TOWARDS IDENTIFYING LATERAL GENE TRANSFER EVENTS

L. ADDARIO-BERRY AND M. HALLETT
McGill Centre for Bioinformatics, McGill University, Montréal, Canada
E-mail: {laddar,hallett}@mcb.mcgill.ca

J. LAGERGREN
Stockholm Bioinformatics Center and Dept. of Numerical Analysis and Computer Science, KTH, Stockholm, Sweden
E-mail: jensl@nada.kth.se

This paper is concerned with evaluating the performance of the model and algorithm in [5] for detecting lateral gene transfers events. Using a Poisson process to describe arrival times of transfer events, a simulation is used to generate "synthetic" gene and species trees. An implementation of an efficient algorithm in [5] is used to estimate the minimum number of transfers necessary to explain disagreements between the generated gene and species trees. Our first result suggests that the algorithm can solve realistic size instances of the problem. Our second result suggests that the mean error and variance are low when saturation does not occur. Additionally, certain plausible evolutionary events allowed by our model of evolution used to generate gene and species trees but not detectable by the algorithm occur rarely implying the framework should work well in practice. Our third, surprising result suggests that the number of optimal scenarios is on average low for realistic input sizes.

1 Introduction

Recent findings have reinforced the view that evolutionary relationships between taxa (i.e. the species tree) cannot be inferred from a single gene family (i.e. a single gene tree) due to genomic events such as *gene duplication*, *gene loss*, *gene convergence*, and *lateral gene transfer* (a.k.a. horizontal gene transfer) [3,4,5,6,7,8,9,10]. In essence, these events cause gene trees to not be equal to the species tree; that is, they "disagree". To explain such disagreements, various models have been developed that assume a simplified evolutionary process restricted to a subset of these genomic events. A natural computational problem is to find the most parsimonious *scenario* that explains how, via these events, the disagreements between the gene tree and species tree arise.

A well studied model is the *duplication, loss model* [4,9]. Here a species tree S and a gene tree T are given. Via a (computationally easy) least common ancestor mapping from the vertices of T to the vertices of S, it is possible to identify all vertices in T that correspond to duplication events and locate where they occur in the evolution represented by S. In a manner analogous to the duplication, loss model, the authors of [5,6] construct a model for lateral gene transfer. Here we are given a (hypothetically correct) species tree S and a (hypothetically correct) gene tree T. The goal is to find a most parsimonious scenario that explains disagreements between the two trees using lateral gene transfer events. This work extends previous work on the *subtree transfer* [7,8] and *network* [10] models in that the resulting scenarios are biologically sound. The price of this biological realism is an increase in complexity of the model itself - it is difficult to study analytically the behavior of the system.

This paper is concerned with evaluating experimentally the performance of the model and algorithm in [5] for detecting lateral transfers. Our experimental technique is analogous to those commonly used by the phylogenetic community. We begin with a method for simulating evolution with lateral transfer events. Using a Poisson process with rate parameter λ to describe the arrival times of transfer events, we use a discrete event simulation to probabilistically generate "synthetic" gene trees w.r.t. the species tree and λ. Next, using an implementation of the algorithm in [5] for what is termed activity level 1 (at most one gene per gene family may exist in a genome at any point in the evolution of the taxa), the minimum number of transfers necessary to explain disagreement between the gene and species tree is estimated.

The simulation of evolution is pessimistic in the sense that biologically plausible evolutionary events can occur that are not detectable by the algorithm. The first such event termed a *useless* transfer is analogous to a "back substitution" in molecular sequence evolution. Useless transfers will not be detected in a parsimony-based framework and hence the algorithm from [5] will underestimate the true number of transfers. The second type of degenerate event is termed a *transfer-loss* event. In certain cases, a transfer event followed by a gene loss event can cause the algorithm from [5] to grossly under-estimate or over-estimate the true number of transfer events for a gene and species tree. It is conjectured that such events, although biologically plausible, occur with low frequency. Furthermore, the authors of [5] conjecture that any algorithm that does detect such transfer-loss events would be computationally infeasible for even small instances of the problem. One of the primary goals of this paper is to test the frequency of harmful transfer-loss events under a reasonable model of evolution.

