BIOINFORMATICS

Vol. 28 ISMB 2012, pages i283–i291
doi:10.1093/bioinformatics/bts225

# Efficient algorithms for the reconciliation problem with gene duplication, horizontal transfer and loss

Mukul S. Bansal [1,*], Eric J. Alm [2] and Manolis Kellis [1,3,*]

[1]Computer Science and Artificial Intelligence Laboratory, [2]Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA and [3]Broad Institute of MIT and Harvard, Cambridge, MA 02142, USA

**ABSTRACT**

**Motivation:** Gene family evolution is driven by evolutionary events such as speciation, gene duplication, horizontal gene transfer and gene loss, and inferring these events in the evolutionary history of a given gene family is a fundamental problem in comparative and evolutionary genomics with numerous important applications. Solving this problem requires the use of a reconciliation framework, where the input consists of a gene family phylogeny and the corresponding species phylogeny, and the goal is to reconcile the two by postulating speciation, gene duplication, horizontal gene transfer and gene loss events. This reconciliation problem is referred to as duplication-transfer-loss (DTL) reconciliation and has been extensively studied in the literature. Yet, even the fastest existing algorithms for DTL reconciliation are too slow for reconciling large gene families and for use in more sophisticated applications such as gene tree or species tree reconstruction.

**Results:** We present two new algorithms for the DTL reconciliation problem that are dramatically faster than existing algorithms, both asymptotically and in practice. We also extend the standard DTL reconciliation model by considering distance-dependent transfer costs, which allow for more accurate reconciliation and give an efficient algorithm for DTL reconciliation under this extended model. We implemented our new algorithms and demonstrated up to 100 000-fold speed-up over existing methods, using both simulated and biological datasets. This dramatic improvement makes it possible to use DTL reconciliation for performing rigorous evolutionary analyses of large gene families and enables its use in advanced reconciliation-based gene and species tree reconstruction methods.

**Availability:** Our programs can be freely downloaded from `http://compbio.mit.edu/ranger-dtl/`.

**Contact:** mukul@csail.mit.edu; manoli@mit.edu

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 INTRODUCTION

Gene families evolve through complex evolutionary processes such as speciation, gene duplication, horizontal gene transfer and gene loss. Accurate inference of these events is crucial not only to understanding gene and genome evolution but also for reliably inferring orthologs, paralogs, and xenologs (Koonin, 2005; Mi *et al.*, 2010; Sennblad and Lagergren, 2009; Storm and Sonnhammer, 2002; van der Heijden *et al.*, 2007; Vilella *et al.*, 2009; Wapinski *et al.*, 2007); reconstructing ancestral gene content and dating gene birth (Chen *et al.*, 2000; David and Alm, 2011; Ma *et al.*, 2008); accurate gene tree reconstruction (Rasmussen and Kellis, 2011; Vilella *et al.*, 2009); and for whole genome species tree reconstruction (Bansal *et al.*, 2007; Burleigh *et al.*, 2011). Indeed, the problem of inferring gene family evolution has been extensively studied. In the typical formulation of this problem, the goal is to reconcile an input gene tree (gene family phylogeny) to the corresponding rooted species tree by postulating speciation, duplication, transfer and loss events. Much of the previous work in gene tree–species tree reconciliation has focused on either duplication loss (DL) (Bonizzoni *et al.*, 2005; Chauve *et al.*, 2008; Durand *et al.*, 2006; Eulenstein and Vingron, 1998; Goodman *et al.*, 1979; Górecki and Tiuryn, 2006; Mirkin *et al.*, 1995; Page, 1994) or transfer loss (TL) (Boc *et al.*, 2010; Hallett and Lagergren, 2001; Hill *et al.*, 2010; Huelsenbeck *et al.*, 2000; Jin *et al.*, 2009; Nakhleh *et al.*, 2004, 2005; Ronquist, 1995), but not on duplication, transfer and loss together. However, duplication and transfer events frequently occur together, particularly in prokaryotes, and the analysis of such families requires reconciliation methods that can simultaneously consider duplication, transfer and loss events. This problem of gene tree–species tree reconciliation by duplication, transfer and loss simultaneously is referred to as the duplication TL *(DTL) reconciliation* problem.

*Previous work*. The DTL reconciliation problem has a long history and is well studied in the literature. This is partly due to its close association with the host–parasite cophylogeny problem (Charleston and Perkins, 2006) which seeks to understand the evolution of parasites (analogous to genes) within hosts (analogous to species). Almost all known formulations of the DTL reconciliation problem are based on a parsimony framework (Charleston, 1998; Conow *et al.*, 2010; David and Alm, 2011; Doyon *et al.*, 2010; Gorbunov and Liubetskii, 2009; Libeskind-Hadas and Charleston, 2009; Merkle and Middendorf, 2005; Merkle *et al.*, 2010; Ovadia *et al.*, 2011; Ronquist, 2003; Tofigh *et al.*, 2011) (but see also Tofigh (2009) for an example of a probabilistic formulation, as well as Csürös and Miklós (2006) for a probabilistic framework based on gene content). Under this framework, each duplication, transfer and loss event is assigned a cost and the goal is to find a reconciliation that has the lowest total reconciliation cost. Optimal DTL reconciliations can sometimes violate temporal constraints; that is, the transfers are such that they induce contradictory constraints on the dates for the internal nodes of the species tree. We refer to such paradoxical reconciliations as *time-inconsistent* (after Doyon *et al.*, 2010). In general, it is desirable to consider only those DTL reconciliations that are *time-consistent* (i.e. paradox-free). Henceforth, we refer to the problem of specifically computing optimal time-consistent

*To whom correspondence should be addressed.

DTL reconciliations as the *tcDTL Reconciliation* problem. Although the DTL reconciliation problem can be solved in polynomial time, solving the tcDTL reconciliation problem is NP-hard (Ovadia *et al.*, 2011; Tofigh *et al.*, 2011). If divergence time information is available for the nodes of the species tree (or if there is a known relative temporal ordering for each pair of internal nodes), then any proposed DTL reconciliation must also respect the temporal constraints imposed by the available timing information, i.e., transfers must be restricted to occur only between coexisting species. When such divergence timing information is available, even the tcDTL reconciliation problem becomes polynomially solvable (Libeskind-Hadas and Charleston, 2009). (Note, however, that time-consistency cannot be guaranteed just by ensuring that transfers occur between coexisting species). In general, the input species tree can be undated, partially dated, or fully dated depending on whether the divergence timing information associated with the nodes of the species tree is absent, partial, or complete, respectively. Thus, in practice, when the species tree is undated or partially dated, one solves the DTL reconciliation problem, and if the species tree is fully dated, one can solve either the DTL reconciliation or the tcDTL reconciliation problem.

Let $m$ denote the number of leaves in the input gene tree and $n$ the number of leaves in the species tree. Both the DTL reconciliation problem and the tcDTL reconciliation problem, along with some of their variants, have been extensively studied in the literature (Charleston, 1998; Conow *et al.*, 2010; David and Alm, 2011; Doyon *et al.*, 2010; Gorbunov and Liubetskii, 2009; Libeskind-Hadas and Charleston, 2009; Merkle and Middendorf, 2005; Merkle *et al.*, 2010; Ronquist, 2003; Tofigh, 2009; Tofigh *et al.*, 2011). The most recent algorithmic work on these problems includes Doyon *et al.* (2010); Tofigh (2009); Tofigh *et al.* (2011); and David and Alm, 2011. The paper by Tofigh *et al.* (2011) studies a restricted version of the reconciliation model that ignores losses (equivalent to assigning a cost of zero for loss events under the DTL reconciliation problem) and shows that, under this restricted model, the DTL reconciliation problem on undated trees can be solved in $O(mn^2)$ time. They also gave a fixed parameter tractable algorithm for enumerating all most parsimonious reconciliations. The time complexity of the $O(mn^2)$-time algorithm was further improved to $O(mn)$ in Tofigh (2009) (under the same restricted reconciliation model). However, with the increasing availability of whole-genome datasets, such a restriction on the reconciliation model can be problematic as losses are a rich source of information that can be critical for accurate reconciliation. Indeed, losses play a fundamental role in the ability to distinguish between duplications and transfers as well as in mapping the nodes of the gene tree into the nodes of the species tree, and thus should be explicitly considered during reconciliation. The paper by Doyon *et al.* (2010) showed that, for fully dated species trees, the tcDTL reconciliation problem could be solved in $O(mn^2)$ time. Recently, an $O(mn^2)$-time algorithm for the tcDTL reconciliation problem on fully dated trees has also been independently developed for Version 2 of the program Jane (Conow *et al.*, 2010). Finally, the recent paper by David and Alm (2011) gave an $O(mn^2)$-time algorithm for the DTL reconciliation problem on undated trees.

