression*
>
> $$\sum_{1 \leq i \leq k} \left[min_{T \in \mathcal{T}(N)} pars(S, T, \lambda^k, i) \right] .$$

3 Hardness of approximation of the PSPN problem

In [23], we proved that the PSPN problem is NP-hard by a reduction from the max-2-sat problem. By [13], there is a constant ζ such that there is no polynomial time algorithm for max-2-sat with performance ratio better than ζ, *i. e.* there are $P1$ and $P2$ such that $gap - max - 2sat[P1, P2]$[6] is NP-hard (see [16] for the definition of gap problems). Thus by the reduction in [23] there is a constant $\zeta^{'}$ such that there is no polynomial time algorithm for $PSPN$, and $gap - PSPN[4 * |C| - P2 + |U|, 4 * |C| - P1 + |U|]$ is NP-hard.

Corollary 1. *There is a constant ζ' such that there is no polynomial time algorithm for PSPN with performance ratio better than ζ'.*

Corollary 2. *The PSPN problem is hard to approximate even for networks of bounded degrees, where each node has at most 20 children.*

This result follows from the fact that the $gap - max - 3sat$ problem, when every variable appears 5 times, is hard.

It is important to note that our reduction in [23] generates networks with no more than *one* HGT between any pair of edges. Thus the hardness of approximation results hold also for such networks. In the next section we provide an approximation algorithm for a network with up to one [7] HGT between each pair of edges.

[6] In a $gap - max - 2sat[A, B]$ problem, where $A < B$, a YES-instance is a formula in which at least B clauses are satisfiable, and a NO-instance is a formula in which at most A clauses are satisfiable. If the number of satisfiable clauses is strictly greater than A and strictly smaller than B, then either answer (YES or NO) can be given.

[7] The algorithm can be generalized to the case where the number of HGTs between each pair of edges is bounded by some constant $c > 1$. This will increase the approximation ratio.

4 A Linear-time Algorithm

Our linear time algorithm builds on the improved heuristic of [21] for the PSPN problem, outlined in Fig. 2. The algorithm is based on the fact that there always exists a *lowest reticulation edge* in a phylogenetic network that satisfies the temporal constraints described in [30]. A reticulation edge $e = (u \rightarrow v)$ is called *a lowest reticulation edge* (or just a lowest edge) if there is no reticulation edge (other than e) adjacent to any node in either T_u or T_v.

```
ExactPSPN(N=(V',E'))

1. If $N$ is not a tree
   (a) Find a lowest reticulation edge $e = (u \rightarrow v)$ in $N$;
   (b) Let $e'$ be the edge between $v$ and its ancestral node on the tree edge;
   (c) By Fitch's algorithm, compute the optimal assignment $A$ of $u$ and $v$;
   (d) If $A(u) \cap A(v) = \emptyset$ then
       return $(V', E' \setminus e)$;
   (e) else if $A(u) \subseteq A(v)$ then
       return $(V', E' \setminus e')$;
   (f) else
       i. $opt = pars(ExactPSPN(V', E' \setminus e))$;
       ii. $A(u) \leftarrow A(v)$;        // update $v'$s values
           $opt' = pars(ExactPSPN(V', E' \setminus e'))$;
       iii. if $opt' < opt$ return $(V', E' \setminus e')$; else return $(V', E' \setminus e)$.
2. else return $Fitch(N)$.
```

Fig. 2. The improved heuristic algorithm.

The algorithm in Fig. 2 checks in each step a lowest reticulation edge of the network. It calculates $A(u)$ and $A(v)$ by Fitch's algorithm. In a case where $\neg((A(u) \bigcap A(v) = \emptyset) \vee (A(u) \subseteq A(v)))$ the algorithm considers recursively (and separately) both the reticulation edge and the (alternative) tree edge (i.e. the network with and the network without $(u \rightarrow v)$) . The running time of the algorithm is exponential with the number of such cases.

