SPR-BASED TREE RECONCILIATION: NON-BINARY TREES AND MULTIPLE SOLUTIONS*

C. THAN and L. NAKHLEH

Department of Computer Science
Rice University
6100 Main Street, MS 132
Houston, TX 77005, USA
Email: {*cvthan,nakhleh*}*@cs.rice.edu*

The SPR (subtree prune and regraft) operation is used as the basis for reconciling incongruent phylogenetic trees, particularly for detecting and analyzing non-treelike evolutionary histories such as horizontal gene transfer, hybrid speciation, and recombination. The SPR-based tree reconciliation problem has been shown to be NP-hard, and several efficient heuristics have been designed to solve it. A major drawback of these heuristics is that for the most part they do not handle non-binary trees appropriately. Further, their computational efficiency suffers significantly when computing multiple optimal reconciliations. In this paper, we present algorithmic techniques for efficient SPR-based reconciliation of trees that are not necessarily binary. Further, we present divide-and-conquer approaches that enable efficient computing of multiple optimal reconciliations. We have implemented our techniques in the PhyloNet software package, which is publicly available at `http://bioinfo.cs.rice.edu`. The resulting method outperforms all existing methods in terms of speed, and performs at least as well as those methods in terms of accuracy.

Keywords: Subtree prune and regraft; phylogenetic tree reconciliation; horizontal gene transfer.

1. Introduction

Comparing phylogenetic trees and quantifying the similarities and differences among their topologies play important roles in studying the quality of phylogeny reconstruction methods and understanding gene evolution within species trees. As such, several tree transformation operations have been introduced, and their induced distance measures have been studied extensively.[1] One such operation is the *subtree prune and regraft* (SPR) operation. An SPR operation, or move, transforms a phylogenetic tree by cutting (pruning) a subtree and attaching (regrafting) it from its root to a different branch in the tree. Studies of this operation and the distance measure it induces on pairs of trees have increased significantly in recent years, mainly due to the central role it plays in detecting *reticulate*, i.e., non-treelike, evolutionary histories, such as horizontal gene transfer, hybrid speciation, and recombination.[2–6] In a nutshell, the occurrence of reticulate evolutionary events results in different genomic regions having incongruent, or disagreeing, trees. One way of identifying these

*This work is supported in part by a Department of Energy grant DE-FG02-06ER25734 and a National Science Foundation grant CCF-0622037.

events is based on the comparison of such trees and determining the minimal set of SPR moves that reconcile the incongruities among these trees, as well as their disagreements with the species tree, if such a tree is known. Therefore, the computational problem that has been addressed in this context is: given two trees T_1 and T_2, find a minimal set of SPR moves that transform T_1 into T_2.

While recently developed methods have made significant progress in terms of accuracy (number and location of the SPR moves) and efficiency (time), there remain two central issues that have not been addressed appropriately by these methods:

Non-binary trees. Reconstructed phylogenetic trees often contain multi-furcating nodes; i.e., nodes with more than two children (in the rooted tree setting); see Figure 1. The way these nodes are handled by methods for estimating SPR moves affects the number and location of those moves, and currently most algorithms and tools do not handle non-binary trees.

Multiple minimal sets of SPR moves. It is often the case that a minimal set of SPR moves that reconciles two trees is not unique, and the number of such sets may be exponential in the size of the set;[7] see Figure 1. Current tools that compute multiple solutions take several days on moderate-sized trees, and run out of memory on larger ones.