In summary, in spite of tremendous methodological and algorithmic advances, even the fastest existing algorithms for DTL reconciliation (David and Alm, 2011; Merkle *et al.*, 2010) as well as for tcDTL reconciliation on fully dated trees (Doyon *et al.*, 2010) still have a time complexity of $\Theta(mn^2)$. This makes them too slow to reconcile trees with more than a few hundred taxa, and completely unsuitable for all but the smallest trees when used in sophisticated applications such as reconciliation-based gene tree or species tree reconstruction that require the reconciliation of a multitude of trees while searching through tree space (Bansal *et al.*, 2007; Burleigh *et al.*, 2011; Rasmussen and Kellis, 2011; Vilella *et al.*, 2009).

*Our contributions.* Recall that the DTL reconciliation problem, even on fully dated species trees, does not guarantee that the optimal reconciliation is time-consistent, whereas the tcDTL reconciliation problem does. However, the tcDTL reconciliation problem suffers from two major drawbacks that limit its applicability in practice. First, the tcDTL reconciliation problem can only be solved efficiently when the species tree is fully dated. This limits its application to only those species tree that contain a relatively small number of taxa (say <100). This is because, it can be extremely difficult to accurately date large species trees (Rutschmann, 2006) and the accuracy of tcDTL reconciliation relies implicitly on having a correctly dated species tree. Second, the time complexity of the fastest known algorithm for the tcDTL reconciliation problem is $O(mn^2)$, which makes it too slow to be used with large datasets (as we also demonstrate later). This also makes it too slow for reconciliation-based gene tree reconstruction of even relatively small gene trees, as it involves repeatedly reconciling a multitude of candidate gene trees against the species tree. Furthermore, the tcDTL reconciliation problem cannot be used for reconciliation-based whole-genome species tree construction (also called gene tree parsimony), as the topology of the species tree is repeatedly modified and so at each step, the species tree is undated.

Thus, in this work, we focus on the DTL reconciliation problem. In particular, we improve upon the current state of the art for the DTL reconciliation problem in the following ways:

(1) We provide an $O(mn)$-time algorithm for the DTL reconciliation problem on undated species trees. This improves on the fastest known algorithm for this problem by a factor of $n$. The DTL reconciliation problem on undated trees is the most common version of the DTL reconciliation problem and arises whenever the species tree cannot be accurately dated, as is usually the case with large gene families, and in applications such as reconciliation-based species tree reconstruction.

(2) For the DTL reconciliation problem on fully dated species trees, we provide an $O(mn\log n)$-time algorithm, which improves on the fastest known algorithm for this problem by a factor of $n/\log n$. Even though the fully dated version of DTL reconciliation does not guarantee time-consistency, as we show later using thorough experimental studies, optimal DTL reconciliations closely approximate optimal tcDTL reconciliations. This algorithm is thus meant as a faster alternative to the $O(mn^2)$-time algorithm for tcDTL reconciliation.

(3) We give a simple $O(mn^2)$-time algorithm for DTL reconciliation that can handle distance-dependent transfer costs and can work with undated, partially dated, or fully dated species trees. This is a factor of $n$ faster than the fastest known algorithm that can handle distance-dependent transfer costs (Conow *et al.*, 2010). Distance-dependent transfer costs capture the biology of transfers more accurately than having

a fixed transfer cost (Andam and Gogarten, 2011), and its use may lead to more accurate DTL reconciliations.

In addition, we also discuss how to efficiently incorporate other enhancements such as detecting transfers from unsampled or extinct lineages that further improve the accuracy of DTL reconciliation. Our $O(mn)$-time algorithm for undated species trees builds on the $O(mn)$-time algorithm from Tofigh (2009) that computes optimal reconciliation scenarios under a simpler reconciliation cost that ignores losses. Specifically, we show how to augment that algorithm to efficiently keep track of losses as well. Fully dated species trees presented a greater algorithmic challenge and to obtain our fast $O(mn\log n)$-time algorithm, we developed a novel algorithmic framework that exploits the structure of fully dated species trees and makes use of recent algorithmic advances on the dynamic range minimum query problem (Brodal *et al.*, 2011).

Our new algorithms and other enhancements represent a great improvement in the runtime and applicability of DTL reconciliation compared with extant approaches. They not only make it possible to analyze large gene families but also to quickly analyze thousands of gene families from across the entire genomes of the species under consideration. Furthermore, and perhaps most importantly, they make DTL reconciliation much more amenable for use in sophisticated applications such as reconciliation-based gene tree or species tree reconstruction. The ability to efficiently handle distance-dependent transfer costs, as well as the other enhancements, in turn, makes it possible to reconstruct the evolutionary history of gene families even more accurately. We benchmark our algorithms to both simulated and biological datasets and demonstrate the dramatic improvements in runtime at a range of dataset sizes. We also assess the accuracy of DTL reconciliation, on both dated and undated species trees, compared with optimal tcDTL reconciliations on fully dated trees and demonstrate the utility of using distance-dependent transfer costs in the reconciliation model. In the interest of brevity, all proofs appear in the Supplementary Material (Section S.1).

## 2 DEFINITIONS AND PRELIMINARIES

Given a tree $T$, we denote its node, edge and leaf sets by $V(T)$, $E(T)$ and $\mathrm{Le}(T)$, respectively. If $T$ is rooted, the root node of $T$ is denoted by $\mathrm{rt}(T)$, the parent of a node $v \in V(T)$ by $\mathrm{pa}_T(v)$, its set of children by $\mathrm{Ch}_T(v)$, and the (maximal) subtree of $T$ rooted at $v$ by $T(v)$. If two nodes in $T$ have the same parent, they are called *siblings*. The set of *internal nodes* of $T$, denoted $I(T)$, is defined to be $V(T)\setminus \mathrm{Le}(T)$. We define $\leq_T$ to be the partial order on $V(T)$, where $x \leq_T y$ if $y$ is a node on the path between $\mathrm{rt}(T)$ and $x$. The partial order $\geq_T$ is defined analogously, i.e., $x \geq_T y$ if $x$ is a node on the path between $\mathrm{rt}(T)$ and $x$. We say that $v$ is an *ancestor* of $u$, or that $u$ is a *descendant* of $v$, if $u \leq_T v$ (note that, under this definition, every node is a descendant as well as ancestor of itself). We say that $x$ and $y$ are *incomparable* if neither $u \leq_T v$ nor $v \leq_T u$. Given a non-empty subset $L \subseteq \mathrm{Le}(T)$, we denote by $\mathrm{lca}_T(L)$, the least common ancestor (LCA) of all the leaves in $L$ in tree $T$; that is, $\mathrm{lca}_T(L)$ is the unique smallest upper bound of $L$ under $\leq_T$. Given $x, y \in V(T)$, $x \to_T y$ denotes the unique path from $x$ to $y$ in $T$. We denote by $d_T(x,y)$ the number of edges on the path $x \to_T y$. Throughout this work, unless otherwise stated, the term tree refers to a rooted binary tree.

A *species tree* is a tree that depicts the evolutionary relationships of a set of species. Given a gene family from a set of species, a *gene tree* is a tree that depicts the evolutionary relationships among the sequences encoding only that gene family in the given set of species. Thus, the nodes in a gene tree represent genes. We assume that each leaf of the gene trees is labeled with the species from which that gene was sampled. This labeling defines a *leaf-mapping* $\mathcal{L}_{G,S}: \mathrm{Le}(G) \to \mathrm{Le}(S)$ that maps a leaf node $g \in \mathrm{Le}(G)$ to that unique leaf node $s \in \mathrm{Le}(S)$ which has the same label as $g$. Note that gene trees may have more than one gene sampled from the same species. Throughout this work, we denote the gene tree and species tree under consideration by $G$ and $S$, respectively, and will implicitly assume that $\mathcal{L}_{G,S}(g)$ is well defined.

