BIOINFORMATICS

Vol. 00 no. 00 2012
Pages 1–8

Inferring Duplications, Losses, Transfers and Incomplete Lineage Sorting with Non-Binary Species Trees

Maureen Stolzer[1,*] Han Lai[1] Minli Xu[2] Deepa Sathaye[3] Benjamin Vernot[4] and Dannie Durand[1,3]

[1]Department of Biological Sciences, Carnegie Mellon University, Pittsburgh, PA, USA 15213
[2]Lane Center for Computational Biology, Carnegie Mellon University, Pittsburgh, PA, USA 15213
[3]Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA 15213
[4]Department of Genome Sciences, University of Washington, Seattle, WA, USA 98195

ABSTRACT

Motivation: Gene duplication (D), transfer (T), loss (L), and incomplete lineage sorting (I) are crucial to the evolution of gene families and the emergence of novel functions. The history of these events can be inferred via comparison of gene and species trees, a process called reconciliation, yet current reconciliation algorithms model only a subset of these evolutionary processes.

Results: We present an algorithm to reconcile a binary gene tree with a non-binary species tree under a DTLI parsimony criterion. This is the first reconciliation algorithm to capture all four evolutionary processes driving tree incongruence and the first to reconcile non-binary species trees with a transfer model. Our algorithm infers all optimal solutions and reports complete, temporally feasible event histories, giving the gene and species lineages in which each event occurred. It is fixed-parameter tractable, with polytime complexity when the maximum species outdegree is fixed. Application of our algorithms to prokaryotic and eukaryotic data shows that use of an incomplete event model has substantial impact on the events inferred and resulting biological conclusions.

Availability: Our algorithms have been implemented in NOTUNG, a freely available phylogenetic reconciliation software package, available at `http://www.cs.cmu.edu/~durand/Notung`.

Contact: mstolzer@andrew.cmu.edu

1 INTRODUCTION

The phylogeny of a gene family evolving by vertical descent will agree with the associated species tree. Gene duplication, gene loss, horizontal gene transfer (HGT), or incomplete lineage sorting (ILS) can result in a gene tree that differs from the species tree [15]. The history of such events can be inferred through topological comparison of gene and species trees, a process called *reconciliation*. Reconciliation encompasses two related problems: event inference and tree inference. Given rooted gene tree and species trees, a mapping from extant genes to extant species, and an event model, the goal of *event inference* is to infer the association between ancestral genes and species and the optimal event history with respect to a combinatorial or probabilistic optimization criterion. A complete solution must include the specific events and the gene and species lineages in which those events occurred. Given a set of gene trees, *tree inference* seeks the species tree that optimizes the combined events resulting from reconciliation with each gene tree in the input set.

Here, we address the event inference problem for a model that captures all four evolutionary processes contributing to gene tree incongruence. Whole genome sequencing data is revealing an ever growing number of cases where all four processes are active (e.g., [1, 32, 26]), leading to calls for algorithms that model multiple evolutionary processes [11, 7]. Algorithms lacking a model of incongruence due to ILS will overestimate the number of duplications and/or transfers. For example, a recent analysis, based on a model that did not consider ILS, reported an inexplicable but dramatic increase in duplications in recently sequenced mammalian genomes [18]. For large scale analysis of multigenome phylogenetic data sets, reconciliation algorithms that allow ILS to be distinguished from other sources of incongruence are essential.

Related Work. Gene tree incongruence has been considered from two perspectives. Multispecies coalescent models focus on ILS as a source of incongruence [reviewed in 7]. The basic assumption underlying this work is that gene tree incongruence arises from ILS due to genetic drift, although some methods also take hybridization and/or recombination into account [reviewed in 7, 11]. The multispecies coalescent explicitly relates the probability of an incongruent gene tree to the time between species divergences and the effective size of the ancestral population. In the context of tree inference, these parameters can be inferred from a collection of gene trees. Event inference, however, requires prior estimates of population parameters since only one tree is under consideration.

