MURPAR: A Fast Heuristic for Inferring Parsimonious Phylogenetic Networks from Multiple Gene Trees

Hyun Jung Park and Luay Nakhleh

Dept. of Computer Science, Rice University, Houston, TX 77005
{hp6,nakhleh}@cs.rice.edu

Abstract. Phylogenetic networks provide a graphical representation of evolutionary histories that involve non-treelike evolutionary events, such as horizontal gene transfer (HGT). One approach for inferring phylogenetic networks is based on reconciling gene trees, assuming all incongruence among the gene trees is due to HGT. Several mathematical results and algorithms, both exact and heuristic, have been introduced to construct and analyze phylogenetic networks. Here, we address the computational problem of inferring phylogenetic networks with minimum reticulations from a collection of gene trees. As this problem is known to be NP-hard even for a pair of gene trees, the problem at hand is very hard. In this paper, we present an efficient heuristic, MURPAR, for inferring a phylogenetic network from a collection of gene trees by using pairwise reconciliations of trees in the collection. Given the development of efficient and accurate methods for pairwise gene tree reconciliations, MURPAR inherits this efficiency and accuracy. Further, the method includes a formulation for combining pairwise reconciliations that is naturally amenable to an efficient *integer linear programming* (ILP) solution. We show that MURPAR produces more accurate results than other methods and is at least as fast, when run on synthetic and biological data. We believe that our method is especially important for rapidly obtaining estimates of genome-scale evolutionary histories that can be further refined by more detailed and compute-intensive methods.

1 Introduction

One aspect of phylogenomic studies entails the reconstruction of evolutionary trees for different genomic regions and combining those trees to obtain an evolutionary history of the genomes under study. When events such as horizontal gene transfer (HGT) occur, the evolutionary history of the genomes is best represented by a *phylogenetic network*, which accounts simultaneously for vertical and horizontal inheritance of genetic material; e.g., see [2,11,15]. Several algorithms, both exact and heuristic, have been proposed for combining a pair of gene trees into a phylogenetic network (which is an NP-hard problem) [6,9,16,20,3,14,8,5,25,26,13]. Given that phylogenomic analyses involve a large number of gene trees, it is imperative to develop algorithms and tools that infer such networks from multiple gene trees.

L. Bleris et al. (Eds.): ISBRA 2012, LNBI 7292, pp. 213–224, 2012.

In this paper, we address the computational problem of inferring a phylogenetic network from a set of gene trees. A few methods were introduced very recently to address this problem. Huson and Rupp [12] proposed a method for summarizing a collection of gene trees using *cluster networks*, which differ from the phylogenetic network model we address here. Beiko and Ragan [4] discussed aggregating inferred HGT events from pairwise tree comparisons, and discussed three strategies for this task; yet, they did not implement the strategies, nor did they study their performance. Iersel *et al.* [24] developed the CASS method, which is an efficient algorithm for inferring a minimal phylogenetic network that contains all the clusters of taxa displayed by the input gene trees, but not necessarily the input gene trees themselves. Further, Wu [27] recently introduced the PIRN algorithm for obtaining lower and upper bounds on the amount of reticulations necessary for reconciling a set of input gene trees. Finally, we introduced two methods for estimating the amount of reticulation, as well as inferring a phylogenetic network, from a collection of gene trees [17]. Both methods are based on obtaining estimates for the set of trees from pairwise distance calculations. In particular, the approach of combining pairwise reconciliations to obtain a solution for the entire set of trees, though *ad hoc*, showed good performance in simulation studies. Here, we discuss the suitability of combining pairwise reconciliations (reconciliations obtained from comparing pairs of trees in the set of all gene trees) to obtain a phylogenetic network that reconciles a set of gene trees. Further, we define the problem formally and provide an integer linear programming (ILP) solution for it. It is important to mention that this makes an improvement on the previous algorithm not only in terms of time, but also accuracy. Finally, we study the performance of the method, and compare it to other methods, on synthetic data sets and one biological data set. The results show that our method is fast in practice, and that it produces accurate estimates of the phylogenetic network. Our method makes two main assumptions about the input gene trees: (1) the trees are accurate (that is, we ignore incongruence among the trees due to error in the gene tree inference), and (2) reticulation is the only biological cause of gene tree incongruence. Of course, these two assumptions may be violated in practice; developing methods that relax these assumptions is the holy grail in this area, and is well beyond the scope of a single paper. Further, it is important to note that related methods also make these assumptions. We believe our method is best used to obtain a quick analysis assuming the two conditions hold, and then following up with careful inspection of the reticulations inferred by the method. Therefore, these strict assumptions notwithstanding, we believe the method amounts to an important contribution.