### 2.1 Reconciliation and DTL scenarios

Reconciling a gene tree with a species tree involves mapping the gene tree into the species tree. Such a mapping allows us to infer the evolutionary events that gave rise to that particular gene tree. In this case, the evolutionary events of interest are speciation, gene duplication, horizontal gene transfer and gene loss. Next, we define what constitutes a valid reconciliation; specifically, we define a DTL scenario (Tofigh *et al.*, 2011) for $G$ and $S$ that characterizes the mappings of $G$ into $S$ that constitute a biologically valid reconciliation. Essentially, DTL scenarios map each gene tree node to a unique species tree node in a consistent way that respects the immediate temporal constraints implied by the species tree and designate each gene tree node as representing a speciation, duplication or transfer event. For any gene tree node, say $g$, that represents a transfer event, DTL scenarios also specify which of the two edges $(g,g')$ or $(g,g'')$, where $g'$ and $g''$ denote the children of $g$, represents the transfer edge on $S$, and identify the recipient species of the corresponding transfer.

*Incorporating available divergence time information.* When accurate divergence time information is available, for some or all of the nodes of the species tree, DTL scenarios must respect the temporal constraints imposed by the available timing information. Specifically, those transfer events that are inconsistent with the available timing information are disallowed (as transfer events could only have happened between two coexisting species). If there is no divergence time information available, then transfers are allowed to occur between any pair of incomparable species on the species tree.

The definition of a DTL scenario below is a generalization of the definition from Tofigh *et al.* (2011). The generalization is necessary to correctly handle optimal reconciliations in cases when the species tree is dated. Specifically, we enforce that, if the species tree is dated, then transfers can only occur between coexisting species and introduce an additional variable to explicitly specify the recipient species for any transfer event.

DEFINITION 2.1 (DTL scenario). A DTL scenario for $G$ and $S$ is a seven-tuple $\langle \mathcal{L}, \mathcal{M}, \Sigma, \Delta, \Theta, \Xi, \tau \rangle$, where $\mathcal{L}: \mathrm{Le}(G) \to \mathrm{Le}(S)$ represents the leaf mapping from $G$ to $S$, $\mathcal{M}: V(G) \to V(S)$ maps each node of $G$ to a node of $S$, the sets $\Sigma$, $\Delta$ and $\Theta$ partition $I(G)$ into speciation, duplication and transfer nodes, respectively, $\Xi$ is a subset of gene tree edges that represent transfer edges, and $\tau: \Theta \to V(S)$ specifies the recipient species for each transfer event, subject to the following constraints:

(1) If $g \in \mathrm{Le}(G)$, then $\mathcal{M}(g) = \mathcal{L}(g)$.

(2) If $g \in I(G)$, and $g'$ and $g''$ denote the children of $g$, then,

(a) $\mathcal{M}(g) \not\leq_S \mathcal{M}(g')$ and $\mathcal{M}(g) \not\leq_S \mathcal{M}(g'')$.

(b) At least one of $\mathcal{M}(g')$ and $\mathcal{M}(g'')$ is a descendant of $\mathcal{M}(g)$.

(3) Given any edge $(g,g') \in E(G)$, $(g,g') \in \Xi$ if and only if $\mathcal{M}(g)$ and $\mathcal{M}(g')$ are incomparable.

(4) If $g \in I(G)$ and $g'$ and $g''$ denote the children of $g$, then,

(a) $g \in \Sigma$ only if $\mathcal{M}(g) = \text{lca}(\mathcal{M}(g'), \mathcal{M}(g''))$ and $\mathcal{M}(g')$ and $\mathcal{M}(g'')$ are incomparable,

(b) $g \in \Delta$ only if $\mathcal{M}(g) \geq_S \text{lca}(\mathcal{M}(g'), \mathcal{M}(g''))$,

(c) $g \in \Theta$ if and only if either $(g,g') \in \Xi$ or $(g,g'') \in \Xi$,

(d) If $g \in \Theta$ and $(g,g') \in \Xi$, then $\mathcal{M}(g)$ and $\tau(g)$ must be incomparable, the species represented by them must be potentially coexisting with respect to the available divergence time estimates, and $\mathcal{M}(g')$ must be a descendant of $\tau(g)$, i.e. $\mathcal{M}(g') \leq_S \tau(g)$.

Constraint 1 above ensures that the mapping $\mathcal{M}$ is consistent with the leaf mapping $\mathcal{L}$. Constraint 2(a) imposes on $\mathcal{M}$ the temporal constraints implied by $S$. Constraint 2(b) implies that any internal node in $G$ may represent at most one transfer event. Constraint 3 determines the edges of $G$ that are transfer edges. Constraints 4(a–c) state the conditions under which an internal node of $G$ may represent a speciation, duplication and transfer, respectively. Constraint 4(d) specifies which species may be designated as the recipient species for any given transfer event.

DTL scenarios correspond naturally to reconciliations and it is straightforward to infer the reconciliation of $G$ and $S$ implied by any DTL scenario. Figure 1 shows two simple DTL scenarios. Given a DTL scenario, one can directly count the minimum number of gene losses in the corresponding reconciliation as follows:

DEFINITION 2.2 (Losses). Given a DTL scenario $\alpha = \langle \mathcal{L}, \mathcal{M}, \Sigma, \Delta, \Theta, \Xi, \tau \rangle$ for $G$ and $S$, let $g \in V(G)$ and $\{g', g''\} = \text{Ch}(g)$. The number of losses $\text{Loss}_\alpha(g)$ at node $g$ is defined to be

- $|d_S(\mathcal{M}(g), \mathcal{M}(g')) - 1| + |d_S(\mathcal{M}(g), \mathcal{M}(g'')) - 1|$, if $g \in \Sigma$
- $d_S(\mathcal{M}(g), \mathcal{M}(g'))$, if $g \in \Delta$ and $\mathcal{M}(g) = \mathcal{M}(g'')$.
- $d_S(\mathcal{M}(g), \mathcal{M}(g')) + d_S(\mathcal{M}(g), \mathcal{M}(g''))$, if $g \in \Delta$, $\mathcal{M}(g) \neq \mathcal{M}(g')$, and $\mathcal{M}(g) \neq \mathcal{M}(g'')$, and
- $d_S(\mathcal{M}(g), \mathcal{M}(g'')) + d_S(\tau(g), \mathcal{M}(g'))$ if $(g,g') \in \Xi$.

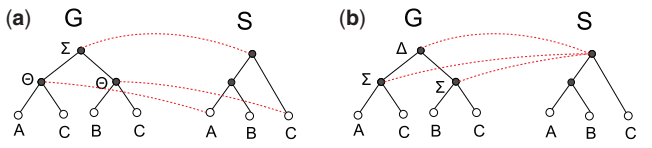


**Fig. 1. Simple DTL scenarios.** (**a**) and (**b**) depict two possible reconciliations of $G$ and $S$: the dotted arcs show the mapping $\mathcal{M}$ (with the leaf mapping being specified by the leaf labels on the gene tree), and the label at each internal node of $G$ specifies the type of event represented by that node. The reconciliation in (a) requires two transfers and one loss and the one in (b) requires one duplication and two losses

We define the total number of losses in the reconciliation corresponding to the DTL scenario $\alpha$ to be $\text{Loss}_\alpha = \sum_{g \in I(G)} \text{Loss}_\alpha(g)$.

Let $P_\Delta, P_\Theta$ and $P_{\text{loss}}$ denote the costs associated with duplication, transfer and loss events respectively. The cost of reconciling $G$ and $S$ according to a DTL scenario $\alpha$ is defined as follows.