We answer several questions concerning the model from [5]. Our first result suggests that, although the running time of the algorithm is high, it is fast enough to solve instances one expects to encounter in practice. On a desktop machine, we managed to compute minimum cost scenarios when the number of leaves of the species tree (taxa) n is 300 and the number of transfers τ is 20. The algorithm from [5] has a worst case running time of $O(2^{4\tau} n^2)$ and consists of two phases. Although it is possible to construct examples where both phases are required, in over $10,000$ experiments we did not find a single example of a scenario that required the second phase (cycle removal). This suggests that the first phase of the algorithm is sufficient for realistic data sets and implies a more optimistic $O(2^{2\tau} n^2)$ running time for such data sets. Our second result suggests that the mean error (actual number of transfers used minus minimum cost of scenario found) and variance are low for all realistic values of n, τ and λ. This indicates that the number of harmful transfer-lost and other degenerate events are negligible and the framework should work well in practice. When λ is sufficiently large as to cause low levels of *saturation*, the algorithm still gives reasonable estimates of the number of transfers. Our third, surprising result suggests that the number of valid minimum cost scenarios is on average low for realistic values of n, τ and λ.

2 Definitions

We consider rooted directed trees where the arcs are directed from the root towards the leaves and a vertex has out-degree at most 2. We call such a tree a *rooted* tree. For such a tree T, $V(T)$ denotes the set of vertices and $A(T)$ denotes the set of arcs. The *internal vertices* of T are $V(T) \setminus L(T)$. The *root* of a tree T is denoted $r(T)$. For $u \in V(T)$, any vertex v reachable from u by a directed path is a *descendant* of u (this means that u is a descendant of u). We denote this by $v \leq_T u$. We also say that u is an *ancestor* of v ($u \geq_T v$). We say that v is a *proper descendant* (*proper ancestor*) of u, if $v \leq_T u$ ($v \geq_T u$) and $v \neq u$ and denote this relationship by $v <_T u$ ($v >_T u$). Both a *gene tree* T and a *species tree* S are binary rooted directed trees. We assume that $n = |L(S)| \geq |L(T)|$. By a rooted forest we mean a union of disjoint rooted trees. The set of leaves of a rooted forest F is denoted $L(F)$. For a vertex $u \in V(F)$, let F_u be the rooted subtree of F consisting of the vertices of $V(F)$ reachable by directed paths from u. Let T be a rooted tree. For $X \subseteq L(T)$, the *least common ancestor* of X in T, written $lca_T(X)$, is defined as follows: if $X = \{v\}$, then $lca_T(X) = v$; otherwise, $lca_T(X)$ is the vertex v such that $X \subseteq L(T_v)$ but $X \nsubseteq L(T_u)$ for each proper descendant u of v. Let T be a gene tree and let F, $F \subset T$, be a forest. The mapping $\lambda_{F,S} : V(F) \to V(S)$ is defined as follows: $\lambda_{F,S}(v) = lca_S(L(F_v))$. A *mixed graph* G is a graph containing arcs as well as undirected edges. The arcs of G are denoted $A(G)$, the edges $E(G)$, and the vertices $V(G)$. If G is a mixed graph and A is a set of arcs, then $G \cup A$ is used to denote the mixed graph with arcs $A(G) \cup A$, edges

$E(G)$, and vertices $V(G)$. For a set of edges E, $G \cup E$ is defined similarly. A *directed mixed cycle* is a mixed graph where each vertex has total degree 2, which contains arcs and edges, and where the cycle can be traversed in a direction that respects the arcs but edges may be traversed in either direction. If A is a set of arcs, then $E(A)$ denotes the underlying undirected edges, i.e. $E(A) = \{(u, v) : \langle u, v\rangle \in A\}$. For $u, v \in V(T)$, let $P^T_{u,v}$ be the unique directed path between u and v in T. Let length of $P^T_{u,v}$, $|P^T_{u,v}|$, be the number of arcs in $P^T_{u,v}$.