2 Phylogenetic Networks and Trees

In this section, we discuss the plausibility, as well as challenges, of obtaining a phylogenetic network that reconciles a set of gene trees by combining networks obtained from analyses of pairs of trees in the set. First, we begin by formally defining networks and the relationship between networks and trees.

Definition 1. *For a set of taxa $\mathscr{X}$, a phylogenetic $\mathscr{X}$-network, or $\mathscr{X}$-network, N is an ordered pair (G, f), where $G = (V, E)$ is a directed, acyclic graph (DAG) with $V = \{r\} \cup V_L \cup V_T \cup V_N$, where*

1. *$indeg(r) = 0$ (r is the* root *of N);*
   $\forall v \in V_L, indeg(v) = 1$ and $outdeg(v) = 0$ (V_L are the leaves of N);
2. *$\forall v \in V_T, indeg(v) = 1$ and $outdeg(v) \geq 2$ (V_T are the tree nodes of N);*
3. *$\forall v \in V_N, indeg(v) = 2$ and $outdeg(v) \geq 1$ (V_N are the reticulation nodes of N); and,*
4. *$E \subseteq V \times V$ are the network's edges. (we distinguish between reticulation edges, edges whose heads are reticulation nodes, and tree edges, edges whose heads are tree nodes or leaves).*

The mapping $f : V_L \to \mathscr{X}$ is a bijection function from V_L to the set of taxa $\mathscr{X}$.

A phylogenetic $\mathscr{X}$-tree is an $\mathscr{X}$-network in which $V_N = \emptyset$. While a network N represents the evolution of a set of genomes, these genomes can be partitioned into (non-recombining) regions $R_1, R_2, \ldots, R_k$, each of which has a treelike evolutionary history T_i. In other words, the set $\mathscr{T} = \{T_1, \ldots, T_k\}$ is a subset of the set of all trees *displayed* by the network N. More formally, $\mathscr{T} \subseteq \mathscr{T}(N)$, where $\mathscr{T}(N)$ is the set of *all* trees obtained as follows from N: (1) for each node of in-degree 2 remove one of the two incoming edges and (2) for each node u of in-degree and out-degree 1, remove u along with its incident edges, and add a new edge to connect u's parent to u's child (this step is repeated until no such nodes u remain).

3 Pairwise and Set-Wise Reconciliation of Trees

A main problem of interest is the following [15].

Problem 1. (Set-wise HGT Inference)

> **Input:** A collection of gene trees $\mathscr{G} = \{GT_1, \ldots, GT_k\}$, each modeling the evolutionary history of a genomic region of a set $\mathscr{X}$ of taxa.
> **Output:** A phylogenetic network N with the smallest number of reticulation nodes (a minimal network) such that $\mathscr{G} \subseteq \mathscr{T}(N)$.

When the input consists of exactly two trees, we refer to this as the Pairwise HGT Inference problem.

We will now make two assumptions that we will use throughout this paper. First, a solution to the Pairwise HGT Inference problem is obtained by solving for $SPR(T_1, T_2)$. In other words, we will take a smallest set Ξ of Subtree Prune and Regraft (SPR) moves that transform T_1 to T_2, and obtain N by adding Ξ to T_1. Second, in the Pairwise HGT Inference problem, we will assume that the first tree is a species tree ST. In this problem, we will assume that a species tree is given, so that the distance from each tree in $\mathscr{G}$ to the species tree is computed. We discuss below a potential solution to the problem when a species tree is not given as part of the input.

A natural and simple candidate for solving the Set-wise HGT Inference problem by utilizing solutions for the pairwise problem is to take

$$SPR(ST, \mathscr{G}) = \cup_{gt \in \mathscr{G}} SPR(ST, gt) \quad (1)$$

as an estimate of the solution of the *Set-wise HGT Inference* problem. Obviously, $|SPR(ST, \mathscr{G})|$ is larger than or equal to the number of reticulation nodes in a solution to the problem. An advantage of this approach is that fast exact algorithms and heuristics exist for obtaining $SPR(ST, gt)$, as described above, and taking the union of pairwise reconciliations is very simple computationally. Indeed, we developed a method, referred to here as M2, based on this idea, and showed that it yields good performance [17]. Nonetheless, several issues need to be resolved.