Our new linear-time algorithm is similar to the exact heuristic algorithm described in Fig. 2 in its recursive style and the search for a lowest reticulation edge at every invocation. However, in contrast, whenever we are unsure of a mutation along that edge, we just take it. Formally, we remove the exponential component from the exact algorithm $PSPN$ and perform step (1e) in any case the condition at step (1d) is not satisfied. The algorithm, Linear-PSPN(N), is outlined in Fig. 3.

Claim. Let $E(N)$ be the set of reticulation and tree edges in N. Then the algorithm terminates and runs in time $O(E(N))$.

4.1 A 3-Approximation Ratio

An algorithm A for a minimization problem P with optimal solution $opt(P)$ (or just opt for short), is a polynomial time α-approximation algorithm if A runs in polynomial

```
**Linear-PSPN**$(N = (V, E))$

1. If $N$ is not a tree
   (a) Find a lowest reticulation edge $e = (u \to v)$ in $N$;
   (b) Let $e'$ be the edge between $v$ and its ancestral node on the tree edge;
   (c) By Fitch's algorithm compute the optimal bottom up assignment $A$
       to $T_u$ and $T_v$, $A(T_u)$ and $A(T_v)$;
   (d) If $A(u) \cap A(v) = \emptyset$ then
         $\hat{\lambda}_1 =$ Linear-PSPN$(V, E \setminus e)$;
   (e) else
         $\hat{\lambda}_1 =$ Linear-PSPN$(V, E \setminus e')$;
   (f) return $\hat{\lambda}_1$.
2. Continue first phase of Fitch on the tree N without changing internal
   labels that have already determined.
3. Perform second phase of Fitch on the tree N.
4. Return the resultant fully labelled tree.
```

Fig. 3. The Linear-PSPN algorithm.

time and the score of the solution returned by A, $A(P)$, satisfies

$$A(P) \leq \alpha \cdot opt(P).$$

We now show that if the number of reticulation edges emanating from a tree edge is at most one, Linear-PSPN yields a 3-approximation algorithm. The analysis relies on Lemma 1 above.

The technique we use is based on the *local ratio* technique which is useful for approximating optimization covering problems such as vertex cover, dominating set, minimum spanning tree, feedback vertex set and more [4, 2, 3]. The technique recursively solves local sub-problems until a solution is found. The way the local sub-problems are solved determines the approximation ratio. In general, we decompose the network into two networks and show that two *separate* optimal solutions to the networks are a lower bound to an optimal solution to the complete network.

Theorem 1. *If the maximum number of reticulation edges emanating from a tree edge is* 1*, then the approximation ratio of* $Linear - PSPN$ *is 3.*

Proof. We start with a central observation to give a lower bound on the optimal score of a given network.

Observation 1 *Let* $e = (u \to v)$ *be a lowest reticulation edge in a network* N*. Let* $N' = N \setminus T_v$ *be the network obtained by pruning* T_v *from* N *(including the edges leading to* v*). Then* $opt(N) \geq opt(N') + opt(T_v)$*.*

Proof. Simply take the tree T with the assignment to internal nodes $A(T)$ yielding $opt(N)$ as an upper bound on $opt(N') + opt(T_v)$.

Corollary 3. *If we find an* α *approximation to both* $opt(N')$ *and* $opt(T_v)$*, we find an* α *approximation to* N*.*

We now show how the 3-ratio is obtained. At any local step, we remove a subtree that was solved optimally and contains no reticulation edges (or contains only such edges that did not incur a mutation). This subtree is connected to the rest of the network by a $(u \rightarrow v)$ reticulation edge with $A(v) \subset A(u)$. Let T_v be the tree removed from the rest of the network. Such a reticulation edge might incur an additional mutation. However, note that $|A(u)| > 1$. Now, since there is no reticulation edge entering T_u that can reduce the number of mutations, there exists an optimal solution with T_u as a subgraph. By Lemma 1 the number of mutations in T_u is at least half the number of nodes u' with $|A(u')| > 1$. By our assumption, every edge entering such a node u' gives rise to at most one extra mutation. We simply change that extra mutation on u' and the theorem follows. The rest of the network is solved recursively.