DEFINITION 2.3 (Reconciliation cost of a DTL scenario). *Given a DTL scenario* $\alpha = \langle \mathcal{L}, \mathcal{M}, \Sigma, \Delta, \Theta, \Xi, \tau \rangle$ *for* $G$ *and* $S$, *the* reconciliation cost *associated with* $\alpha$ *is given by* $\mathcal{R}_\alpha = P_\Delta \cdot |\Delta| + P_\Theta \cdot |\Theta| + P_{loss} \cdot Loss_\alpha$.

Given $G$ and $S$, our goal is to find a most parsimonious reconciliation of $G$ and $S$. More formally.

PROBLEM 1. [Most parsimonious reconciliation (MPR)] *Given* $G$ *and* $S$, *the MPR problem is to find a DTL scenario for* $G$ *and* $S$ *with minimum reconciliation cost.*

Based on whether the species tree is undated or fully dated, we distinguish two versions of the MPR problem: (i) The *undated MPR (U-MPR)* problem where the species tree is undated and (ii) the *fully dated MPR (D-MPR)* problem where every node of the species tree has an associated divergence time estimate (or there is a known total order on the internal nodes of the species tree). We will exploit the local structure unique to each version to develop faster algorithms for them.

## 3 COMPUTING THE MOST PARSIMONIOUS RECONCILIATION

In this section, we first develop our fast algorithms for the U-MPR and D-MPR problems and then give a simple $O(mn^2)$-time algorithm for the (general) MPR problem that can efficiently handle distance-dependent transfer costs. Before we proceed, we need a few definitions and additional notation.

***Definitions***: Given any $g \in I(G)$ and $s \in V(S)$, let $c_\Sigma(g,s)$ denote the cost of an optimal reconciliation of $G(g)$ with $S$ such that $g$ maps to $s$ and $g \in \Sigma$. The terms $c_\Delta(g,s)$ and $c_\Theta(g,s)$ are defined similarly for $g \in \Delta$ and $g \in \Theta$, respectively. Given any $g \in V(G)$ and $s \in V(S)$, we define $c(g,s)$ to be the cost of an optimal reconciliation of $G(g)$ with $S$ such that $g$ maps to $s$. Thus,

$$c(g,s) = \begin{cases} 0 & \text{if } g \in \text{Le}(G) \text{ and } s = \mathcal{M}(g), \\ \infty & \text{if } g \in \text{Le}(G) \text{ and } s \neq \mathcal{M}(g), \\ \min\{c_\Sigma(g,s), c_\Delta(g,s), c_\Theta(g,s)\} & \text{otherwise.} \end{cases}$$

Furthermore, let $\text{in}(g,s) = \min_{x \in V(S(s))}\{P_{\text{loss}} \cdot d_S(s,x) + c(g,x)\}$, $\text{out}(g,s) = \min_{x \in V(S) \text{ incomparable to } s} c(g,x)$, and $\text{inAlt}(g,s) = \min_{x \in V(S(s))} c(g,x)$. In other words, $\text{inAlt}(g,s)$ is the cost of an optimal reconciliation of $G(g)$ with $S$ such that $g$ may map to any node in $V(S(s))$; $\text{out}(g,s)$ is the cost of an optimal reconciliation of $G(g)$ with $S$ such that $g$ may map to any node from $V(S)$ that is incomparable to $s$; and $\text{in}(g,s)$ is the cost of an optimal reconciliation of $G(g)$ with $S$ such that $g$ may map to any node, say $x$, in $V(S(s))$ but with an additional reconciliation cost of one loss event for each edge on the path from $s$ to $x$.

Note that the optimal reconciliation cost of $G$ and $S$ is simply: $\min_{s \in V(S)} c(\text{rt}(G), s)$. The equation for $c(g,s)$ above, used in a dynamic programming framework and coupled with methods for computing the values of $c_\Sigma(g,s)$, $c_\Delta(g,s)$ and $c_\Theta(g,s)$, forms the basis of all our algorithms.

### 3.1 An $O(mn)$-time algorithm for U-MPR

The following algorithm solves the U-MPR problem in $O(mn)$ time. Our algorithm builds on the $O(mn)$-time dynamic programming algorithm from Tofigh (2009) that computes optimal reconciliation scenarios under a simpler reconciliation cost that ignores losses. We compute the values $c_\Sigma(g,s)$, $c_\Delta(g,s)$ and $c_\Theta(g,s)$ for each $g \in V(G)$ and $s \in V(S)$ by performing a nested post-order traversal of $G$ and $S$. For efficiency, we save and reuse as much of the computation from previous steps as possible, and the values $in(\cdot,\cdot)$, $\text{inAlt}(\cdot,\cdot)$ and $\text{out}(\cdot,\cdot)$ help us in efficiently computing the values $c_\Sigma(g,s)$, $c_\Delta(g,s)$, and $c_\Theta(g,s)$ at each dynamic programming step. For instance, for any $g \in I(G)$, the value of $c_\Sigma(g,s)$ is simply: $\infty$ if $s \in \text{Le}(S)$, and $\min\{\text{in}(g',s')+\text{in}(g'',s''),\ \text{in}(g'',s')+\text{in}(g',s'')\}$, where $\{g',g''\}=\text{Ch}_G(g)$ and $\{s',s''\}=\text{Ch}_S(s)$, if $s \in I(S)$. The values of $c_\Delta(g,s)$ and $c_\Theta(g,s)$ can be similarly computed; see Steps 10 and 18 of Algorithm *U-Reconcile* for $c_\Delta(g,s)$ and Steps 11 and 19 for $c_\Theta(g,s)$. The nested post-order traversal ensures that when computing the values $c_\Sigma(g,s)$, $c_\Delta(g,s)$ and $c_\Theta(g,s)$ at nodes $g \in G$ and $s \in S$, all the required $\text{in}(\cdot,\cdot)$, $\text{inAlt}(\cdot,\cdot)$, $\text{out}(\cdot,\cdot)$ and $c(\cdot,\cdot)$ values have already been computed.

**Algorithm** U-Reconcile$(G,S,\mathcal{L})$

1. **for** each $g \in V(G)$ and $s \in V(S)$ **do**
2. Initialize $c(g,s)$, $c_\Sigma(g,s)$, $c_\Delta(g,s)$, $c_\Theta(g,s)$, $\text{in}(g,s)$, $\text{inAlt}(g,s)$, and $\text{out}(g,s)$ to $\infty$.
3. **for** each $g \in \text{Le}(G)$ **do**
4. Initialize $c(g,\mathcal{L}(g))$ to 0, and, for each $s \geq_S \mathcal{L}(g)$, initialize $\text{in}(g,s)$ to $P_{\text{loss}} \cdot d_S(s,\mathcal{L}(g))$ and $\text{inAlt}(g,s)$ to 0.
5. **for** each $g \in I(G)$ in post-order **do**
6. **for** each $s \in V(S)$ in post-order **do**
7. Let $\{g',g''\}=\text{Ch}_G(g)$.
8. **if** $s \in \text{Le}(S)$ **then**
9. $c_\Sigma(g,s)=\infty$.
10. $c_\Delta(g,s)=P_\Delta+c(g',s)+c(g'',s)$.
11. If $s \neq \text{rt}(S)$, then $c_\Theta(g,s)=P_\Theta+\min\{\text{in}(g',s)+\text{out}(g'',s),\ \text{in}(g'',s)+\text{out}(g',s)\}$.
12. $c(g,s)=\min\{c_\Sigma(g,s),c_\Delta(g,s),c_\Theta(g,s)\}$.
13. $\text{in}(g,s)=c(g,s)$.
14. $\text{inAlt}(g,s)=c(g,s)$.
15. **else**
16. Let $\{s',s''\}=\text{Ch}_S(s)$.
17. $c_\Sigma(g,s)=\min\{\text{in}(g',s')+\text{in}(g'',s''),\text{in}(g'',s')+\text{in}(g',s'')\}$.
18. $$c_\Delta(g,s)=P_\Delta+\min\begin{cases} c(g',s)+\text{in}(g'',s'')+P_{\text{loss}},\\ c(g',s)+\text{in}(g'',s')+P_{\text{loss}},\\ c(g'',s)+\text{in}(g',s'')+P_{\text{loss}},\\ c(g'',s)+\text{in}(g',s')+P_{\text{loss}},\\ c(g',s)+c(g'',s),\\ \text{in}(g',s')+\text{in}(g'',s'')+2P_{\text{loss}},\\ \text{in}(g',s'')+\text{in}(g'',s')+2P_{\text{loss}},\\ \text{in}(g',s')+\text{in}(g'',s')+2P_{\text{loss}},\\ \text{in}(g',s'')+\text{in}(g'',s'')+2P_{\text{loss}}. \end{cases}$$
19. If $s \neq \text{rt}(S)$, then $c_\Theta(g,s)=P_\Theta+\min\{\text{in}(g',s)+\text{out}(g'',s),\ \text{in}(g'',s)+\text{out}(g',s)\}$.
20. $c(g,s)=\min\{c_\Sigma(g,s),c_\Delta(g,s),c_\Theta(g,s)\}$.
21. $\text{in}(g,s)=\min\{c(g,s),\text{in}(g,s')+P_{\text{loss}},\text{in}(g,s'')+P_{\text{loss}}\}$.
22. $\text{inAlt}(g,s)=\min\{c(g,s),\text{inAlt}(g,s'),\text{inAlt}(g,s'')\}$.
23. **for** each $s \in I(S)$ in pre-order **do**
24. Let $\{s',s''\}=\text{Ch}_S(s)$.
25. $\text{out}(g,s')=\min\{\text{out}(g,s),\text{inAlt}(g,s'')\}$, and $\text{out}(g,s'')=\min\{\text{out}(g,s),\text{inAlt}(g,s')\}$.
26. Return $\min_{s \in V(S)} c(\text{rt}(G),s)$.