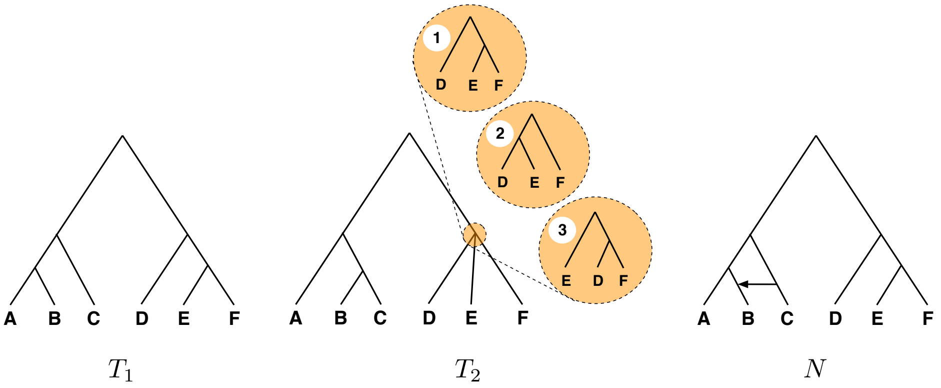

Fig. 1. An illustration of multiple solutions and non-binary trees. The multi-furcating ancestral node of **D**, **E**, and **F** in T_2 can be refined, or resolved, in three different ways. However, refinement (1) results in a clade that is identical to that in tree T_1, and hence requires no SPR moves. On the other hand, each of the two refinements (2) and (3) requires one SPR move to reconcile the clade between the two trees. As for the clade that contains the leaves **A**, **B**, and **C**, one SPR move is needed to reconcile it. Nonetheless, three such SPR moves (in T_1) are possible: (1) **C**→**B**, (2) **B**→**C**, and (3) **X**→**A**, where **X** is a node on the edge from the root to the ancestor of all three leaves. The phylogenetic network N is obtained by adding the set $\Xi = \{\mathbf{C} \rightarrow \mathbf{B}\}$ of edges to T_1.

In this paper we address these two issues by introducing algorithms for efficient refinement of trees to yield minimal sets of SPR moves, as well as algorithms for collapsing identical components of the trees to enable efficient handling of large trees in terms of time and space requirements, while not affecting the accuracy of the computed set of SPR moves. For computing multiple minimal sets of SPR moves, we utilize the sharing among solutions and present algorithmic techniques for efficient computing and displaying of multiple solutions. Besides their value in taming the computational complexity of the problem, the outcome of these techniques has biological significance since it summarizes the "essentiality" of the SPR moves (i.e., which moves *must* be considered in order to account for incongruence, and which ones have *alternatives* that can be considered and have the same effect), and hence the support for the corresponding reticulate evolutionary event.

We have extended the RIATA-HGT method[6] for reconciling trees by incorporating our new algorithms and techniques. The resulting method outperforms all existing methods in terms of computing time, and performs at least as well in terms of accuracy (number of SPR moves in a minimal set, and number of such minimal sets) on binary trees. For non-binary trees, most existing methods are not able to handle them (the tools simply quit, giving an error message stating that the input trees are not binary). The extended method has been implemented in the PhyloNet software package, which is available at `http://bioinfo.cs.rice.edu`. In this paper we use HGT as the guiding biological example of a reticulate evolutionary event, but the method can be applied to trees even when other reticulate evolutionary events have occurred.

2. Preliminaries

Trees and networks. Let $T = (V, E)$ be a tree, where V and E are the *tree nodes* and *tree edges*, respectively, and let $\mathscr{L}(T)$ denote its leaf set. Further, let $\mathscr{X}$ be a set of taxa (species). Then, T is a phylogenetic tree over $\mathscr{X}$ if there is a bijection between $\mathscr{X}$ and $\mathscr{L}(T)$. A tree T is said to be *rooted* if the edges in E are directed and there is a single internal node r with in-degree 0. Let $T = (V, E)$ be a rooted tree, and u be a node in V. We denote by T_u the subtree of T whose root is node u, and $L(u)$ the set of leaves in T_u. A phylogenetic tree t is a *clade* of a phylogenetic tree $T = (V, E)$ if there exists a node $v \in V$ such that $t = T_v$. Given two phylogenetic trees $T = (V, E)$ and $T' = (V', E')$, with $\mathscr{L}(T) = \mathscr{L}(T')$, a *maximal pair of matching clades* is a pair $\langle t, t' \rangle$ such that $t = T_u$ and $t' = T'_{u'}$ for $u \in V$ and $u' \in V'$, $t = t'$, and (1) either u and u' are the roots of the two trees, or (2) $(x, u) \in E$, $(x', u') \in E'$, and $T_x \neq T'_{x'}$. Given a set $X \subseteq \mathscr{L}(T)$, we denote by $lca_T(X)$ the *least common ancestor* of X in T.