First, it is possible that the optimal solutions for $SPR(ST, gt)$ do not lead to the optimal solution for $SPR(ST, \mathscr{G})$, and cause overestimation in a "global" estimate of reticulations. For example, consider the HGT scenarios shown in Fig. 1. If we take the union of the four pairwise solutions computed by $SPR(ST, GT_i)$,

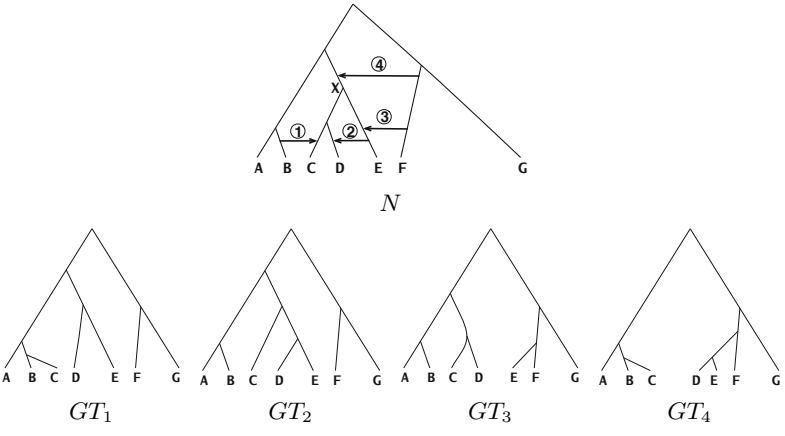

Fig. 1. A phylogenetic network with four independent HGT scenarios. The species tree ST in this case is the network N without the four HGT edges. The gene whose tree is GT_i $(1 \leq i \leq 3)$ underwent HGT event (i), and the gene whose tree is GT_4 underwent HGT events (1) and (4). The combined effect of HGTs (1) and (4) on the gene tree topology is the same as the combined effect of HGTs (1), (2) and (3).

for $1 \leq i \leq 4$, we obtain a set of four HGT edges: $\Xi_1 = \{[B \to C], [E \to D], [F \to E], [F \to X]\}$. However, a smallest solution for the Set-wise HGT Inference problem given the species tree ST and the four gene trees contains three HGT edges, which is the set $\Xi_2 = \{[B \to C], [E \to D], [F \to E]\}$. In this case, the HGT edge $[F \to X]$ is not needed, since its effect can be simulated by the two HGT edges $[E \to D]$ and $[F \to E]$, once the HGT edge $[B \to C]$ is applied. However, notice that in this case, the solution that truly reflects what

happened is Ξ_1, since the HGT event denoted by $[F \rightarrow X]$ did occur, even though its effect on the topologies of gene trees can be simulated by the other three HGT edges. In other words, while the union of pairwise solutions may not provide a smallest global solution, it may provide a solution that is closer to the true HGT scenarios that took place at the genomic level. Further, under these scenarios, where a smallest solution is a proper subset of the union of pairwise solutions, post-processing via gradual elimination of members of the union can yield a smallest solution. However, it is not guaranteed that the smallest solution is a subset of the union of pairwise reconciliations.

Second, $SPR(ST, gt)$ may not be unique; in fact, the number of possible solutions to the Pairwise HGT Inference problem can be exponential in the size of a solution [21]. To account for this issue, we need to consider all solutions, or a large number of them when obtaining all is computationally infeasible, for each pair of trees. Without accounting for multiple solutions, methods such as M2 [17] that agglomerate pairwise solutions obtain biased estimates.

A third issue that requires special attention is the following: while solutions to the Pairwise HGT Inference problem may be acyclic (that is, the inferred HGT events, when added to the species tree, do not result in cycles), taking the union of solutions cannot guarantee acyclicity. When this occurs, the solution is not a phylogenetic network as given by Definition 1.

4 MURPAR

As mentioned above, our former method M2 [17] uses the approach given by Eq. (1), but it does not address the last two issues raised above. Here, we propose our heuristic MURPAR (MUlti-tree Reconciliation using PAirwise Reconciliations) for addressing these issues. Let ST be a species tree and $\mathscr{G} = \{GT_1, \ldots, GT_k\}$ be a collection of gene trees. Also let $S_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,m_i}\}$ be the set of all optimal pairwise solutions on the pair $\langle ST, GT_i \rangle$. Then, M2 counts the frequency with which each potential reticulation edge appears throughout $s_{i,j}, 1 \leq i \leq k, 1 \leq j \leq m_i$, and calculates the set of reticulation edges such that 1) it covers at least one solution for all trees, and 2) it maximizes its frequency value, in the assumption that the frequency reflects how often a reticulation edge is used in each solution and maximizing the frequency would result in a smallest set of reticulation edges.