***Remarks:*** (i) Note that, while the above algorithm only outputs the optimal reconciliation cost, it can be easily adapted, without affecting its time complexity, to output the DTL scenario itself. (ii) The algorithm above implicitly assumes that if $g \in I(G)$ is a transfer node such that $(g,g') \in \Xi$, then $\tau(g)=\mathcal{M}(g')$. The reason for this is easy to see: any reconciliation in which $\tau(g)$ is not $\mathcal{M}(g)$ (and losses have a strictly positive cost), cannot be most parsimonious. This, however, only holds true for the U-MPR problem, and we will be unable to make this assumption when working with partially or fully dated species trees.

We have the following theorem. (all proofs are available in the Supplementary Material).

THEOREM 3.1. The U-MPR problem on $G$ and $S$ can be solved in $O(mn)$ time.

### 3.2 An $O(mn\log n)$-time algorithm for D-MPR

In the D-MPR problem, there exists a total ordering of the internal nodes of the species tree based on their divergence times. Thus, in this setting, for any given pair of species tree edges, it is known whether the two species represented by those edges overlapped in their time of existence, and transfers are only allowed between two species if they are coexisting.

We assign consecutive positive integers, starting with one, to the internal nodes of the species tree according to the total order. These numbers are referred to as time stamps and they represent the temporal order in which the species represented by these nodes diverged. Given a node $s \in V(S)$, we denote its time stamp by $t(s)$. If the largest time stamp assigned to the internal nodes is $k$, then we assign time stamp $k+1$ to each leaf of $S$. Any two consecutive time stamps $x, x+1$ define the *time zone* labeled $x$ on $S$.

Given a node $s \in V(S) \setminus \text{rt}(S)$, the species represented by that node exists along the edge $(\text{pa}(s),s)$ and is consequently associated with the time stamp interval $[t(\text{pa}(s)),t(s)]$ and the time zones $t(\text{pa}(s)),\ldots,t(s)-1$. Observe that any edge from $E(S)$ is associated with at least one time zone. Given any pair of nodes $s,s' \in V(S) \setminus \text{rt}(S)$, a transfer is allowed between the species represented by those nodes if and only if the two edges $(\text{pa}(s),s)$ and $(\text{pa}(s'),s')$ overlap in at least one time zone.

Our algorithm for the D-MPR problem, called Algorithm *D-reconcile*, makes use of the same overall dynamic programming structure as Algorithm *U-Reconcile*, and the procedure for computing the values $c_\Sigma(\cdot,\cdot)$ and $c_\Delta(\cdot,\cdot)$ remains identical. The difference is in the way $c_\Theta(\cdot,\cdot)$ is computed, as we can no longer rely on the $out(\cdot,\cdot)$ values. Instead, we need a more elaborate procedure that can efficiently yield the 'best receiver' for a transfer originating at the species tree node currently under consideration, from among the relevant time zones. More concretely, suppose we want to compute the value $c_\Theta(g,s)$ assuming that $(g,g') \in \Xi$, where $g' \in Ch(g)$, for each $s \in V(S)$. Our algorithm first efficiently computes the locally best and locally second-best receivers of gene $g$ in each time zone based on the values $c(g',\cdot)$. Then, for each candidate node $s$ under consideration, we efficiently compute the best receiver, for a transfer originating at $s$, by choosing the globally

optimal value from among the previously computed locally best and locally second-best receivers for the relevant time zones. For efficiency, our algorithm makes use of (i) a binomial heap data structure and (ii) a dynamic range minimum query data structure. The binomial heap data structure maintains a set of $P$-values while supporting find-min, insert and delete operations in O(1), $O(\log p)$ and $O(\log p)$ time, respectively (Cormen *et al.*, 2009; Vuillemin, 1978). The dynamic range minimum query data structure maintains an ordered list of numbers and can answer queries that seek the smallest element in a given query range in $O(\log p)$ time and also supports update operations that change the value of an element in the list in $O(\log p)$ time (Brodal *et al.*, 2011).

*Definitions.* Let $k$ denote the number of time zones on the species tree. Given a time zone $i$ ($1 \leq i \leq k$), let $Z(i)$ denote the set of edges from $E(S)$ that are associated with time zone $i$. Let Best$(g,i)$ and secondBest$(g,i)$ denote, respectively, the two edges from $Z(i)$ with the smallest value of in$(g,\cdot)$.

*Preprocessing.* Before running Algorithm D-Reconcile, we assume that we have precomputed, for each time zone $i$ ($1 \leq i \leq k$), the following: (i) the set of edges $(\text{pa}(s),s) \in E(S)$ for which $t(s)-1=i$ (i.e. $(\text{pa}(s),s)$ is associated with $Z(i)$, but not with $Z(i+1)$), referred to as end$(i)$ and (ii) the set of edges $(\text{pa}(s),s) \in E(S)$ for which $t(\text{pa}(s))=i$ (i.e. $(\text{pa}(s),s)$ is associated with $Z(i)$, but not with $Z(i-1)$), referred to as begin$(i)$.

The algorithm below makes use of the procedure bestReceiver which takes as input a node $g \in I(G)$, a child $x$ of $g$, and an edge $s$ from $S$ and returns, from among all those edges that share at least one time zone with $s$, an edge $(\text{pa}(y),y)$ for which the value in$(g,y)$ is smallest. Essentially, the returned edge $(\text{pa}(y),y)$ implies that, in a scenario where $g$ maps to $s$ and $g$ is a transfer node with $(g,x) \in \Xi$, the best possible mapping for $x$ (i.e. one for which $c_\Theta(g,s)$ is minimized) is $y$.