A phylogenetic network $N = N(T) = (V', E')$ over the taxa set $\mathscr{X}$ is derived from $T = (V, E)$ by adding a set Ξ of edges to T, where each edge $h \in \Xi$ is added in three steps: (1) split an edge $e = (u, v) \in E$ by adding a new node, v_e, and replacing e by two edges (u, v_e) and (v_e, v); (2) split an edge $e' = (u', v') \in E$ by adding a new node, $v_{e'}$, and replacing e' by two edges $(u', v_{e'})$ and $(v_{e'}, v')$; and (3) add a directed *HGT edge* from v_e to $v_{e'}$. In this case, we write $N = T + \Xi$. Figure 1 shows a phylogenetic network obtained by adding a single HGT edge to the tree T_1. Finally, we denote by $\mathscr{T}(N)$ 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 (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 incident edges, and add a new edge from u to v. This operation is called a *forced* *contraction*. For example, in Figure 1, the tree T_1 and tree T_2 (with clade refinement (1)) are the only members of $\mathscr{T}(N)$.

Reticulate evolution and the SPR operation. Let $T = (V, E)$ be a rooted tree. An SPR move involving edges $e = (u, v)$ and $e' = (u', v')$ in E (u' is not reachable from the root of the tree T through node v) deletes edge e, splits edge e' into two edges by adding a new node $v_{e'}$, as described above, and adds a new edge from $v_{e'}$ to v. Equivalently, the SPR move may involve a node x instead of edge e', in which case, the move deletes edge e and adds a new edge from x to v. As mentioned above, when HGT occurs, the evolutionary history of the species may not be represented by phylogenetic trees; rather, *phylogenetic networks* are the appropriate model.[8] In the phylogeny-based HGT detection problem, a pair of trees T_1 and T_2 (usually, a species/gene tree pair) is given, and a minimal set Ξ of edges is sought so that $T_2 \in \mathscr{T}(N)$, where $N = T_1 + \Xi$. The minimization requirement simply reflects a parsimony criterion: in the absence of any additional biological knowledge, the simplest solution is sought. This problem has been shown to be related to finding the minimal set of SPR moves that transform T_1 into T_2[a] and several heuristics for solving the problem using SPR moves have been recently introduced.[4–6,9–12]

Non-binary trees and tree compatibility. An edge $e = (u, v)$ in a rooted tree T is contracted by deleting it and merging the two nodes u and v into a single node x (the edges incident from x are the union of the edges incident from u and v). We say a tree T' is a *contraction* of tree T, if T' is obtained by contracting a set of edges in T. Equivalently, we say that T is a *refinement* of tree T'. An edge $(u, v) \in E$ induces a *split* $A|B$, where $A = L(v)$, and $B = \mathscr{L}(T) - A$. A split $A|B$ is non-trivial if $|A| > 1$ and $|B| > 1$. We say that two splits $A|B$ and $C|D$ are *compatible* if at least one of the four intersections $A \cap C$, $A \cap D$, $B \cap C$ and $B \cap D$ is empty. We denote by $\pi(T)$ the set of all splits induced by the edges of tree T. We say that two trees T_1 and T_2 are compatible if $\pi(T_1)$ and $\pi(T_2)$ are pairwise compatible. When one or both of the trees T_1 and T_2 are not necessarily binary, the phylogeny-based HGT detection problem is slightly modified, since we seek a minimal set of SPR moves that makes T_1 compatible with, and not necessarily identical to, tree T_2. In other words, a minimal set Ξ of HGT edges is sought so that (1) $N = T_1' + \Xi$, (2) $T_2' \in \mathscr{T}(N)$, and (3) T_1', T_2' are refinements of T_1 and T_2, respectively, that result in the minimum size of such a set Ξ. The network N in Figure 1, with a single HGT edge, is an example of a solution to the problem for the pair of trees T_1 and T_2.

3. Algorithmic Techniques

As mentioned above, existing methods for solving the phylogeny-based HGT detection problem do not handle non-binary tree appropriately (in fact, most tools do not run on non-binary trees), nor do they handle multiple minimal solutions efficiently. In this section we present algorithmic techniques for efficient handling of these two cases.

[a] An HGT edge involving two edges e and e', or an edge e and a node x is obtained by computing the SPR move as defined, with the only difference that edge e is not deleted.

3.1. *Handling Non-binary Trees: Refine and Collapse*

As illustrated in Figure 1, when multi-furcating nodes are present, different refinements may lead to different estimates of the minimum number of SPR moves needed to reconcile two trees. Since we seek the minimum number of SPR moves to reconcile two trees T_1 and T_2, which are not necessarily binary, our proposed solution is to *maximally refine* both trees to obtain two trees T_1' and T_2', respectively, such that the number of SPR moves required to reconcile T_1' and T_2' is minimum among all possible refinements of T_1 and T_2. For example, under this approach, refinement (1) of tree T_2 in Figure 1 is preferred over the other two possible refinements. We now present an efficient algorithm for solving this problem.