3 Lateral Transfer Scenarios

The following section gives a simplified version of the model and an intuitive explanation of the algorithm that appeared in [5] for *lateral transfer scenarios*. A more detailed description of this work can be found in [6].

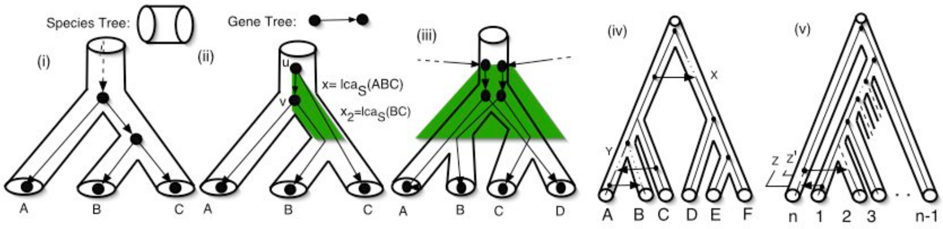

Figure 1. (i) The gene tree (thin lines and vertices) is drawn within the species tree (thick pipes). We do not explicitly direct the thick pipes of the species tree; however, note that the species tree is also rooted and directed. Here the gene tree and species trees agree. That is, the root of the species tree and the root of the gene tree both have A and BC (the ancestor of B and C) as their children. In (ii), the gene tree disagrees with the species tree since the root of the gene tree has C and AB as children. If we postulate a lateral transfer event from either v to child A or v to B, the resulting scenario would then be 1 active (see H-moves below). At any point during the evolution represented by the shaded region in (ii-iii), there exist two copies of the gene in the genome of these ancestral organisms. These examples are said to be 2 active. A second example of how activity levels > 1 can arise is depicted in (iii). Here two lateral transfer events have occurred prior to the root of $ABCD$. (iv) Two examples of useless transfers. (v) A harmful transfer-loss event.

The $\lambda_{T,S}$ mapping from a gene tree T to a species tree S places each vertex of the gene tree at its least common ancestor in the species tree. Figure 1 graphically depicts gene and species trees that agree (i) and trees that disagree (ii-iii). The model from [5] is built upon the assumption that: *Since A and B are siblings in the gene tree, either the ancestral gene AB must have been present in the ancestor of A and B in the species tree (i.e. the $lca_S(A, B)$)*

or a lateral transfer event has occurred from the A lineage to the B lineage (or vice versa). The postulated transfer thus explains why the gene tree has the sibling pair A and B. To capture this mapping of *gene trees into species trees*, we introduce the notions of *lateral transfer schemes* and *scenarios*.

Definition 1 *A* lateral transfer scheme *(or, simply* scheme*) for a species tree S is a pair (S', A') where S' is a subdivision of S and $A' \subseteq \{\langle x, y \rangle : x, y \in V(S') \setminus V(S), x \neq y\}$ such that: (1) the mixed graph $S' \cup E(A')$ does not contain a directed mixed cycle, (2) the tail of each arc in A' has in-degree 1 and out-degree 2 in $S' \cup A'$, and (3) the head of each arc in A' has in-degree 2 and out-degree 1 in $S' \cup A'$.*

Figure 2 gives an example of a scheme. In order for our *scenario* for a gene tree T and species tree S to be biologically meaningful, it must satisfy the following constraints.

Definition 2 *A* lateral transfer scenario *(or simply* scenario*) for a species tree S and a gene tree T is a triple (S', A', g) where (S', A') is a scheme for S and $g : V(S') \to V(T)$ such that: (1) $g(r(S')) = r(T)$; (2) if v_1 and v_2 are children of v_0 in T, then there exists x_0 with children x_1 and x_2 in $S' \cup A'$ (where $x_1 \neq x_2$) s.t. $v_i = g(x_i)$, for $i = 0, 1, 2$; (3) for each $v \in V(T)$, the vertices $\{x \in V(S') : g(x) = v\}$ induce a directed path in S'; (4) $g(l) = l$, for all $l \in L(S)$.*

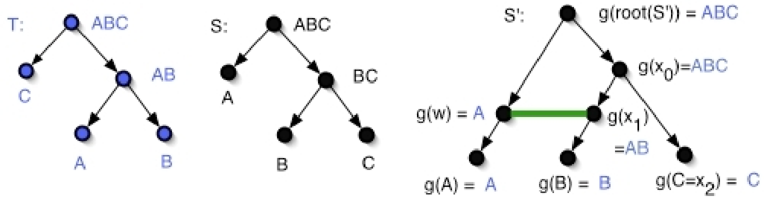

Figure 2. An example of a *scenario* for gene tree T and species tree S. S' represents a scheme for S. In S' there is only one transfer (between x_1 and w in S) and this arc is included in A'.