**Algorithm** $D-$Reconcile$(G,S,\mathcal{L})$

1. Let $k$ denote the number of time zones on $S$.
2. **for** each $g \in V(G)$ and $s \in V(S)$ **do**
3. Initialize $c(g,s)$, $c_\Sigma(g,s)$, $c_\Delta(g,s)$, $c_\Theta(g,s)$ and in$(g,s)$ to $\infty$.
4. **for** each $g \in \text{Le}(G)$ **do**
5. Initialize $c(g,\mathcal{L}(g))$ to 0, and, for each $s \geq_S \mathcal{L}(g)$, initialize in$(g,s)$ to $P_{\text{loss}} \cdot d_S(s,\mathcal{L}(g))$.
6. **for** each $g \in I(G)$ in post-order **do**
7. Let $\{g',g''\} = \text{Ch}_G(g)$.
8. **for** each $x \in \{g',g''\}$ **do**
9. Create an empty binomial heap data structure $\mathcal{H}$.
10. Consider each edge $(\text{pa}(y),y)$ from $Z(k)$ and add it to $\mathcal{H}$ based on the value in$(x,y)$.
11. Query the heap $\mathcal{H}$ to assign Best$(x,k)$ and secondBest$(x,k)$.
12. **for** each time zone $i$ in decreasing order from $k-1$ to 1 **do**
13. Update the heap $\mathcal{H}$ by deleting from it all the edges in begin$(i+1)$ and inserting all the edges in *end*$(i)$ (according to their in$(x,\cdot)$ scores).
14. Query the heap $\mathcal{H}$ to assign Best$(x,i)$ and secondBest$(x,i)$.
15. Add all the edges Best$(x,\cdot)$ and secondBest$(x,\cdot)$, labeled by their $c(x,\cdot)$ scores, to a dynamic range minimum query data structure, indexed by their time zones (Note that, as stated, each index gets assigned two values, which makes for an ill-defined range minimum query data structure. However, this is easy to get around by assigning Best$(x,i)$ to index $2i-1$, and secondBest$(x,i)$ to index $2i$, and querying the data structure accordingly). We denote this data structure by $\Phi_x$.
16. Delete the heap $\mathcal{H}$.
17. **for** each $s \in V(S)$ in post-order **do**
18. If $s \neq \text{rt}(S)$, then let $(\text{pa}(u),u) = \text{bestReceiver}(g,g',s)$, and $(\text{pa}(v),v) = \text{bestReceiver}(g,g'',s)$.
19. This part of the algorithm is identical to Steps 8 through 22 of Algorithm *U-Reconcile*, except,
    (a) Steps 11 and 19 are replaced by the following:
    If $s \neq \text{rt}(S)$, then $c_\Theta(g,s) = P_\Theta + \min\{\text{in}(g',s) + c(g'',v),\ \text{in}(g'',s) + c(g',u)\}$, and,
    (b) Steps 14 and 22 are removed.
20. Delete the data structures $\Phi_{g'}$ and $\Phi_{g''}$.
21. Return $\min_{s \in V(S)} c(\text{rt}(G),s)$.

Procedure bestReceiver is implemented as follows:

**Procedure** bestReceiver$(g,x,s)$

1. Query the data structure $\Phi_x$ with the query range $[t(\text{pa}(s)), t(s)-1]$. Let $e$ denote the returned edge.
2. If $e$ happens to be the edge $(\text{pa}(s),s)$, then remove $e$ from $\Phi_x$, and repeat the above step.
3. Reinsert any removed edges back into $\Phi_x$.
4. Return $e$.

THEOREM 3.2. The D-MPR problem on $G$ and $S$ can be solved in $O(mn \log n)$ time.

### 3.3 Considering distance-dependent transfer costs

Under the current reconciliation model, all transfers have the same cost irrespective of the span of the transfer. However, it has been observed that transfers are more likely to occur between closely related species than between distantly related ones (Andam and Gogarten, 2011). This suggests that, ideally, the cost of a transfer should depend on the phylogenetic distance between the donating and receiving species. Such a cost scheme could be implemented in several different ways: one straightforward way to implement this is to define the transfer cost between species $a$ and $b$ to be $P_\Theta(a,b) = \theta_1 + d_S(a,b) \cdot \theta_2$, where $\theta_1, \theta_2 \geq 0$. If branch lengths are available on the species tree, $d_S(a,b)$ could also be replaced by a term that counts the total branch length between $a$ and $b$. A simpler alternative is to have different constant transfer costs for different ranges of transfer spans.

Next, we give a simple $O(mn^2)$-time algorithm for the (general) MPR problem that can work with undated, partially dated, or fully dated species trees and can handle distance-dependent transfer costs. This makes it a factor of $n$ faster than the fastest known algorithm that can handle distance-dependent transfer costs. Our algorithm, which we will refer to as algorithm *reconcile*, is essentially the same as algorithm *U-Reconcile*, except that we remove our dependence on the *out* array and assign a cost of $\infty$ to those transfers that violate any given time constraints. Specifically, we (i) remove Lines 14, 22, and 23 through 25 and (ii) replace Steps 11 and 19 with the following five:

Let $X = \{x \in V(S)$: $x$ is incomparable to and potentially coexisting with $s\}$.

**If** $X \neq \emptyset$ **then**

**for** each $x \in X$

$$\text{Temp}(x) = P_\Theta(s,x) + \min\{\text{in}(g',x) + \text{in}(g'',x), \text{in}(g'',x) + \text{in}(g',x)\}.$$

$$c_\Theta(g,s) = \min_{x \in X} \text{Temp}(x)$$

Given any $a$ and $b$, the value of $P_\Theta(a,b)$ under distance-dependent transfer costs can be computed in constant time as long as the value $d_S(a,b)$ (or its equivalent in terms of branch lengths) can be computed in constant time. This can be achieved after an $O(n)$ preprocessing of the species tree, which (i) allows constant time LCA querying (Bender *et al.*, 2005) and (ii) labels each species tree node with its distance (or total branch length) from the root. This yields the following theorem.

THEOREM 3.3. The MPR problem on $G$ and $S$ with distance-dependent transfer costs can be solved in $O(mn^2)$ time.

### 3.4 Algorithmic extensions

*Unrooted gene trees.* If the input gene trees are unrooted, each possible rooted version of the unrooted gene tree is reconciled against the species tree and the goal is to find a reconciliation that has minimum cost among all rootings. Each of our three algorithms described earlier can be easily extended to work with unrooted gene trees without any increase in their respective time complexities. This is done by relying on the oft-used observation (Chen *et al.*, 2000) that, w.r.t. any internal node $g$, all rootings of the tree can be partitioned into three sets, depending on which of the tree edges incident on the node is closest to the root node. We have implemented this feature into our software RANGER-DTL.

*Multiple optimal solutions.* It should be noted that, for any given values of the event costs $P_\Delta$, $P_\Theta$ and $P_{\text{loss}}$, there may be more than one optimal solution for the MPR problem. The $O(mn^2)$ algorithm above can be easily adapted to output all possible optimal reconciliations for any given problem instance.

*Further enhancements.* It is also possible to extend each of our three algorithms, without any increase in their time complexities, to consider more complex biological scenarios, such as transfers from potentially extinct or unsampled lineages, or transfer from a species that then loses its copy of that gene. A more detailed discussion of these enhancements appears in the Supplementary Material (Section S.2).

## 4 EXPERIMENTAL EVALUATION

We implemented our fast algorithms into a software package called RANGER-DTL (Rapid ANalysis of Gene family Evolution using Reconciliation-DTL). Since the accuracy and utility of DTL and tcDTL reconciliation for inferring gene family evolution have already been demonstrated elsewhere (David and Alm, 2011; Doyon *et al.*, 2010; Gorbunov and Liubetskii, 2009; Tofigh, 2009), we do not attempt to do so here. Instead, our goal is to (i) demonstrate the immense speedup in running time achieved by our algorithms over existing state-of-the-art programs; (ii) compare the solutions obtained by DTL reconciliation on undated and fully dated species trees against tcDTL reconciliation on fully dated trees (which can be thought of as a 'gold standard'); and (iii) demonstrate the utility of enhancements such as distance-dependent transfer costs. To that end, we applied RANGER-DTL to a variety of simulated and biological datasets. Specifically, we created 500 simulated datasets (gene tree–species tree pairs), 100 each with 50, 100, 200, 500 and 1000 taxa generated using the probabilistic gene evolution model described in Arvestad *et al.* (2009); Tofigh (2009); Tofigh *et al.* (2011). We ensured that each simulated gene tree had at least one gene from each species in the corresponding species tree, and they contained on average 98.2, 195, 334.3, 618.8 and 1423.5 leaves, respectively, for the 50, 100, 200, 500 and 1000 taxa datasets. We also created a 10 000-taxon gene tree–species tree pair with random topologies to demonstrate the feasibility of analyzing even extremely large trees with RANGER-DTL. We point out that the running time depend only on the sizes of the input gene and species trees and are thus independent of the actual rate parameters used to generate the simulated trees and of the event costs used to compute the reconciliation. Our biological dataset was derived from David and Alm (2011) and consists of over 4700 unrooted gene trees with a species tree of 100 (predominantly prokaryotic) species sampled broadly across the tree of life. This biological dataset was analyzed using the same cost parameters ($P_\Delta = 2$, $P_\Theta = 3$, $P_{\text{loss}} = 1$) used in David and Alm (2011).