In contrast, reconciliation focuses on incongruence that arises from processes that change the number of loci in a gene family; i.e., duplication, loss, and transfer. Most event inference algorithms consider either gene duplication or HGT [9, 20, 19], but not both. Exact algorithms with exponential time complexity have been presented for the Duplication-Transfer (DT) [28] and Duplication-Transfer-Loss (DTL) models [6], under a parsimony criterion. Event inference with transfers is NP-complete [12], but can be solved in polynomial time under a restricted model where only transfers between contemporaneous species are considered. This model [reviewed in 13, 9] requires estimates of speciation times, which are frequently not known. In addition, algorithms for this restricted model may fail to recognize transfers if they involve a taxon missing from the data set [19, 13].

Reconciliation implicitly assumes that inter-speciation times are sufficiently long that genetic drift and incomplete lineage sorting may be safely excluded from consideration. This assumption breaks down when the species tree contains polytomies or very short branches. In these situations, allelic variation can survive multiple speciation events, leading to gene trees with branching patterns that differ from the species tree. Such cases are increasingly common due to increased sequencing of closely related species. Methods that

*to whom correspondence should be addressed

do not consider ILS will incorrectly interpret incongruence arising from ILS as evidence of duplication or transfer.

To avoid this problem, algorithms that can distinguish between ILS and other events are needed. In fact, one parsimony criterion that considers ILS has been proposed: Minimization of the number of extra gene lineages on a species branch due to Deep Coalescence (MDC) has been used as a criterion for tree inference [15, 16, 27, 21, 17]. However, the MDC criterion assumes *all* incongruence is due to ILS. MDC is not a suitable basis for event inference because it cannot distinguish between extra lineages arising from ILS and those arising from duplication or transfer [31]. Two approaches to the event inference problem combine ILS with gene duplication and loss in a single model (DLI). In earlier work, we presented the first event inference algorithm for the DLI model under a parsimony criterion [29]. An event inference algorithm for a DLI model based on the multispecies coalescent relates the probability of ILS to branch lengths and population sizes explicitly [23]. These models have different strengths. The model based on the coalescent captures more detail, but is limited to the small number of data sets for which estimates of ancestral population sizes and speciation times are available. To our knowledge, no reconciliation algorithms that consider ILS and transfer are in existence.

Our contributions: We present the first reconciliation algorithm for a DTLI event model that captures all four major causes of gene tree incongruence. Our algorithm is also the first to allow transfers in reconciliation with a non-binary species tree. Our algorithm is based on a simple, elegant model that recognizes ILS as a source of incongruence, but avoids the computational overhead of a full coalescent model and does not require estimates of ancestral population sizes and speciation times.

Our parsimony-based algorithm reconciles a binary gene tree with a non-binary species tree and distinguishes between incongruence that could only arise through duplication or HGT and incongruence that can be more parsimoniously explained by ILS. Our algorithm places no restriction on speciation times and reports all optimal reconciliations that are temporally feasible. For a fixed k^*, the time complexity of our algorithm is $O(h_S|V_G||V_S|^2)$ time, where k^* is the out-degree of the largest polytomy in the species tree, h_S is the height of the species tree, and $|V_G|$ and $|V_S|$ are the number of vertices in the gene and species trees, respectively. Given a binary species tree, our algorithm infers histories under the DTL model.

Both the DTL and DTLI algorithms have been implemented in Java and integrated in NOTUNG, a freely available software package for phylogenetic reconciliation. Our software offers a unique and comprehensive combination of functions: it includes losses in the optimization criterion, does not require estimates of speciation times, and reports all optimal event histories. Reported solutions are complete, temporally feasible event histories, giving the gene and species lineages in which each event occurred.