5 Experimental Results

We implemented the approximation algorithm and evaluated both its accuracy and execution time through experiments on both simulated and biological datasets. We performed experiments on a 2.4 GHz Intel Pentium 4 PC. Accuracy of the approximation algorithm was measured as the difference of the parsimony scores computed by the approximation algorithm and the exact algorithm normalized by the parsimony score computed by the exact algorithm, presented as percentage. Execution times of both the approximation algorithm and the exact algorithm were measured and speedups of the approximation algorithm over the exact algorithm were reported.

Simulated Datasets For the simulated datasets, we first used the `r8s` tool [35] to generate a random birth-death phylogenetic tree on 20 taxa. The `r8s` tool generates molecular clock trees; we deviated the tree from this hypothesis by multiplying each edge in the tree by a number randomly drawn from an exponential distribution. The resulting tree was taken as the species tree. The expected evolutionary diameter (longest path between any two leaves in the tree) was 0.2. A model phylogenetic network was generated by adding 5 HGT edges to the model tree.

Based on the model network, we used the Seq-gen tool [34] to evolve 26 datasets of DNA sequences of length 1500 down the “species” tree and DNA sequences of length 500 down the other tree contained inside the network (the one that exhibits all HGT events). Both sequence datasets were evolved under the K2P+γ model of evolution, with shape parameter 1 [25]. Finally, we concatenated the two datasets.

Biological Datasets We have included experimental results on three biological datasets we previously studied [22]. The first biological dataset is the rubisco gene *rbcL* of a group of 46 plastids, cyanobacteria, and proteobacteria, which was analyzed by Delwiche and Palmer [6]. This dataset consists of 46 aligned amino acid sequences (each of length 532), 40 of which are from Form I of rubisco and the other 6 are from Form II of rubisco. The first 21 and the last 14 sites of the sequence alignment were excluded from the analysis, as recommended by the authors. The species tree for the dataset was created based on information from the ribosomal database project (http://rdp.life.uiuc.edu) and the work of [6]. The second dataset consists of the ribosomal protein *rpl12e* of a group of 14 Archaeal organisms, which was analyzed by Matte-Tailliez *et al.* [28].

This dataset consists of 14 aligned amino acid sequences, each of length 89 sites. The authors constructed the species tree using Maximum Likelihood, once on the concatenation of 57 ribosomal proteins (7,175 sites), and another on the concatenation of SSU and LSU rRNA (3,933 sites). The two trees are identical, except for the resolution of the *Pyrococcus* three-species group; we used the tree based on the ribosomal proteins. The third dataset consists of the ribosomal protein gene *rps11* of a group of 47 flowering plants, which was analyzed by Bergthorsson *et al.* [5]. This data set consists of 47 aligned DNA sequences, each with 456 sites. The authors analyzed the 3' end of the sequences separately; this part of the sequences contains 237 sites. The species tree was reconstructed based on various sources, including the work of [29] and [24].

5.1 Results and Analysis

We evaluated the performance of the algorithms in terms of accuracy and speedup. Since the running time of the exact algorithm for computing the parsimony score of a phylogenetic network is affected by the number of trees that it considers inside the network, we also plotted the average numbers of trees that the exact algorithm considers, so that we understand the gains in speed for the approximation algorithm, which considers exactly one tree in all cases.

Fig. 4 shows the results of the 26 simulated datasets for networks with up to 6 HGT edges. The results were collected from 1000 sampled valid networks for each case of the multiple gene transfers. HGTs in each network are distributed differently. Overall, the approximation algorithm is very accurate with the statistical mean being about 1% different in the parsimony scores computed, compared with the exact algorithm. All parsimony scores computed by the approximation algorithm were within 3.5% of the optimal scores. For the networks with less then 5 HGTs, the approximation algorithm achieves about the same accuracy of the exact algorithm in most of the networks. The figure also shows that the approximation algorithm is up to 70% faster than the exact algorithm, with statistical mean around 32%. The improved execution time of the approximation algorithm came from the fewer number of trees created for computing parsimony score. Fig. 4 also shows the average number of trees that the exact algorithm considers. The average number of trees created increases as the number of HGTs increases. For networks with 6 HGTs (simulated dataset), the average number of trees can be up to 2.