*Running time.* To compare the running time of our algorithms, we used an implementation of our algorithm for DTL reconciliation on undated species trees, referred to as the RANGER-DTL-U program, and compared it against AnGST (David and Alm, 2011) and Mowgli (Doyon *et al.*, 2010) which are two of the most advanced programs implementing the fastest known algorithms for DTL reconciliation on undated species trees and tcDTL reconciliation on fully dated species trees, respectively. When running RANGER-DTL-U and AnGST on these datasets, all divergence-time information (branch lengths) on the nodes of the species trees was ignored. Moreover while both RANGER-DTL and AnGST can efficiently handle unrooted gene trees, Mowgli cannot; thus, we first randomly rooted each of the 4733 gene trees of the biological dataset. Table 1 depicts the results. We find a dramatic improvement in runtime and scalability over both AnGST and Mowgli. For instance, on the 100 simulated 100-taxon datasets, RANGER-DTL-U is an impressive 300 and 4500 times faster than AnGST and Mowgli, respectively. Similar speedups are observed on the biological dataset as well, with RANGER-DTL-U requiring just over a minute to analyze the entire dataset of 4733 gene trees. (Even when run directly on the original unrooted gene trees, it requires only about 2 min to analyze the entire dataset). Moreover, the speedups are, as anticipated, even greater for larger datasets. AnGST required between 8 and 10 h on each of the 10 randomly chosen 500-taxon datasets that we tried, suggesting a running time of at least 800 h on all 100 datasets, and it crashed immediately on the 1000-taxon datasets. Similarly, Mowgli crashed after ~4 h of running time on each of the 10 randomly chosen 500-taxon datasets that we tried, and did not terminate in 60 h (after which we stopped the program) on any one of the 10 1000-taxon datasets we ran it on. This suggests a total running time of at least 400 and 6000 h on all 100 of the 500- and 1000-taxon datasets, respectively, for Mowgli. In contrast, RANGER-DTL-U required <2 s on each 1000-taxon dataset, which is, remarkably, over 100 000 times faster than Mowgli. While neither AnGST nor Mowgli can be run on the 10 000-taxon dataset, RANGER-DTL-U required only ~4 h to analyze it.

*Solution quality.* Note that it is ineffective to compare the actual reconciliations themselves as the presence of multiple

**Table 1.** Runtime comparison

| Dataset type | Dataset size | RANGER-DTL-U | AnGST | Mowgli |
|---|---|---|---|---|
| Simulated | 50 taxa (100 datasets) | 2 s | 3 m:26 s | 28 m:30 s |
| | 100 taxa (100 datasets) | 3 s | 15 m:4 s | 3 h:52 m |
| | 200 taxa (100 datasets) | 9 s | 1 h:2 m | 29 h:43 m |
| | 500 taxa (100 datasets) | 35 s | >800 h | >400 h |
| | 1000 taxa (100 datasets) | 2 m:57 s | — | >6000 h |
| | 10 000 taxa (1 dataset) | 4 h:7 m | — | — |
| Biological | 4733 gene trees, 100 taxa species tree | 1 m:03 s | 3 h:45 m | 41 h:36 m |

This table shows the runtimes of RANGER-DTL-U, AnGST and Mowgli on simulated and biological datasets. Times are shown in hours (h), minutes (m) and seconds (s). Experiments were performed on a desktop computer with a 3.2 GHz Intel Core i3 processor and 4 GB of RAM.

optimal reconciliations confounds the ability to make meaningful comparisons. Thus, we focused on comparing the reported optimal reconciliation costs. On all datasets, the reconciliation costs reported by RANGER-DTL-U are, as expected, identical to those reported by AnGST. When compared with Mowgli, we observed that the reconciliation costs reported by RANGER-DTL-U were 7.9% lower on the biological dataset. The fact that the costs reported by RANGER-DTL-U are smaller is unsurprising as it ignores all timing information, while Mowgli uses it. The timing information on the biological species tree is also likely to be at least slightly inaccurate, further contributing to the difference in reconciliation costs. On the simulated datasets, we observed practically no difference in the scores for RANGER-DTL-U and Mowgli, even on datasets with high rates of duplication, transfer and loss (results not shown), likely due to the fact that simulations inherently simplify the evolutionary process and yield less complex gene trees. We also ran the fully dated version of DTL reconciliation, RANGER-DTL-D, on the biological dataset and observed that, compared with Mowgli, the reported costs are on average only 3.7% lower. Overall, our experiments show that (i) on fully dated trees, solutions to the DTL reconciliation problem closely approximate solutions obtained by tcDTL reconciliation; and (ii) even when the species trees are undated, the DTL reconciliation problem yields solutions that are largely similar to those obtained with perfect timing information.

*Distance-dependent transfer costs.* To test the utility of incorporating distance-dependent transfer costs, we modified the RANGER-DTL-D program so as to increase the transfer cost by 2 over its current constant value of 3 whenever the transfer edge spanned more than 10 edges (which represents a sizable distance in a species tree with only 100 taxa). We observed that the reported costs, on the biological dataset, were on average 17.2% higher than the unmodified RANGER-DTL-D. This implies that the computed optimal reconciliations contain a large number of transfer events that span >10 edges. This strongly suggests that using distance-dependent transfer costs is likely to have a significant impact on the quality of the inferred reconciliations.

It is worth mentioning that even our general $O(mn^2)$ algorithm for the MPR problem with distance-dependent transfer costs significantly outperforms AnGST and Mowgli in terms of running time. For example, on the entire biological dataset of 4733 gene trees, it requires ∼13 min of running time, compared with almost 4 h by AnGST and over 41 h by Mowgli. Even on the 1000-taxon datasets, it required <15 min per dataset. Although we have not yet implemented our fast $O(mn\log n)$-time algorithm for the D-MPR problem (since the general $O(mn^2)$ algorithm solves the D-MPR problem as well), its runtime can be expected to be only slightly higher than that of RANGER-DTL-U.

RANGER-DTL can be freely downloaded from `http://compbio.mit.edu/ranger-dtl/`.

## 5 DISCUSSION AND CONCLUSION

In this article, we addressed the DTL reconciliation problem for reconstructing gene family evolution. We proposed new algorithms that are dramatically faster than any existing algorithms for this problem and proposed several enhancements necessary for improving the utility and accuracy of the computed solutions. Our work represents a substantial improvement in the ability to accurately analyze large gene families. It also enables, for the first time, the use of powerful, reconciliation-based gene tree and species tree reconstruction methods for prokaryotes. For instance, to reconstruct a 100-taxon species tree by gene tree parsimony, using a standard local search heuristic, one would need to reconcile on the order of many millions of gene tree/species tree pairs; using even the fastest existing DTL reconciliation algorithms, such as AnGST, one would require several years of computing time to perform such an analysis, compared with just a few days using RANGER-DTL.

There are a number of ways to further improve the accuracy of DTL reconciliation and we would like to explore these in the future. For instance, it would help to explicitly distinguish between two types of transfers: ones that contribute an additional gene to the recipient genome and those that recombine with an existing gene copy and replace it. Under the current DTL reconciliation models, recombining transfers are counted as a transfer followed by a loss. Moreover, our current implementation assumes that the input gene tree topology is correct and it would be very useful to have an effective way to deal with any uncertainty in gene tree topologies.

## ACKNOWLEDGEMENTS

The authors thank Ali Tofigh for help with the tree simulation software, Lawrence David for providing the biological dataset, and Matt Rasmussen for helpful discussions.

*Funding*: National Science Foundation CAREER award 0644282 to M.K., National Institutes of Health RC2 HG005639 to M.K. and National Science Foundation AToL 0936234 to E.J.A. and M.K.

*Conflict of Interest*: none declared.

## REFERENCES


Andam,C.P. and Gogarten,J.P. (2011) Biased gene transfer in microbial evolution. *Nat. Rev. Microbiol.*, **9**, 543–555.