However, with multiple solutions (e.g., obtained by using RIATA-HGT [16,20]), a reticulation edge occurring multiple times in $s_{i,j}, \forall 1 \leq j \leq m_i$, contributes more to the frequency than those occurring once. As a result, a solution would be biased towards the edges occurring multiple times in solutions. If we define $S = \bigcup_{1 \leq i \leq k, 1 \leq j \leq m_i} s_{i,j}$, then MURPAR seeks the smallest set $S' \subseteq S$ that satisfies the property $[\forall 1 \leq i \leq k,\ \exists s_{i,j} \in S_i\ s.t.\ s_{i,j} \subseteq S']$. MURPAR solves this problem using integer linear programming (ILP). We define binary variables as follows:

(A) B_s, $\forall s \in S$. B_s will take value 1 if SPR move s is selected as an element of S', and 0 otherwise.
(B) P_{ij}, $\forall 1 \leq i \leq k, \forall 1 \leq j \leq m_i$. P_{ij} will take value 1 if all SPR moves in the optimal pairwise solution s_{ij} are selected, and 0 otherwise.

Finally, the ILP program is:

minimize $\sum B_s$
subject to
- $P_{ij} = \left[\wedge_{y \in s_{ij}} B_y\right], \forall 1 \leq i \leq k, \forall 1 \leq j \leq m_i$
- $[\vee_{1 \leq j \leq m_i} P_{ij}] = 1, \forall 1 \leq i \leq k$

Here, $\wedge$ and $\vee$ represent logical 'and' and logical 'or', respectively. All these constraints can be turned into linear constraints by introducing auxiliary variables as follows:

- $a = (b_1 \wedge b_2 \wedge \cdots \wedge b_p)$, where all variables are binary, can be turned into the linear inequalities $-1 \leq 2b_1 + 2b_2 + \cdots + 2b_p - 2pa \leq 2p - 1$.
- $(b_1 \vee b_2 \vee \cdots \vee b_p) = 1$, where all variables are binary, can be turned into the linear inequality $b_1 + b_2 + \cdots + b_p \geq 1$.

When a species tree ST is not given, we repeat MURPAR with $GT_i, 1 \leq i \leq k$ as the species tree and choose the smallest S'_i as S'.

Addressing cyclicity. As discussed above, the solution thus far may result in cyclic graphs, which are not phylogenetic networks. To address this issue, MURPAR post-processes the results to avoid the solutions with cycles using a straightforward cycle detection algorithm. If a minimal solution is found to have a cycle, MURPAR skips it and inspects the next minimal solution (minimal in terms of the number of reticulation nodes). While all solution candidates found by MURPAR may have cycles, and thus MURPAR returns no solution, we found through extensive simulations that this was never the case. Similarly, it was shown in [1] and confirmed in [23] that cycles may not be a major concern for reticulation detection algorithms when run on real data sets or synthetic data sets that are generated under realistic models.

5 Experimental Evaluation

5.1 Data

Simulations were conducted on 30- and 50-taxon phylogenies. For 30-taxon data sets, 10 random trees were generated using PHYL-O-GEN tool [18] as "species tree" under birth-death model, and 5 horizontal gene transfer events were simulated between pairs of branches on the species trees using Galtier's tool [7]. The simulation of horizontal gene transfer were conducted individually 10 times on each species tree, so totally 100 networks are generated from the simulation. Since Galtier's tool does not provide the details of simulated transfer events, we modified the tool to have it report the simulated transfers that it added. From

the set of 32 gene trees contained in each network, 4, 8, 12, 16, 24, and 32 gene trees were randomly sampled and used as input to the methods.

For 50-taxon data sets, the same procedure as above was applied, except that the number of horizontal gene transfer events simulated was 10, and the sampling was made over 1024 gene trees.