For the rubisco gene *rbcL* dataset, We tested networks with up to 8 HGTs. In each case of the multiple gene transfers, we selected 500 valid networks with HGTs being placed differently. As the results in Fig. 4 show, the approximation algorithm is almost as accurate as the exact algorithm (within 0.5%; see the small boxes or the lower quartile for 7-HGT case at the bottom). Very few outliers exist across different numbers of HGTs. On the other hand, the approximation algorithm performs very efficiently. It performs up to a factor of 7 faster than the exact algorithm. The statistical mean of the improvement increases as the number of HGTs increases, with an exception in the case of 8 HGTs, where the sampled networks are probably not distributed well enough.

Similar trends are observed with the other two biological datasets, as shown in Fig. 4. The figures show that the statistical mean of the difference in accuracy is almost 0 in all cases, which indicates that the approximation algorithm computes almost

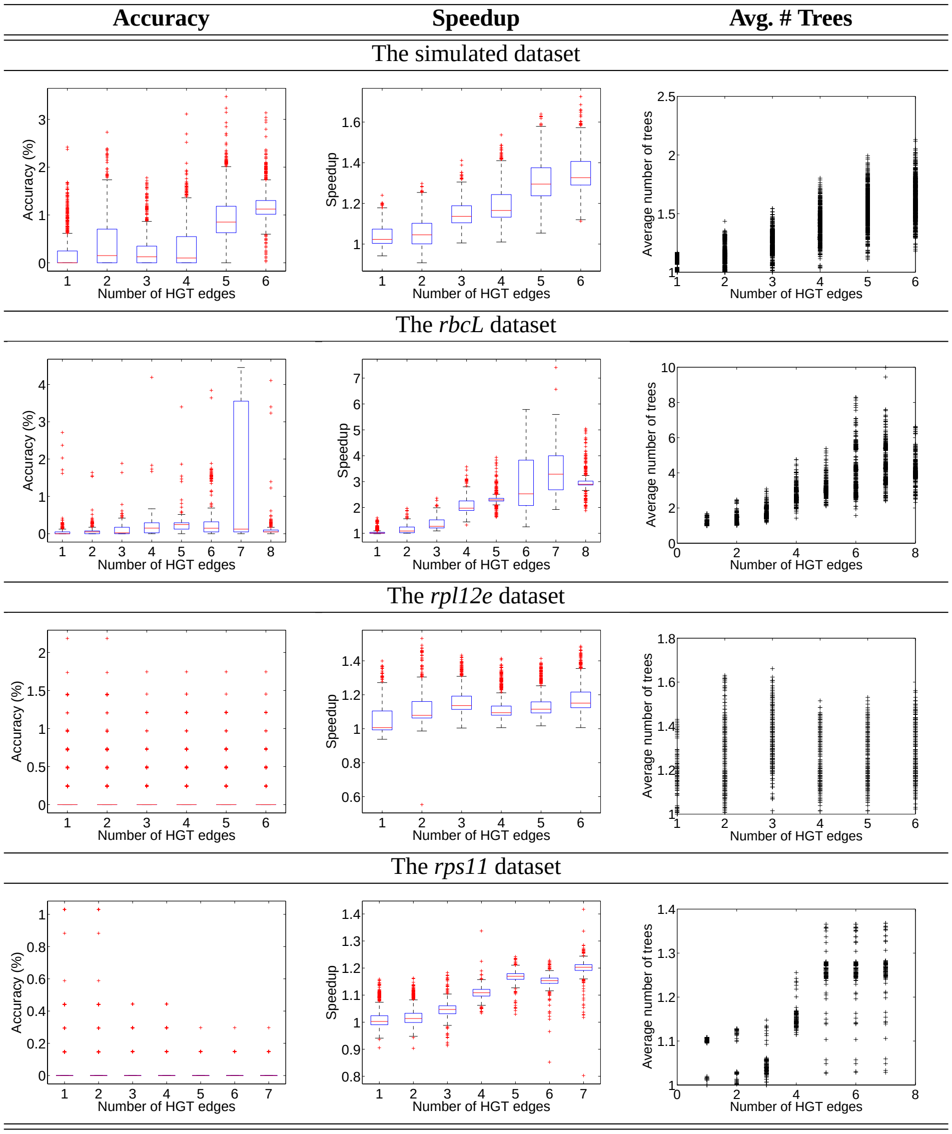


Fig. 4. Results for the four datasets. Accuracy is computed as$((MP_{approx} - MP_{exact})/MP_{exact})$, and shown as percentage. Speedup is computed as the execution time of the exact algorithm divided by the that of the approximation algorithm. The right column shows the average number of trees created for computing parsimony by the exact algorithm.

identical scores as the exact algorithm, in most cases. The speedup factors, and their correlations to the numbers of trees the exact algorithm considers, are also shown, and they show improvements up to a factor of 1.5. We expect that for larger datasets the gains in performance (speedup) will be even more pronounced. If one hopes to detect HGT events in large prokaryotic groups, for example, such a speedup is essential.

Acknowledgments

This work was supported in part by the Rice Terascale Cluster funded by NSF under grant EIA-0216467 and a partnership between Rice University, Intel, and HP. Luay Nakhleh was supported in part by the Department of Energy grant DE-FG02-06ER25734, the National Science Foundation grant CCF-0622037, and the George R. Brown School of Engineering Roy E. Campbell Faculty Development Award. Tamir Tuller was supported by the Edmond J. Safra Bioinformatics program at Tel Aviv University.