(1) Generate all nontrivial splits of T_1 and T_2.

(2) For each split $A|B$ of T_1 that is not a split of T_2 but compatible with every split of T_2:

   (a) Let $u = lca_{T_2}(A)$, and let $x_1, x_2, \dots x_k$ be the children of u such that $A = \cup_{i=1}^{k} L(x_i)$. If no such set of children exists, redo this step for $u = lca_{T_2}(B)$.

   (b) Delete all edges (u, x_i), for $1 \leq i \leq k$, add a new node x' with new edge (u, x'), and add k new edges (x', x_i) for $1 \leq i \leq k$.

(3) Repeat Step 2 for all splits of T_2 with respect to T_1.

The algorithm takes $O(|\mathscr{L}(T_1)|^2)$ time and maximally refines the two trees T_1 and T_2 without affecting the number of SPR moves required to reconcile them; details are omitted due to space constraints.

Once the two trees are refined, we *collapse* them to achieve reduction in the size of the trees, without affecting the set of SPR moves. The idea is that clades that are identical in both trees do not require any SPR moves to reconcile them, and hence we can preprocess the trees by collapsing them into single leaf nodes. Formally, for every maximal pair of matching clades $\langle t, t' \rangle$ in the two trees, replace both t and t' by a single node that is labeled with the same label ℓ_t, where ℓ_t is unique (per tree). If k is the minimum number of SPR moves required to transform tree T_1 into tree T_2, then the same SPR moves are required to transform T_1' into T_2', where T_1' and T_2' are obtained from T_1 and T_2, respectively, through the application of any number of collapse operations.[5]

A special case that requires special handling is that of *identical chains* in the two trees. Allen and Steel[2] handled chains in binary trees. We now generalize that to include trees that are not necessarily binary (assuming that the collapse operation has been applied maximally to the trees, as described above). Two sequences $\mathscr{P} = \langle u_1, u_2, \dots, u_k \rangle$ where $u_i \in V(T_1)$ and $\mathscr{P}' = \langle u_1', u_2', \dots, u_k' \rangle$ where $u_i' \in V(T_2)$, $k \geq 2$, are said to be identical if (1) u_{i+1} is parent of u_i and u_{i+1}' is parent of u_i', $1 \leq i \leq k-1$; (2) all clades whose roots are children of u_i, u_i' and not in $\mathscr{P}, \mathscr{P}'$ are identical, $2 \leq i \leq k$; and (3) all clades whose roots are children of u_1 and u_1', except for exactly one, are identical. (Note that the requirement in (3) is to distinguish identical chains from identical clades.) The value $k-1$ is the chain length. Bordewich and Semple[5] showed that an identical chain can be replaced in both binary trees by an identical chain of only three leaves $\langle a, b, c \rangle$, without affecting the SPR moves. With the definition of identical chains above, this rule can be applied to non-binary trees. The reason is that clades whose roots are children of u_i (u_i') and not in $\mathscr{P}$ ($\mathscr{P}'$) can be thought of as being "contained" in *one* big clade, and therefore the rule for binary trees can be used. Figure 2 shows an example of identical chains in non-binary trees and how they can be replaced. Applying this operation can further reduce the size of trees, as the collapse operation does not apply in this case.

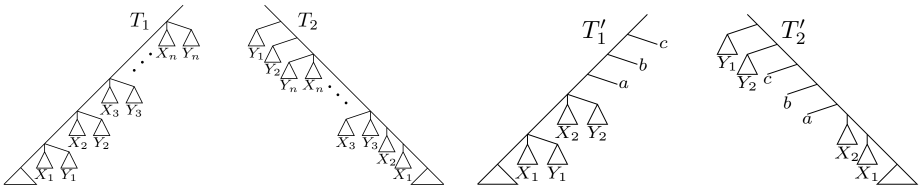

Fig. 2. Replacement of identical chains in non-binary trees. The identical chain $\langle(X_3, Y_3), \ldots, (X_n, Y_n)\rangle$ in the trees T_1 and T_2 is replaced by the chain $\langle a, b, c\rangle$, which results in a significant decrease in the size of trees, without affecting the number of SPR required to reconcile the two trees.