For biological data, we used the Poacaea data set, which was originally sequenced by the Grass Phylogeny Group, and was used to test both CASS [24] and PIRN [27]. Binary trees were constructed for six loci: *ITS*, *ndhF*, *phyB*, *rbcL*, *rpoC* and *waxy* [19]. Since the gene trees had different sets of leaves, we selected the gene trees for *ndhF*, *phyB*, *rbcL*, *rpoC2* and *ITS*, and restricted them to 14 leaves that they have in common.

5.2 Methods and Accuracy Measure

The Pairwise HGT Inference problem was solved using the RIATA-HGT method [16,20] as implemented in the PhyloNet software package [22]. Other pairwise inference tools including SPRIT [10] were tested as well, but results were almost identical; therefore, all results reported here are based on pairwise solutions obtained by RIATA-HGT. We used the GLPK ILP Solver to solve the ILP formulation of MURPAR. For comparison, we also ran PIRN [27] and Method M2 of [17].

We do not use CASS [24] for comparison. As indicated by the authors in [24], while CASS computes a minimal network N from an input set $\mathscr{C}$ of clusters of a set of gene trees $\mathscr{T}$, it is not guaranteed that $\mathscr{T} \subseteq \mathscr{T}(N)$. More formally, if $\mathscr{C}$ is the set of all clusters of taxa displayed by the trees in $\mathscr{T}$, the network N computed by CASS is the minimal network that displays all clusters in $\mathscr{C}$. It is important to note that if a network N displays all clusters of a set of trees, N does not necessarily display all the trees themselves. It is easy to see that if N is minimal for $\mathscr{C}$ and N' is minimal for $\mathscr{T}$ ($\mathscr{C}$ is the set of all clusters of trees in $\mathscr{T}$), then the number of reticulation nodes in N is smaller than or equal to that in N', because the problem with $\mathscr{C}$ is less restrictive than that with $\mathscr{T}$. An illustration of this issue is given in Fig. 2.

When a method is run on a collection $\mathscr{T} = \{T_1, \ldots, T_k\} \subseteq \mathscr{T}(N)$ displayed by network N, we record the number of reticulations that the method estimates. The accuracy of the methods is considered better as the estimated number becomes smaller. Besides the accuracy, we also assess the run time of the methods.

5.3 Results on Simulated Data

Running Times. Due to space limitations, we focus on the 30-taxon data sets; we observed very similar trends on 50-taxon data sets. We first assess the running times of the methods; It is important for a method to maintain reasonable computation speed in order to be of general application for large-scale data analysis. Through the experimental validation, we record the run times of the 100 cases for the methods by sample size in Fig. 3. In interpreting the figure, we focus on two properties regarding the computation speed, the overall run time and the frequency of outliers in time.

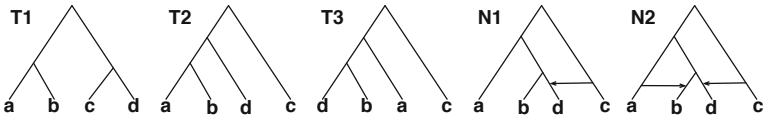

Fig. 2. Illustration of the difference between the formulation used by MURPAR, M2 [17] and PIRN [27] on the one hand, and the one used by CASS [24] on the other hand. For the input set of gene trees $\mathscr{T} = \{T1, T2, T3\}$, CASS computes a network with a single reticulation node ($N1$), since this network displays all clusters of the gene trees. However, MURPAR, M2, and PIRN compute minimal networks with two reticulation nodes, such as $N2$, since 2 is the minimum number of reticulation nodes required in a network that displays all the three gene trees.

In terms of the overall run time, it is clear that MURPAR runs fast across varying sample sizes. Compared with M2, the gain in speed of MURPAR is mainly attributed to the use of the ILP solver, since they share the underlying computation structure. The figure also shows that the gap between MURPAR and PIRN closes as the number of trees increases. However, the outliers in the case of PIRN are still much slower than those of MURPAR and M2. This point requires further elaboration. Holding the input size (in terms of the number of gene trees) fixed, the variance in the speed of MURPAR is very similar, whereas that of PIRN is very large. This somehow indicates dependence of the PIRN on the structure of the problem, and lack of dependence of MURPAR on such a structure. It may be that the smaller the number of gene trees, the fewer the constraints, and hence the larger the space that PIRN explores. In the case of MURPAR, the pairwise solutions constrain the search space significantly, giving the method gains in speed.