References

[1] V. Bafna and V. Bansal. Improved recombination lower bounds for haplotype data. In *Proceedings of the Ninth Annual International Conference on Computational Molecular Biology*, pages 569–584, 2005.

[2] V. Bafna, P. Berman, and T. Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM J. on Discrete Mathematics*, 12:289–297, 1999.

[3] R. Bar-Yehuda. One for the price of two: A unified approach for approximating covering problems. *Algorithmica*, 27:131–144, 2000.

[4] R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the weighted vertex cover problem. *Annals of Discrete Mathematics*, 25:27–46, 1985.

[5] U. Bergthorsson, K.L. Adams, B. Thomason, and J.D. Palmer. Widespread horizontal transfer of mitochondrial genes in flowering plants. *Nature*, 424:197–201, 2003.

[6] C. F. Delwiche and J. D. Palmer. Rampant horizontal transfer and duplication of rubisco genes in eubacteria and plastids. *Mol. Biol. Evol*, 13(6), 1996.

[7] W.F. Doolittle, Y. Boucher, C.L. Nesbo, C.J. Douady, J.O. Andersson, and A.J. Roger. How big is the iceberg of which organellar genes in nuclear genomes are but the tip? *Phil. Trans. R. Soc. Lond. B. Biol. Sci.*, 358:39–57, 2003.

[8] J.A. Eisen. Assessing evolutionary relationships among microbes from whole-genome analysis. *Curr. Opin. Microbiol.*, 3:475–480, 2000.

[9] I.T. Paulsen *et al.* Role of mobile DNA in the evolution of Vacomycin-resistant Enterococcus faecalis. *Science*, 299(5615):2071–2074, 2003.

[10] W. Fitch. Toward defining the course of evolution: minimum change for a specified tree topology. *Syst. Zool*, 20:406–416, 1971.

[11] D. Gusfield and V. Bansal. A fundamental decomposition theory for phylogenetic networks and incompatible characters. In *Proceedings of the Ninth Annual International Conference on Computational Molecular Biology*, pages 217–232, 2005.

[12] M. Hallett, J. Lagergren, and A. Tofigh. Simultaneous identification of duplications and lateral transfers. In *Proceedings of the Eighth Annual International Conference on Computational Molecular Biology*, pages 347–356, 2004.

[13] J. Hastad. Some optimal inapproximability results. *STOC97*, pages 1–10, 1997.

[14] J. Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, 98:185–200, 1990.