While Bordewich and Semple stated this operation can be done in time that is polynomial in the number of leaves in a binary tree, we now present an $O(|\mathscr{L}(T_1)|^2)$ time algorithm for applying the collapse operation maximally to a pair of trees (not necessarily binary) T_1 and T_2 to replace all identical chains by 3-leaf chains. In order to collapse identical maximal chains in the two trees, a bottom-up scan of the clades of T_1, comparing them for compatibility with clades of T_2, and replacing such pairs of identical clades by leaf nodes, can be achieved in $O(|\mathscr{L}(T_1)|^2)$ as well.

(1) Compute the set of leaves $L(v)$ for every internal node v in T_1 and T_2.
(2) Starting from a deepest node v in T_1 , let u be the parent of v, and do the following:
    (a) Compute $L(u) - L(v)$ and find an edge $(u', v') \in E(T_2)$ such that $L(u') - L(v') = L(u) - L(v)$. If no such edge (u', v') exists, let v be the next deepest node that has not been examined, and go to step (2).
    (b) Restrict tree T_1 and T_2 to the leaves $L(u) - L(v)$. If the two restricted trees are identical, replace u, u' by their parents in T_1 and T_2, and repeat this substep.
    (c) Identical chains obtained in (2)(b) are maximal. If their length is at least 3, replace them by 3 new leaves that preserve orientation relative to the roots of T_1 and T_2.
(3) Repeat step (2) with the next deepest node that has not been examined.

The algorithm replaces all maximal identical chains in trees by 3-leaf chains, and takes $O(|\mathscr{L}(T_1)|^2)$ time; details are omitted due to space constraints.

3.2. *Algorithmic Techniques for Efficient Enumeration of Minimal Solutions*

Than *et al.* have recently shown that the number of solutions to the phylogeny-based detection problem is $O(3^k)$, where k is the minimum number of SPR moves required to reconcile the two trees.[7,13] In this section, we exploit the fact that there are only $O(n^2)$ possible distinct SPR moves that can be applied to a tree on n taxa, to design strategies for efficient enumeration of all minimal solutions.

We denote by $T - t$ the tree obtained from T by removing the clade t and applying forced contractions. Let T_1 and T_2 be two phylogenetic trees on the same set of taxa $\mathscr{X}$, with sets $S(T_1)$ and $S(T_2)$ of clades. In this section we assume the trees have been maximally refined and collapsed. We denote by $\mathbf{Sol}(T_1, T_2)$ the set of all minimal sets of SPR moves that reconcile the two trees (i.e., the set of all solutions to the HGT detection problem). We define a mapping $f : S(T_1) \rightarrow S(T_2)$, such that $f(t_1) = t_2$ when $L(t_1) = L(t_2)$, and $f(t_1) = nil$ when there does not exist $t_2 \in S(T_2)$ such that $L(t_1) = L(t_2)$. Given two trees T_1 and T_2, and the mapping f, we process the trees as follows. Suppose there are m clades $t_1^1, \ldots, t_1^m$ in T_1 such that $f(t_1^i) \neq nil$. Then, we generate $m + 1$ pairs of trees $\langle\alpha^i, \beta^i\rangle$, $1 \leq i \leq m + 1$,where α^i and β^i, $1 \leq i \leq m$, are obtained from t_1^i and $f(t_1^i)$ by replacing in each of them every clade t' and $f(t')$ $(f(t') \neq nil)$, respectively, by a single leaf with the same name in both clades. The last pair $\langle\alpha^{m+1}, \beta^{m+1}\rangle$ is obtained from T_1 and T_2 by removing from them all clades t_1^i and $f(t_1^i)$, $1 \leq i \leq m$, respectively. We call these $m + 1$ pairs, the *decomposition* of the pair of trees T_1 and T_2, denoted $\mathscr{D}(T1, T2)$.