Accuracy. The numbers of reticulation nodes estimated by each of the methods are shown in Fig. 4. It is worth mentioning that even though the true networks were produced by adding 5 HGT edges, the true number of reticulations may be smaller, depending on the size of the input, since, for example, when only 4 gene trees are sampled, some HGT events may not be observable. Even though network N has $m > 1$ reticulation nodes, this does not necessarily mean that the collection $\mathscr{T}$ of trees, with $|\mathscr{T}| \geq 2$ will have all trees to allow for detecting m. For example, consider the collection $\mathscr{T}$ that has only a pair of trees whose SPR distance is 1. In this case, the number of detectable HGTs is 1, and not m. However, the number of detectable HGTs is expected to increase as more trees are given, and the results in the figure satisfy the expectation.

As more trees are sampled as the input, a naive method for the reticulation estimation would estimate more reticulations simply because the size of the data increases. M2 follows the expectation, in that the estimation gets larger with more trees. However, MURPAR overcomes this problem, even though it is based on the same idea. Rather, both the estimations of PIRN and MURPAR hardly increase with the input size. Between them, it is clear that as more trees are given,

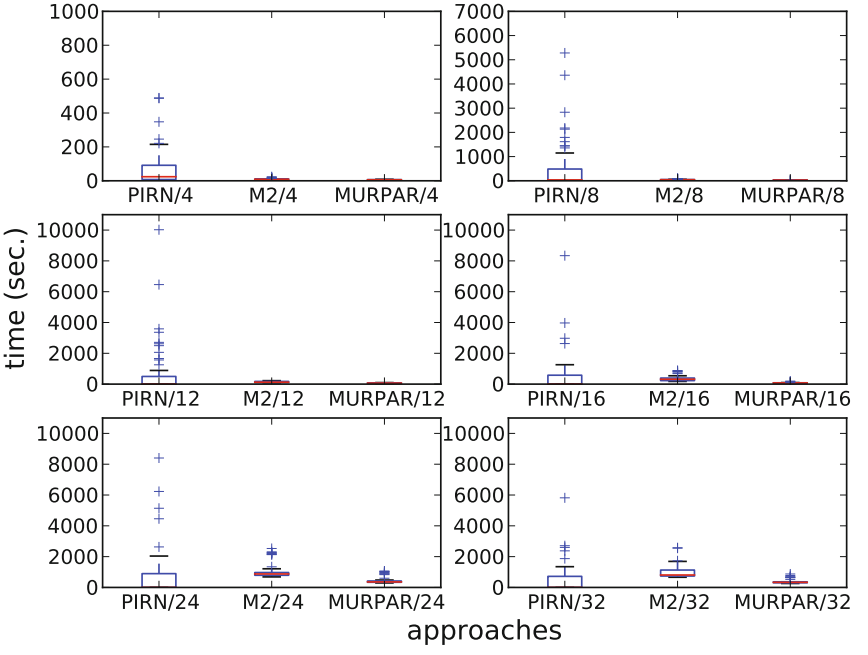

Fig. 3. Running times (in sec.) of the three methods PIRN, M2, and MURPAR. The numbers after the '/' are the numbers of gene trees in the input.

the estimate becomes more accurate in MURPAR than in PIRN, particularly for inputs of sizes ≥ 12. Even though the maximum difference between PIRN and MURPAR is one reticulation event on average, this difference gets larger for larger data sets.

5.4 Results on Biological Data

In order to evaluate how well the methods perform for biological data, we ran them on the five gene trees of the *Poaceae* data set (see Section 5.1 for details of this data set). Table 1 reports the estimated number of reticulations and the amount of time taken by the methods. Notice that we also ran PIRN in coarse mode (CoarsePIRN) on them (which is a faster, yet less accurate, version of PIRN).

PIRN identifies the lowest estimate of the number of reticulations, but it took the longest time to obtain the estimate. On the other hand, M2 and MURPAR obtained estimates that are higher by just one reticulation event, two and three orders of magnitude faster, respectively. In other words, MURPAR and M2 produce very accurate results within very short amounts of time. Between MURPAR and M2, the difference is negligible on this size of data. However, it is easy to see that the difference will grow with the increase of the input data, and