The *cost* of (S', A', g) w.r.t. T is simply $|A'|$. The *cost of S and T*, denoted $\tau(S, T)$, is the minimum cost of any scenario (S', A', g) for S and T. A detailed justification of each of these conditions along with a more in-depth discussion of mixed cycles in given in [5,6]. It is easy to verify that Figure 2 depicts a scenario for the gene tree T and species tree S with cost $|A'| = 1$. We note that Definition 2 is a simplification of a more general definition in [5,6]. In particular, the above definition holds only for what is termed *activity level* 1. Figure 1 (ii-iii) give an intuitive explanation of activity level.

The input to the τ-Transfer Problem is a species tree S, a gene tree

T, and an integer τ. The output is a τ' lateral transfer scenario for S and T, $\tau' \leq \tau$. Let $T' \subseteq T$. The two basic operations the algorithm performs are H-moves and I-moves. In [6], the authors prove that these moves are sufficient to find all optimal scenarios.

Definition 3 (H-fat) *A vertex $x \in V(S)$ is H-*fat *for T' iff there exist $u, v \in \lambda^{-1}_{T',S}(x)$ such that: (1) v is $\leq_T$-minimal in $\lambda^{-1}_{T',S}(x)$, (2) u has out-degree* 2 *in T', and (3) $v <_T u$. If x is H-fat for T', then the two outgoing arcs of v (call them e_1 and e_2) are H-moves at x in T'.*

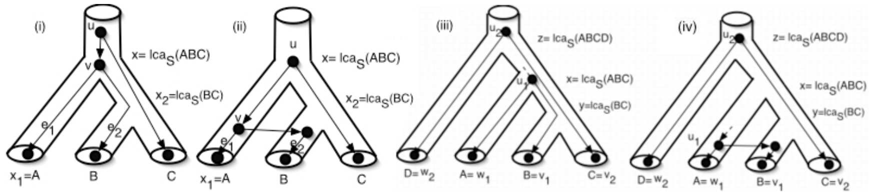

Figure 3. (i) An H-fat vertex x. (ii) one of 2 alternative H-moves at x. (iii) An I-fat vertex x, (iv) one of 4 alternative I-moves.

Definition 4 (I-fat) *A vertex $x \in V(S)$ is I-*fat *for T' iff, for a child y of x, there are arcs $\langle u_1, v_1\rangle, \langle u_2, v_2\rangle \in A(T')$ such that: $\lambda_{T',S}(v_i) \leq_S y$ for $i \in \{1, 2\}$, $\lambda_{T',S}(u_1) = x$, $x \leq_S \lambda_{T',S}(u_2)$, and u_i is $\leq_T$-minimal in $\lambda_{T',S}(\lambda^{-1}_{T',S}(u_i))$ for $i \in \{1, 2\}$. Notice that the latter implies that there are arcs $\langle u_1, w_1\rangle, \langle u_2, w_2\rangle \in A(T') \setminus \{\langle u_1, v_1\rangle, \langle u_2, v_2\rangle\}$. If x is I-fat for T', then the arcs $\langle u_1, w_1\rangle$,$\langle u_2, w_2\rangle$, $\langle u_1, v_1\rangle$, and $\langle u_2, v_2\rangle$, defined as above, are I-moves at x in T'.*

Figure 3 depicts graphically (i) an H-fat vertex, (ii) one of the two H-moves, (iii) an I-fat vertex, and (iv) one of the four I-moves.

A vertex is *fat* for T' iff it is H-fat for T' or I-fat for T'. A *candidate* F where $F \subseteq T$ is a directed forest without isolated vertices such that out degree ≤ 1 for all $v \in V(F)$.