Lemma 3.1. *Let $T1$ and $T2$ be two phylogenetic trees whose decomposition is $\mathscr{D} = \{\langle\alpha^i, \beta^i\rangle : 1 \leq i \leq p\}$. Then, $\mathbf{Sol}(T_1, T_2) = \mathbf{Sol}(\alpha^1, \beta^1) \times \cdots \times \mathbf{Sol}(\alpha^p, \beta^p)$.*

This lemma states that a minimal solution for a pair of trees can be obtained by taking the union of minimal solutions from each of the pairs in the decomposition $\mathscr{D}$, and gives the basis for our divide-and-conquer strategy. In this strategy, a decomposition of the two trees is first performed, the HGT detection problem is solved on each pair in the decomposition separately, and then the cartesian product of the sets of minimal solutions of these pairs is taken as the set of all minimal solutions of the trees.

Notwithstanding the gains achieved by the divide-and-conquer approach, it may be the case that a few pairs have large clades in them. However, empirical performance shows that large clades may have fewer solutions, given the lack of "locality" in the HGT events involved.[7] To enable efficient handling of these clades, we consider HGT event equivalence, and describe how this concept may lead to further reductions in time for computing minimal solutions.

Equivalence of minimal sets of SPR moves. Given a tree T and a set Ξ of SPR moves defined on T, we denote by $T \uparrow \Xi$ the tree obtained from T by applying the SPR moves, followed by forced contractions, that correspond to the HGT edges in Ξ.

Definition 3.1. Given two sets Ξ_1 and Ξ_2 of HGT edges defined on a tree T, we say that Ξ_1 is equivalent to Ξ_2 (with respect to tree T), denoted $\Xi_1 \equiv \Xi_2$, if $T \uparrow \Xi_1$ is compatible with $T \uparrow \Xi_2$.

The $\equiv$ relation on sets of SPR moves is an equivalence relation. Further, equivalent sets of SPR moves from two minimal solutions have the same cardinality, as we now show.

Lemma 3.2. *Let $T1$ and $T2$ be two trees, and Ξ_1 and Ξ_2 be two sets of SPR moves in $\mathbf{Sol}(T_1, T_2)$. If $X' \equiv Y'$ for $X' \subseteq \Xi_1$ and $Y' \subseteq \Xi_2$, then $|X'| = |Y'|$.*

Based on the above observations and the defined equivalence relation, we have the following strategy for efficient enumeration of multiple equivalent solutions to the HGT detection problem:

(1) Find a solution Ξ to the problem.
(2) Partition Ξ into $\Xi_1, \ldots \Xi_m$ such that for any other solution Y, Y can be partitioned into $Y_1, \ldots, Y_m$ where
   (a) $\forall \Xi_i$, $\exists Y_j$ such that $\Xi_i \equiv Y_j$, and
   (b) m is the maximum cardinality of such a partition of Ξ.
(3) For each Ξ_i, compute its equivalence class $[\Xi_i]$.
(4) The set of solutions is $Z_1, Z_2, \ldots, Z_m$, where $Z_i \in [\Xi_i]$.

As described above, when the HGT events (SPR moves) are "local", i.e., do not span a large portion of the tree, the decomposition process yields small components, and hence the number of equivalence classes is large and their sizes are small. However, when the HGT events are more global, we expect the number of solutions to be smaller, and hence the number of equivalence HGT edge-sets (SPR moves) to be small as well.[7]