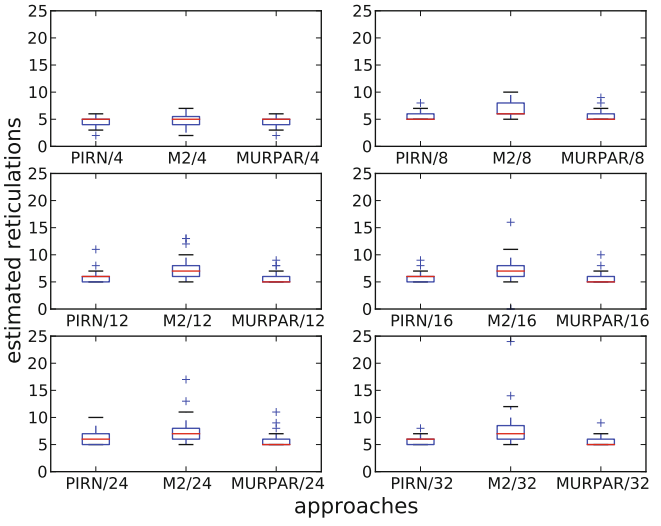

Fig. 4. Numbers of reticulations estimated by each of the three methods PIRN, M2, and MURPAR. The numbers after the '/' are the numbers of gene trees in the input.

Table 1. The number of estimated reticulation events and the run time (in seconds) of the methods on the five gene trees for *ndhF*, *phyB*, *rbcL*, *rpoC2*, and *ITS* in the Poaceae dataset

Approaches	#Reticulations	Time (sec.)
PIRN	13	2143
M2	14	16
MURPAR	14	8
CoarsePIRN	16	58

MURPAR will be preferred in large-scale data analysis. For PIRN, we also ran it in coarse mode. Notice that while PIRN in coarse mode is much faster than PIRN, and is comparable to M2 in terms of speed, the estimates it produces are higher than the other three methods. Considering the run times they take to work on the small input data (5 trees on 14 species), it is clear that PIRN would run the slowest in large-scale data analyses.

6 Conclusions

In this paper, we introduced MURPAR, a method for inferring a phylogenetic network from a collection of gene trees, under the assumption that all incongruence in the gene trees is due to reticulate evolutionary events. While MURPAR is not guaranteed to compute a minimal network, it produces an upper bound

on the minimum number of reticulations required to reconcile all gene trees in the input. Performance analysis on both synthetic and biological data sets shows that the MURPAR method is both accurate and fast. Further, MURPAR's run time does not vary much within the same sample size, and has fewer outliers than other methods.

The idea of employing pairwise reconciliations in reconciling a set of gene trees has added advantages in that pairwise reconciliations can be computed in parallel or in a distributed fashion, thus speeding up the overall computation, and improvements to pairwise reconciliation methods will automatically translate into improvement of the MURPAR method. Direct interpretability of the results from the direct relationship of the solutions between Set-wise HGT Inference and Pairwise HGT Inference is another advantage of MURPAR that it is easy to identify the dynamics between any gene trees in the tree set from the estimates of Set-wise HGT Inference.

Acknowledgement. This work was supported in part by NSF grant DBI-1062463, grant R01LM009494 from the National Library of Medicine, and an Alfred P. Sloan Research Fellowship to L.N. Further, the work was supported in part by the Shared University Grid at Rice funded by NSF under Grant EIA-0216467, and a partnership between Rice University, Sun Microsystems, and Sigma Solutions, Inc.

References

1. Addario-Berry, L., Hallett, M., Lagergren, J.: Towards identifying lateral gene transfer events. In: Proc. Eighth Pacific Symp. Biocomputing (PSB 2003), pp. 279–290 (2003)
2. Semple, C., Baroni, M., Steel, M.: A framework for representing reticulate evolution. Annals of Combinatorics 8, 391–408 (2004)
3. Beiko, R.G., Hamilton, N.: Phylogenetic identification of lateral genetic transfer events. BMC Evolutionary Biology 6, 15+ (2006)
4. Beiko, R.G., Ragan, M.A.: Untangling hybrid phylogenetic signals: Horizontal gene transfer and artifacts of phylogenetic reconstruction. Methods Mol. Biol. 532, 241–256 (2009)
5. Bordewich, M., Linz, S., John, K.S., Semple, C.: A reduction algorithm for computing the hybridization number of two trees. Evolutionary Bioinformatics 3, 86–98 (2007)
6. Bordewich, M., Semple, C.: On the computational complexity of the rooted subtree prune and regraft distance. Annals of Combinatorics 8, 409–423 (2004)
7. Galtier, N.: A model of horizontal gene transfer and the bacterial phylogeny problem. Systematic Biology 56(4), 633–642 (2007)
8. Goloboff, P.A.: Calculating SPR distances between trees. Cladistics 24, 591–597 (2007)
9. Hallett, M.T., Lagergren, J.: Efficient algorithms for lateral gene transfer problems. In: Proc. 5th Ann. Int'l Conf. Comput. Mol. Biol. (RECOMB 2001), pp. 149–156. ACM Press, New York (2001)