[15] J. Hein. A heuristic method to reconstruct the history of sequences subject to recombination. *Journal of Molecular Evolution*, 36:396–405, 1993.

[16] D. S. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1997.

[17] D.H. Huson, T. Klopper, P.J. Lockhart, and M. Steel. Reconstruction of reticulate networks from gene trees. In *Proceedings of the Ninth Annual International Conference on Computational Molecular Biology*, pages 233–249, 2005.

[18] T.N.D. Huynh, J. Jansson, N.B. Nguyen, and W.K. Sung. Constructing a smallest refining galled phylogenetic network. In *Proceedings of the Ninth Annual International Conference on Computational Molecular Biology*, pages 265–280, 2005.

[19] R. Jain, M.C. Rivera, J.E. Moore, and J.A. Lake. Horizontal gene transfer in microbial genome evolution. *Theoretical Population Biology*, 61(4):489–495, 2002.

[20] R. Jain, M.C. Rivera, J.E. Moore, and J.A. Lake. Horizontal gene transfer accelerates genome innovation and evolution. *Molecular Biology and Evolution*, 20(10):1598–1602, 2003.

[21] G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Efficient parsimony-based methods for phylogenetic network reconstruction. *Bioinformatics*, 23:e123–e128, 2006.

[22] G. Jin, L. Nakhleh, S. Snir, and T. Tuller. Inferring phylogenetic networks by the maximum parsimony criterion: A case study. *Molecular Biology and Evolution*, 24(1):324–337, 2007.

[23] G. Jin, L. Nakhleh, S. Snir, and T. Tuller. On approximating the parsimony score of phylogenetic networks. Under review, 2007.

[24] W.S. Judd and R.G. Olmstead. A survey of tricolpate (eudicot) phylogenetic relationships. *American Journal of Botany*, 91:1627–1644, 2004.

[25] M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16:111–120, 1980.

[26] C.R. Linder, B.M.E. Moret, L. Nakhleh, and T. Warnow. Network (reticulate) evolution: biology, models, and algorithms. In *The Ninth Pacific Symposium on Biocomputing (PSB)*, 2004. A tutorial.

[27] V. Makarenkov, D. Kevorkov, and P. Legendre. Phylogenetic network reconstruction approaches. *Applied Mycology and Biotechnology (Genes, Genomics and Bioinformatics)*, 6, 2005. To appear.

[28] O. Matte-Tailliez, C. Brochier, P. Forterre, and H. Philippe. Archaeal phylogeny based on ribosomal proteins. *Molecular Biology and Evolution*, 19(5):631–639, 2002.

[29] F.A. Michelangeli, J.I. Davis, and D.Wm. Stevenson. Phylogenetic relationships among Poaceae and related families as inferred from morphology, inversions in the plastid genome, and sequence data from mitochondrial and plastid genomes. *American Journal of Botany*, 90:93–106, 2003.

[30] B.M.E. Moret, L. Nakhleh, T. Warnow, C.R. Linder, A. Tholse, A. Padolina, J. Sun, and R. Timme. Phylogenetic networks: modeling, reconstructibility, and accuracy. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):13–23, 2004.

[31] L. Nakhleh, G. Jin, F. Zhao, and J. Mellor-Crummey. Reconstructing phylogenetic networks using maximum parsimony. *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB2005)*, pages 93–102, August 2005.

[32] L. Nakhleh, T. Warnow, and C.R. Linder. Reconstructing reticulate evolution in species: theory and practice. In *Proceedings of the Eighth Annual International Conference on Computational Molecular Biology*, pages 337–346, 2004.

[33] C. T. Nguyen, N. B. Nguyen, W. K. Sung, and L Zhang. Reconstructing recombination network from sequence data: The small parsimony problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 2006.

[34] 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.

[35] M. Sanderson. `r8s` software package. Available from http://loco.ucdavis.edu/r8s/r8s.html.

[36] D. Sankoff. Minimal mutation trees of sequences. *SIAM Journal on Applied Mathematics*, 28:35–42, 1975.