4. Empirical Performance

We used the `r8s` tool[14] to generate four random trees, T_i, $i \in \{10, 25, 50, 100\}$, where i denotes the number of taxa in the tree. The `r8s` tool generates molecular clock trees; we deviated the trees from this hypothesis by multiplying each edge in the tree by a number randomly drawn from an exponential distribution. The expected evolutionary diameter (longest path between any two leaves in the tree) is 0.2. Then, from each model "species" tree T_i, we generated five different "gene" trees, $T_{i,j}$, $j \in \{1, 2, 3, 4, 5\}$, where j denotes the number of simulated HGT events (SPR moves) applied to T_i to obtain $T_{i,j}$.For each T_i and $T_{i,j}$, $i \in \{10, 25, 50, 100\}$ and $j \in \{1, 2, 3, 4, 5\}$, and for each sequence length $\ell \in \{250, 500, 1000, 2000, 4000, 8000\}$, we generated 30 DNA sequence alignments $S_i^\ell[k]$ and $S_{i,j}^\ell[k]$, $1 \leq k \leq 30$, whose evolution was simulated down their corresponding trees under the GTR+Γ+I (gamma distributed rates, with invariable sites) model of evolution, using the Seq-gen tool.[15] We used the parameter settings of.[16] Then, from each sequence alignment, we reconstructed a tree TNJ using the Neighbor Joining (NJ) method.[17] At the end of this process we had 4 trees T_i, 20 trees $T_{i,j}$, 720 NJ trees $TNJ_i^\ell[k]$, and 3600 NJ trees $TNJ_{i,j}^\ell[k]$ ($i \in \{10, 25, 50, 100\}$, $j \in \{1, 2, 3, 4, 5\}$, $1 \leq k \leq 30$, and $\ell \in \{250, 500, 1000, 2000, 4000, 8000\}$). To compute solutions to the HGT detection problem, as well as the number of such solutions, we applied two methods to pairs of species and gene trees: LatTrans[10,18] and the extended RIATA-HGT,[6] which implements the strategies for handling non-binary trees and computing multiple minimal scenarios, as described in the previous section. Both tools were applied to pairs $(TNJ_i^\ell[k], TNJ_{i,j}^\ell[k])$ of binary trees; since LatTrans cannot handle non-binary trees, we do not report any comparisons for that. Due to space limitations, we only show results for 50-taxon data sets, shown in Figure 3. In each run of a tool on a pair of trees, we computed two values: the minimum number of inferred HGT events (SPR moves), and the number of such minimal solutions found by the method. We report the average of all 30 runs and actual running times for each combination of i, j, and ℓ. In Figure 3 we observe a similar relative performance between the two methods in terms of the number of HGT events estimated and the number of minimal solutions computed. Notice that both methods almost identically overestimate

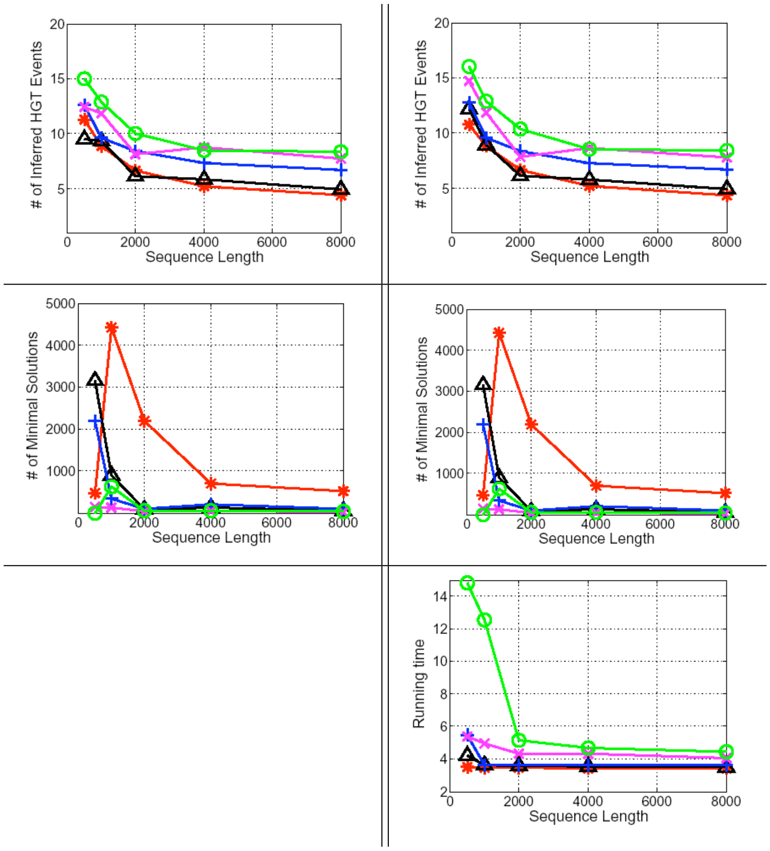

Fig. 3. The performance of LatTrans (left column) and RIATA-HGT (right column) in terms of the minimum number of HGT edges (top row), number of minimal solutions (middle row), and actual running times in seconds (bottom row), as functions of the sequence length. All results are obtained from 50-taxon NJ trees. LatTrans took several days on each pair of 50-taxon trees, and for sequence length 250 it crashed after 4 days without returning results (hence we omit its running time graph). Each curve corresponds to one of the five actual numbers of HGT events: $\star$: 1 HGT; $\triangle$: 2 HGTs; $+$: 3 HGTs; $\times$: 4 HGTs; and $\circ$: 5 HGTs.