10. Hill, T., Nordstrom, K., Thollesson, M., Safstrom, T., Vernersson, A., Fredriksson, R., Schioth, H.: Sprit: Identifying horizontal gene transfer in rooted phylogenetic trees. BMC Evolutionary Biology 10(1), 42+ (2010)
11. Huson, D.H., Bryant, D.: Application of phylogenetic networks in evolutionary studies. Molecular Biology and Evolution 23(2), 254–267 (2006)
12. Huson, D.H., Rupp, R.: Summarizing Multiple Gene Trees Using Cluster Networks. In: Crandall, K.A., Lagergren, J. (eds.) WABI 2008. LNCS (LNBI), vol. 5251, pp. 296–305. Springer, Heidelberg (2008)
13. Linz, S., Semple, C.: A cluster reduction for computing the subtree distance between phylogenies. Annals of Combinatorics 15, 465–484 (2011)
14. MacLeod, D., Charlebois, R.L., Doolittle, F., Bapteste, E.: Deduction of probable events of lateral gene transfer through comparison of phylogenetic trees by recursive consolidation and rearrangement. BMC Evolutionary Biology 5 (2005)
15. Nakhleh, L.: Evolutionary phylogenetic networks: models and issues. In: Heath, L., Ramakrishnan, N. (eds.) The Problem Solving Handbook for Computational Biology and Bioinformatics, pp. 125–158. Springer, New York (2010)
16. Nakhleh, L., Ruths, D., Wang, L.-S.: RIATA-HGT: A Fast and Accurate Heuristic for Reconstructing Horizontal Gene Transfer. In: Wang, L. (ed.) COCOON 2005. LNCS, vol. 3595, pp. 84–93. Springer, Heidelberg (2005)
17. Park, H.J., Jin, G., Nakhleh, L.: Algorithmic strategies for estimating the amount of reticulation from a collection of gene trees. In: Proceedings of the 9th Annual International Conference on Computational Systems Biology, pp. 114–123 (2010)
18. Rambaut, A.: Phylogen: Phylogenetic tree simulator package (2002), `http://evolve.zoo.ox.ac.uk/software/PhyloGen/main.html`
19. Schmidt, H., Martin, W.: Phylogenetic Trees from Large Datasets Inaugural–Dissertation zur. PhD thesis, Heinrich-Heine-Universitt, Dsseldorf (2003)
20. Than, C., Nakhleh, L.: SPR-based tree reconciliation: Non-binary trees and multiple solutions. In: Proceedings of the Sixth Asia Pacific Bioinformatics Conference, pp. 251–260 (2008)
21. Than, C., Ruths, D., Innan, H., Nakhleh, L.: Confounding factors in HGT detection: Statistical error, coalescent effects, and multiple solutions. Journal of Computational Biology 14(4), 517–535 (2007)
22. Than, C., Ruths, D., Nakhleh, L.: PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. BMC Bioinformatics 9, 322 (2008)
23. Tofigh, A., Hallett, M., Lagergren, J.: Simultaneous identification of duplications and lateral gene transfers. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1–19 (January 2011)
24. van Iersel, L., Kelk, S., Rupp, R., Huson, D.H.: Phylogenetic networks do not need to be complex: using fewer reticulations to represent conflicting clusters. Bioinformatics [ISMB] 26(12), i124–i131 (2010)
25. Wu, Y., Wang, J.: Fast Computation of the Exact Hybridization Number of Two Phylogenetic Trees. In: Borodovsky, M., Gogarten, J.P., Przytycka, T.M., Rajasekaran, S. (eds.) ISBRA 2010. LNCS, vol. 6053, pp. 203–214. Springer, Heidelberg (2010)
26. Whidden, C., Beiko, R.G., Zeh, N.: Fast FPT Algorithms for Computing Rooted Agreement Forests: Theory and Experiments. In: Festa, P. (ed.) SEA 2010. LNCS, vol. 6049, pp. 141–153. Springer, Heidelberg (2010)
27. Wu, Y.: Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees. Bioinformatics [ISMB] 26(12), 140–148 (2010)