The algorithm proceeds in two phases. Phase I (given below) consists of repeatedly picking a fat vertex and making the appropriate H- or I-moves, until there are no longer fat vertices. The resulting set of candidates is then examined for mixed cycles in phase II. Let C be a queue initially equal to one empty forest $F = \emptyset$ and let $X = \emptyset$. Phase I outputs X when the queue C is empty.

1. Dequeue F from queue C; Let $T' \leftarrow T \setminus F$. Compute $\lambda_{T',S}$. Pick a vertex x of S which is fat for T'.
2. If x is H-fat, let F_1 and F_2 be the candidates obtained from F by making the H-moves e_1 and e_2 respectively;

3. else if x is I-fat, let F_1, F_2, F_3, and F_4 be the candidates obtained from F by making the I-moves e_1, e_2, e_3, e_4 respectively.
4. If there does not exist a fat vertex in $T' \setminus F_i$, then let $X \leftarrow X \cup \{F_i\}$ else enqueue F_i in C.

Although each candidate $F \in X$ found in phase I is guaranteed to be 1-active, it may be the case that a mixed cycle is present. (In [6], an example of a gene and species tree is given where a cycle will exist after phase I). Phase II of the algorithm involves examining each candidate F and finding a minimal set of transfers to make the resultant candidate F' acyclic. The algorithm that enumerates these sets and tests if F' has no mixed cycle has time complexity $4^{|F|}n^2$. The overall running time of the algorithm is therefore $2^{4\tau}n^2$.

4 Simulations

We sketch how “synthetic” gene and species trees are generated for use in the experiments and detail a number of *degenerate* events that the simulation can generate.

Tree Generation. *Species Trees.* We begin by creating a random species tree S on n leaves as follows. Starting with a forest of n singleton vertices and stopping when only one tree exists, we remove two distinct trees x, y from the forest, create an internal vertex z with children x and y, and placed z back in the set. We verified experimentally that the trees had expected $\Theta(lg\ n)$ depth, as predicted analytically. Since species trees reflect the true evolutionary relationships between taxa, they are ultrametric and the weights on arcs correspond to time. The weight of a root to leaf path in S is always 1. Let an *ultrametric species tree* S be a binary rooted directed tree that is arc-weighted by $w : A \rightarrow [0..1)$. For vertex $u \in V(S)$, let $w(u) = \Sigma_{(p,p') \in P^S_{r(S),u}} w(p, p')$. In order to randomly assign weights between $[0..1)$ to the arcs of S s.t. every root to leaf path has total weight 1, we use the following routine in a root to leaf fashion on S:

1. Let $u \in V(S)$. Let l be a leaf in $L(S_u)$ that maximizes $|P^{S_u}_{u,l}|$ and such that no arc in $P^{S_u}_{u,l}$ has yet been assigned a weight. Let $P = P^{S_u}_{u,l}$ be the path $u = p_0, p_1, \ldots, p_{|P|} = l$.
2. Pick uniformly variates $\rho_1, \ldots, \rho_{|P|}$ from interval $[w(u)..1)$; sort $\rho_1, \ldots, \rho_{|P|}$ and assign the resulting weight $\rho'_{i+1} - \rho'_i$ to arc $\langle p_i, p_{i+1} \rangle$.

We experimented with several alternative approaches for generating ultrametric species trees and found that this produces trees that minimize the difference between arc weights under the L_∞ norm. This is important during the gene tree creation phase as it tends to "distribute" the lateral transfer events more evenly throughout the species tree.