To demonstrate the advantages of a full DTLI model on real data, we applied our algorithm to two phylogenetic datasets that have been used in previous analyses of HGT and phylogenetic incongruence [8, 24, 33]. First, if no incongruent trees have patterns that could be most parsimoniously explained as ILS, then models with and without ILS should give same results. In fact, we observed just the opposite. The models that did not correct for ILS substantially overestimated duplications and transfers. A recent study using a quartet decomposition approach reported several highways of gene transfer between specific pairs of cyanobacterial species [2]. We observed the same highways using the DTL algorithm. Only one of these highways remained when using the DTLI algorithm. Second, since many published algorithms do not include losses in the optimization criterion [4, 14, 34, 28, e.g.,], we compared models with losses (DTLI, DTL) and without losses (DTI, DT). Explicit inclusion of losses in the optimization function resulted in substantial changes to the inferred ratio of duplications to transfers, suggesting that the practice of *post hoc* inference of losses should be revisited.

Finally, when the event model includes transfers, the minimum cost event history is not, in general, unique. All algorithms cited above report only one of possibly many optimal solutions. We applied our algorithm to assess the extent to which multiple optimal solutions occur. We discovered that multiple optimal solutions are a frequent occurrence, especially in data sets where transfer is the dominant process. In the analysis reported here, 20% of 1128 cyanobacterial trees had multiple optimal solutions with inconsistent event histories. In other words, for one in five trees, the arbitrary selection of a single optimal solution could lead to conclusions that might not be supported by other optimal solutions. The results presented here are exciting and important, as they demonstrate that degeneracy and the applied event model have substantial impact on the histories inferred and, hence, on the resulting biological conclusions.

Notation: Given a tree $T_i = (V_i, E_i)$, $L(T_i)$ designates the leaf set of T_i, and ρ_i designates its root. We use $g \in V_G$ and $s \in V_S$ to represent genes and species, respectively. $T_i(v)$ is the subtree of T_i rooted at $v \in V_i$. $C(v)$ and $P(v)$ denote the children and the parent of v, respectively, with $c_j \in C(v)$ denoting the jth child of v. We adopt the notation that if $(u,v) \in E_i$, $P(v) = u$. Given nodes $u, v \in V_i$, if u is on the path from v to ρ, then u is an ancestor of v, designated $u \geq_i v$, and v is a descendant of u, designated $v \leq_i u$. If $v \not\geq_i u$ and $u \not\geq_i v$, u and v are *incomparable*, designated $u \not\gtrless_i v$.

2 ALGORITHMS

Here, we propose a reconciliation model based on duplication-transfer-loss parsimony that distinguishes between regions of the species tree where ILS is likely, and those where only gene duplication and transfer need be considered. These differences are specified using a non-binary species tree: At binary nodes, we assume that ILS is so rare that incongruence is always evidence of gene duplication or transfer. At polytomies, ILS is considered, and gene duplication and transfer are invoked only if topological disagreement cannot be explained by ILS. This model can be invoked for both non-binary species trees and for binary species trees with short branches where ILS is suspected: Even when the binary branching order of the species tree is known, the user can collapse edges in the species tree to indicate in which lineages ILS should be considered as an alternate hypothesis.

A key aspect of our model is that even when ILS is allowed, it is not possible to explain all incongruence in terms of ILS, even in a uniquely labeled gene tree. Let g be a node in T_G and let $s \in V_S$ be the associated node in the species tree. We wish to determine whether the divergence at g is consistent with a co-divergence at s or whether it can only be explained by events that give rise to a new locus; i.e., duplication and transfer. If the branch point at g arose through a co-divergence with s, then each species lineage descending from s should inherit at most one of descendant of g. The presence of more than one descendant of g indicates that the divergence at g must be due to acquisition of an additional locus by duplication or transfer. An operational test for detecting more than one descendant on a branch results from the observation that any branching pattern