the minimum number of HGT events, or SPR moves needed to reconcile the two trees, and this overestimation decreases as the sequence length increases. This is a result of the large amount of wrong edges in the trees inferred by NJ, and the fact that these errors made by NJ decrease as the sequence length increases, since NJ is *statistically consistent*. Further, notice that as the sequence length increases and the estimates of the number of HGT events decreases, the number of minimal solutions decreases drastically, which is in agreement with the results showing that the number of solutions is proportional to their size, and can be exponential in these sizes.[7] However, where the big difference is pronounced between the two methods is in terms of running times. RIATA-HGT and LatTrans found the same number of minimal solutions; yet, RIATA-HGT found these solutions in a few seconds, whereas LatTrans ran for several days on each of these data sets, and crashed on all data sets for sequence length 250—which is the case where a large number of HGT events are identified. Notice that even though the number of solutions for the case of 1 HGT is much larger than that of 5 HGTs, RIATA-HGT finds the solutions in the former case much more quickly, which is a consequence of the algorithmic strategies employed by RIATA-HGT to exploit sharing and avoid explicit enumeration of all solutions.

5. Conclusions

In this paper, we considered the problem of reconciling a pair of phylogenetic trees, mainly to estimate the amount of non-treelike evolutionary events in the evolution of a set of organisms. We addressed the two issues of appropriate handling of non-binary trees and efficient enumeration of equally optimal solutions. We developed a set of algorithmic techniques for handling both issues, and incorporated these techniques into the RIATA-HGT method. The outcome was a method that performed at least as accurately as existing methods, and significantly outperformed existing methods.

References

1. C. Semple and M. Steel, *Phylogenetics*
2. B. Allen and M. Steel, *Annals of Combinatorics* **5**, 1 (2001).
3. M. Baroni, C. Semple and M. Steel, *Annals of Combinatorics* **8**, 391 (2004).
4. L. Nakhleh, T. Warnow and C. Linder, Reconstructing reticulate evolution in species–theory and practice, in *Proc. 8th Ann. Int'l Conf. Comput. Mol. Biol. (RECOMB04)*, 2004.
5. M. Bordewich and C. Semple, *Annals of Combinatorics* , 1 (2005).
6. L. Nakhleh, D. Ruths and L. Wang, RIATA-HGT: A fast and accurate heuristic for reconstrucing horizontal gene transfer, in *Proceedings of the Eleventh International Computing and Combinatorics Conference (COCOON 05)*, ed. L. Wang2005. LNCS #3595.
7. C. Than, D. Ruths, H. Innan and L. Nakhleh, *Journal of Computational Biology* **14**, 517 (2007).
8. B. Moret, L. Nakhleh, T. Warnow, C. Linder, A. Tholse, A. Padolina, J. Sun and R. Timme, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **1**, 13 (2004).
9. V. Makarenkov, *Bioinformatics* **17**, 664 (2001).
10. M. Hallett and J. Lagergren, Efficient algorithms for lateral gene transfer problems, in *Proc. 5th Ann. Int'l Conf. Comput. Mol. Biol.*, (ACM Press, New York, 2001).
11. R. Beiko and N. Hamilton, *BMC Evolutionary Biology* **6** (2006).
12. D. MacLeod, R. Charlebois, F. Doolittle and E. Bapteste, *BMC Evolutionary Biology* **5** (2005).
13. C. Than, D. Ruths, H. Innan and L. Nakhleh, Identifiability issues in phylogeny-based detection of horizontal gene transfer, in *Proceedings of the Fourth RECOMB Comparative Genomics Satellite Workshop*, eds. N. El-Mabrouk and G. Bourque, Lecture Notes in Bioinformatics (LNBI), Vol. 42052006.
14. M. Sanderson, Analysis of rates (r8s) of evolution (2006), Available from http://loco.biosci.arizona.edu/r8s/.
15. A. Rambaut and N. C. Grassly, *Comp. Appl. Biosci.* **13**, 235 (1997).
16. D. Zwickl and D. Hillis, *Systematic Biology* **51**, 588 (2002).
17. N. Saitou and M. Nei, *Mol. Biol. Evol.* **4**, 406 (1987).
18. L. Addario-Berry, M. Hallett and J. Lagergren, Towards identifying lateral gene transfer events, in *Proc. 8th Pacific Symp. on Biocomputing (PSB03)*, 2003.