Gene Trees. Using an ultrametric species tree S with n leaves generated as described above, we create a gene tree via a discrete event simulation. We assume that lateral transfer events occur according to a Poisson process with rate parameter λ. The four essential events in the simulation are *speciation*, *termination*, *transfer*, and *loss*. We start at the root of S (a speciation event at time 0) and work towards the leaves (termination events at time 1). For vertex $x_0 \in V(S)$ with children x_1, x_2, we generate the appropriate exponential variates for the arrival time of a transfer event along arc $\langle x_0, x_1 \rangle$. If the variate is less than $w(x_0, x_1)$, we schedule the transfer event and check for additional arrivals in the remaining interval along $\langle x_0, x_1 \rangle$. Otherwise, a speciation event for x_1 is scheduled. The process is repeated for arc $\langle x_0, x_2 \rangle$. When a transfer event is encountered at time t along an arc (call it $\langle x, y \rangle$), a second (distinct) arc is chosen uniformly randomly from all other arcs that exist at time t. In other words, only those arcs $\langle x', y' \rangle$ where $w(x') \leq t$ and $w(y') \geq t$ are considered.

Suppose that arc $\langle x, y \rangle$ is the tail and $\langle x', y' \rangle$ is the head of the lateral transfer. If there already exist events scheduled for time t', $t' \geq t$, along arc $\langle x', y' \rangle$, then all such events are aborted. The gene lineages associated with these events are *lost*. This corresponds to the foreign gene "knocking out" the resident gene in the genome of the organism. Such a protocol is necessary if we are to guarantee the 1-activity constraint of the scenario.

Degenerate Events. The simulation of evolution is pessimistic in the sense that the resulting scenarios may contain evolutionary events that are biologically plausible but not detectable by the algorithm. In this sense, the simulation is more general than our model for identifying transfer events. These events have varying effects on the ability of the algorithm to identify the correct number and location of lateral transfer events. We classify these events into two categories: *useless transfers* and *transfer-loss events.*

Useless Transfers. Consider Figure 1 (iv). At the point of evolution marked by X, there is a lateral transfer between two arcs in the species tree that share a common parent. Clearly, the gene tree is not changed by such transfers. In other words, the root of the species tree has children ABC and DEF and the root of the gene tree has children ABC and DEF even though a transfer has occurred at X. In the subtree labeled Y of the species tree, we show an example of two useless transfers that together do not cause the gene tree to disagree with the species tree. This subtree of the species tree has the ancestor AB and C as siblings. In the gene tree, A remains closer to B due

to the "later" lateral transfer and C remains being a sibling with the ancestor AB via the "earlier" lateral transfer.
Transfer-Loss Events. Consider Figure 1 (v). At the point marked Z' in the diagram, a lateral transfer occurs from taxon n to taxon 2. Between point Z' and Z, this lineage is lost. Note that one child of the vertex of the gene tree at point Z' is a transfer event and one child is a loss event; we term this a *transfer-loss event.* Let T be a gene tree and S be a species tree and let τ be the true number of lateral transfer events that occurred during the period of evolution (the true number of lateral transfer events generated by our simulation of evolution). Let τ' be the minimum cost of a scenario for T w.r.t. S (the minimum cost scenario found by our algorithm). When a transfer-loss event occurs, it may be the case that $\tau' < \tau$, $\tau' = \tau$ or $\tau' > \tau$. We term these *helpful*, *harmless*, and *harmful* resp. The example in Figure 1 (v) shows that a single harmful transfer-loss event can cause the algorithm to require $\Omega(n)$ lateral transfers to explain the disagreement between the gene and species tree. It is easy to verify that the minimum cost scenario for this particular example requires $n - 2$ transfer events. It is equally easy to create examples of helpful and harmless transfer-loss events.

5 Experimental Results

For the remainder of this section, let n represent the number of leaves in the species tree, and Ω represent the number of repetitions performed for each experiment. Let τ represent the true number of lateral transfer events generated by a simulation and τ' represent the minimum cost scenario found by the algorithm. Let λ represent the rate parameter in our Poisson process. A *trial* is a species tree and gene tree pair generated by the simulation for a specified λ and n.

The largest gene and species tree for which we can compute the minimum cost scenario has $\tau = 20$ transfers and $n = 300$ leaves. The computation takes approximately 3 days on a standard desktop PC. For $\tau = 10$ and $n = 20$, the computation takes approximately 30 seconds on a standard PC.