Arvestad,L. *et al.* (2009) The gene evolution model and computing its associated probabilities. *J. ACM*, **56**, 7:1–7:44.

Bansal,M.S. *et al.* (2007) Heuristics for the gene-duplication problem: a $\Theta(n)$ speed-up for the local search. In Speed, T.P. and Huang, H. (eds), *RECOMB*, Vol. 4453 of *Lecture Notes in Computer Science*, Springer (Berlin Heidelberg), pp. 238–252.

Bender,M.A. *et al.* (2005) Lowest common ancestors in trees and directed acyclic graphs. *J. Algor.*, **57**, 75–94.

Boc,A. *et al.* (2010) Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Syst. Biol.*, **59**, 195–211.

Bonizzoni,P. *et al.* (2005) Reconciling a gene tree to a species tree under the duplication cost model. *Theor. Comput. Sci.*, **347**, 36–53.

Brodal,G.S. *et al.* (2011) Path minima queries in dynamic weighted trees. In F. Dehne, *et al.* (eds), *WADS.*, Vol. 6844 of *Lecture Notes in Computer Science*, Springer, pp. 290–301.

Burleigh,J.G. *et al.* (2011) Genome-scale phylogenetics: inferring the plant tree of life from 18,896 gene trees. *Syst. Biol.*, **60**, 117–125.

Charleston,M. (1998) Jungles: a new solution to the host–parasite phylogeny reconciliation problem. *Math. Biosci.*, **149**, 191–223.

Charleston,M.A. and Perkins,S.L. (2006) Traversing the tangle: algorithms and applications for cophylogenetic studies. *J. Biomed. Inform.*, **39**, 62–71.

Chauve,C. *et al.* (2008) Gene family evolution by duplication, speciation, and loss. *J. Comput. Biol.*, **15**, 1043–1062.

Chen,K. *et al.* (2000) Notung: a program for dating gene duplications and optimizing gene family trees. *J. Comput. Biol.*, **7**, 429–447.

Conow,C. *et al.* (2010) Jane: a new tool for the cophylogeny reconstruction problem. *Algorithm. Mol. Biol.*, **5**, 16.

Cormen,T.H. *et al.* (2009) *Introduction to Algorithms*, 3rd edn. MIT press.

Csürös,M. and Miklós,I. (2006) A probabilistic model for gene content evolution with duplication, loss, and horizontal transfer. In Apostolico, A. *et al.* (eds.), *RECOMB*, Vol. 3909 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg, pp. 206–220.

David,L.A. and Alm,E.J. (2011) Rapid evolutionary innovation during an archaean genetic expansion. *Nature*, **469**, 93–96.

Doyon,J.-P. *et al.* (2010) An efficient algorithm for gene/species trees parsimonious reconciliation with losses, duplications and transfers. In Tannier, E. (ed.), *RECOMB-CG*, Vol. 6398 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg, pp. 93–108.

Durand,D. *et al.* (2006) A hybrid micro-macroevolutionary approach to gene tree reconstruction. *J. Comput. Biol.*, **13**, 320–335.

Eulenstein,O. and Vingron,M. (1998) On the equivalence of two tree mapping measures. *Discrete Appl. Math.*, **88**, 101–126.

Goodman,M. *et al.* (1979) Fitting the gene lineage into its species lineage. A parsimony strategy illustrated by cladograms constructed from globin sequences. *Syst. Zool.*, **28**, 132–163.

Gorbunov,K.Y. and Liubetskii,V.A. (2009) Reconstructing genes evolution along a species tree. *Mol. Biol.*, **43**, 946–958.

Górecki,P. and Tiuryn,J. (2006) Dls-trees: a model of evolutionary scenarios. *Theor. Comput. Sci.*, **359**, 378–399.

Hallett,M.T. and Lagergren,J. (2001) Efficient algorithms for lateral gene transfer problems. In Lengauer, T. (ed), *Proceedings of the fifth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, ACM (New York), pp. 149–156.

Hill,T. *et al.* (2010) Sprit: identifying horizontal gene transfer in rooted phylogenetic trees. *BMC Evol. Biol.*, **10**, 42.

Huelsenbeck,J.P. *et al.* (2000) A Bayesian framework for the analysis of cospeciation. *Evolution*, **54**, 352–364.

Jin,G. *et al.* (2009) Parsimony score of phylogenetic networks: hardness results and a linear-time heuristic. *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, **6**, 495–505.

Koonin,E.V. (2005) Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.*, **39**, 309–338.

Libeskind-Hadas,R. and Charleston,M. (2009) On the computational complexity of the reticulate cophylogeny reconstruction problem. *J. Comput. Biol.*, **16**, 105–117.

Ma,J. *et al.* (2008) Dupcar: reconstructing contiguous ancestral regions with duplications. *J. Comput. Biol.*, **15**, 1007–1027.

Merkle,D. and Middendorf,M. (2005) Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. *Theor. Biosci.*, **123**, 277–299.

Merkle,D. *et al.* (2010) A parameter-adaptive dynamic programming approach for inferring cophylogenies. *BMC Bioinform.*, **11**(Suppl. 1), S60.

Mi,H. *et al.* (2010) Panther version 7: improved phylogenetic trees, orthologs and collaboration with the gene ontology consortium. *Nucleic Acids Res.*, **38**(Suppl. 1), D204–D210.

Mirkin,B. *et al.* (1995) A biologically consistent model for comparing molecular phylogenies. *J. Comput. Biol.*, **2**, 493–507.

Nakhleh,L. *et al.* (2004) Reconstructing reticulate evolution in species: theory and practice. In Bourne and Gusfield (eds), *Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, 2004, ACM (New York), pp. 337–346.

Nakhleh,L. *et al.* (2005) RIATA-HGT: a fast and accurate heuristic for reconstructing horizontal gene transfer. In Wang, L. (ed.), *COCOON*, Vol. 3595 of *Lecture Notes in Computer Science*, Springer, pp. 84–93.

Ovadia,Y. *et al.* (2011) The cophylogeny reconstruction problem is np-complete. *J. Comput. Biol.*, **18**, 59–65.

Page,R.D.M. (1994) Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. *Syst. Biol.*, **43**, 58–77.

Rasmussen,M.D. and Kellis,M. (2011) A Bayesian approach for fast and accurate gene tree reconstruction. *Mol. Biol. Evol.*, **28**, 273–290.

Ronquist,F. (1995) Reconstructing the history of host–parasite associations using generalised parsimony. *Cladistics*, **11**, 73–89.

Ronquist,F. (2003) Parsimony analysis of coevolving species associations. In Page, R.D.M. (ed.), *Tangled Trees: Phylogeny, Cospeciation and Coevolution*, The University of Chicago Press, Chicago, pp. 22–64.

Rutschmann,F. (2006) Molecular dating of phylogenetic trees: a brief review of current methods that estimate divergence times. *Divers. Distrib.*, **12**, 35–48.

Sennblad,B. and Lagergren,J. (2009) Probabilistic orthology analysis. *Syst. Biol.*, **58**, 411–424.

Storm,C.E.V. and Sonnhammer,E.L.L. (2002) Automated ortholog inference from phylogenetic trees and calculation of orthology reliability. *Bioinformatics*, **18**, 92–99.

Tofigh,A. (2009) *Using trees to capture reticulate evolution: lateral gene transfers and cancer progression*. PhD Thesis, KTH Royal Institute of Technology, Sweden.

Tofigh,A. *et al.* (2011) Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **8**, 517–535.

van der Heijden,R. *et al.* (2007) Orthology prediction at scalable resolution by phylogenetic tree analysis. *BMC Bioinform.*, **8**, 83.

Vilella,A.J. *et al.* (2009) Ensemblcompara genetrees: complete, duplication-aware phylogenetic trees in vertebrates. *Genome Res.*, **19**, 327–335.

Vuillemin,J. (1978) A data structure for manipulating priority queues. *Commun. ACM*, **21**, 309–315.

Wapinski,I. *et al.* (2007) Natural history and evolutionary principles of gene duplication in fungi. *Nature*, **449**, 54–61.