In Figure 4 (a), we see that the number of transfers in a simulation rises linearly as a function of λ for a fixed n. This is consistent with a Poisson process and our species tree generation routine. As the rate parameter λ grows, the number of transfers τ will eventually become sufficiently large so that no further transfers will be detected be the algorithm. This is trivially the case if τ exceeds $n - 2$ for a species tree with n leaves. Figure 4 (b) plots the average estimated number of transfers τ' versus the λ. As τ' is consistently less than τ, we may conclude the majority of transfers are not harmful transfer-

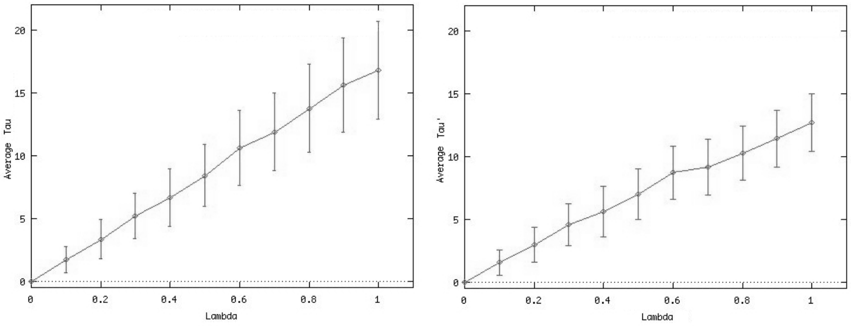

Figure 4. (a) Average τ versus λ. (b) Average τ' versus λ. $\Omega = 300$, $n = 80$.

loss events. When $\lambda = 0.6$, the average value for τ is approximately 11 and τ' is 8.6 with variance 2.11. Note that for $\lambda > 0.6$, one can see slight saturation occurring. To test the saturation point (defined informally as the point where average τ' stops increasing), we generated a large set of random trees. Using half of the set as species trees and the other half as gene trees, the average value of τ' is computed. This should give a very good estimate of the saturation point. For $n = 11$, the trial requires an average of 5.81 lateral transfers with variance 0.65. For $n = 21$, this average is 13.27 with variance 0.87. Trials for trees with larger n suggest that the saturation point is slightly above $n/2$ (graph not shown). At this time, our computational power does not give us an accurate estimate of where this converges. However, it allows us to say with some confidence that any scenario for a gene and species tree, where the cost of the scenario is $> n/2$ transfers, does not represent a meaningful explanation of disagreements between the trees.

When 10 transfers occur in a species tree of size $n = 20$, the algorithm under-estimated the number of transfers by 4.4 with variance 0.85. We note that such a 1 : 2 ratio of transfers to leaves is unrealistic for real-world data sets. We (informally) conjecture that a ratio of 1 : 10 might be more accurate. If such a ratio were true, our algorithm would tend to predict 9 transfer events in a 100 taxa tree where the true transfer number is 10.

Figure 5 and additional graphs available on-line reaffirm that harmful transfer-loss events are very rare and, when they do occur, their effect is negated by the occurrence of useless transfers and the effects of saturation or via the existence of alternative scenarios with approximately the same overall cost. In under 1% of all trials, the algorithm did not find a valid scenario with cost $\leq \tau$. Without exception, every trial had a scenario with cost $\leq \tau + 3$.

We also note that over some $10,000$ trials, we did not find a scenario that

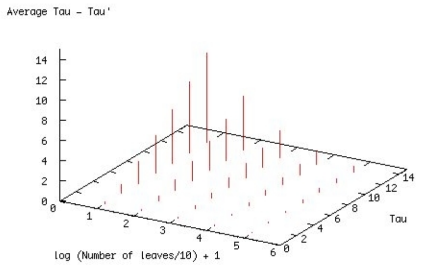

Figure 5. Error versus τ versus logarithm of the number of leaves.

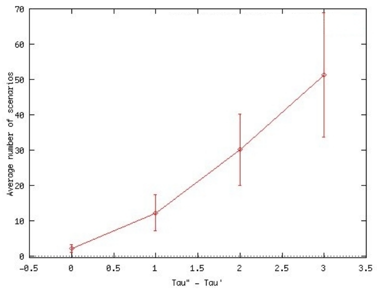

Figure 6. Average number of minimum cost scenarios versus $\tau'' - \tau'$. $\Omega = 300$, $n = 20$. $\lambda = 0.3$

required cycle elimination (Phase II of the algorithm). Although it is possible to construct an example where the algorithm will require this phase, it appears that these scenarios are extremely rare or the rate of useless transfers and helpful transfer-loss events is sufficiently high that a scenario with cost $\tau' \leq \tau$ is created. Readers familiar with Hannehalli-Pevzner theory might note that this result is similar to that found for *hurdles* [1].

Figure 6 captures how many minimum and near minimum cost scenarios exists for a gene and species tree trial. Consider a gene and species tree where τ is the actual number of transfers that occurred during the simulation. Let τ' be the minimum cost over all scenarios found by the algorithm. For this graph, only trials where $\tau' \leq \tau$ were used for simplicity. Let k range from 0 to 3. The x-axis of this graph shows the number of scenarios (with cost $\tau'' = \tau' + k$) found on average. For $n = 20$ and $\lambda = 0.3$, we would expect that $\tau \approx 5$ (see Figure 4 (a)). When $k = 0$ ($\tau'' = \tau'$), there exist on average 2.09 scenarios with cost τ'. Approximately half of the time (probability 0.46), the scenario is unique. When $k = 3$, ($\tau'' = \tau' + 3$), the number of scenarios with cost τ'' is 51.32 with variance 34.52. It is surprising that so few scenarios exist given that is extremely easy to construct gene and species trees by hand that have exponentially many minimum cost scenarios. We repeated this experiment with various values of n and λ in such a way that the $\tau : n$ ratio was preserved, however the curve did not change significantly.

6 Conclusions and Open Problems

This paper demonstrates the feasibility of the model and algorithm presented in [5] and provides empirical evidence of the relative frequency of degenerate events. Our experiments suggest that transfer-loss events which cause the algorithm to over-estimate the true number of transfer events occur with small probability. Furthermore, the algorithm provides near-optimal scenarios when λ, the rate parameter for lateral transfers, is low enough as to not cause saturation. For realistic size instances of the problem, it was the case roughly half of the time that the minimum cost scenario was unique. In over $10,000$ trials, we did not find a single example of a scenario that required the cycle elimination phase of the algorithm. This suggests that the first phase of the algorithm is sufficient for realistic data sets and implies a $O(2^{2\tau}n^2)$ running time.

It is important to consider various extensions to this framework. It would be interesting to combine this model with *gene order*-based models such as those for *gene reversals*, since gene order will provide important clues as to where and when lateral transfers have occurred. In collaboration with L. Graur's group, we are now extending our framework to included species trees where the arcs are labelled with time. Lastly, for a fixed n and λ, it seems feasible that one could analytically compute the expected number of lateral transfer events needed for a random species tree and random gene tree. A first step would be to do this for, e.g., a balanced species tree and a random gene tree. An implementation of our algorithm and additional experimental results are available at `www.cs.mcgill.ca/ ~laddar/lattrans`.

References

1. A. Caprara, *J. of Comb. Opt.*, 3:149-18 (1999)
2. C. Delwiche and J. Palmer, *Mol. Biol. Evol.*, 13(6), pp. 873–882 (1996).
3. M. Goodman et. al. *Syst. Zool.*, 28 (1979)
4. R. Guigó et. al. *Molec. Phylogenet. and Evol.*, 6, 2, pp. 189-213 (1996)
5. M. Hallett and J. Lagergren *RECOMB '01*, Montreal, pp. 141-148 (2001)
6. M. Hallett and J. Lagergren Models and Algorithms for Lateral Gene Transfer Problems. Full journal version. Available at `www.mcb.mcgill.ca/~hallett` (2002)
7. J. Hein *J. Mol. Evol.*, 36, pp. 396-405 (1993)
8. J. Hein, T. Jiang, L. Wang, and K. Zhang *Disc. Appl. Math.*, 71, pp. 153-169 (1996)
9. R. D. M. Page and M. A. Charleston *Molecular Phylogenetics and Evolution*, 7, pp. 231-240 (1997)
10. A. von Haseler and G. A. Churchill *J. Mol. Evol.*, 37, pp. 77-85 (1993)