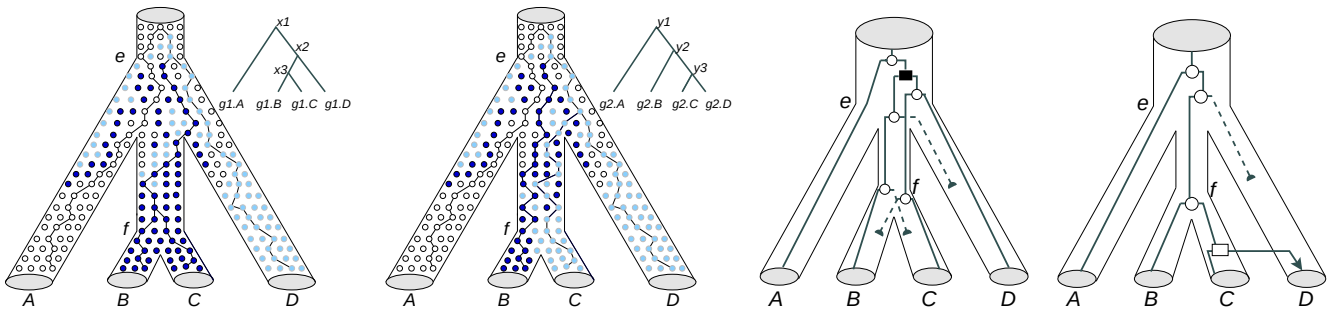


Fig. 1: Reconciliation of binary gene trees with a non-binary species tree under our DTLI model. *(a)* A binary gene tree that is consistent with a binary resolution of the species tree. The divergences at x_1 and x_2 are consistent with ILS. *(b)* A gene tree that does not correspond to any binary resolution of the species tree. Node y_2 is not consistent with deep coalescence: the embedding requires two descendants of y_2 on the branch from e to f, a violation of model constraints. This can only be explained by persistent polymorphism (light and dark dots) on a long branch. DTLI reconciliation of the gene tree in (b) with the non-binary T_S results in two optimal solutions for suitable choices of δ, λ and τ: *(c)* one duplication followed by three losses and *(d)* one transfer and a loss. Duplications are represented by a filled boxes, speciations by open circles, transfers by open boxes and arrows, and losses by dashed lines and filled half-circles. Each dot represents an allele of a single individual, with the dot's color indicting the type of allele. Rows represent generations of individuals.

that is consistent with a binary resolution of the polytomy can be explained by lineage sorting.

For example, the gene tree in Fig. 1a represents a valid, binary resolution of the species tree, consistent with ILS. The embedding of the gene tree in the species tree shows that each species tree lineage inherits exactly one descendant of x_1 and at most one descendant of x_2. Both x_1 and x_2 can be interpreted as deep coalescences. In contrast, there is no binary resolution of the species tree that corresponds to the gene tree in Fig 1b. The embedding of this gene tree requires two descendants of y_2 in the lineage from e to f, a violation of model constraints. The only way to explain two descendants of y_2 on the branch from e to f is by inferring a duplication (Fig. 1c) or a transfer (Fig. 1d).

Prior to introducing our algorithm, we discuss the meaning of a polytomy in our model. A species polytomy can be considered from two perspectives: a "hard" polytomy represents simultaneous divergence of three or more populations. A "soft" polytomy represents a binary branching process in which the branching order is unknown. Our model assumes that a polytomy represents rapid or simultaneous species divergence. However, it also admits a useful interpretation for soft polytomies. A soft polytomy can be viewed as a set of hypotheses, namely the set of binary resolutions of the polytomy. Our model offers a conservative stance: events are only inferred when the topology of the gene tree does not correspond to any of these hypotheses. Note that in some cases, the hard and soft polytomy models are closely linked: the branching order of species that arose through multiple speciations in rapid successions [22, 10] is often difficult to resolve.

2.1 The DTLI Algorithm

In our DTLI model, divergence in a gene tree arises through one of four events: duplication ($\mathcal{D}$), transfer ($\mathcal{T}$), speciation ($\mathcal{S}$), and deep coalescence ($\mathcal{C}$). The score of a reconciliation under this model is the weighted sum of the number of duplications ($N_{\mathcal{D}}$), losses ($N_{\mathcal{L}}$), and transfers ($N_{\mathcal{T}}$):

$$\pi = \delta \cdot N_{\mathcal{D}} + \lambda \cdot N_{\mathcal{L}} + \tau \cdot N_{\mathcal{T}}, \qquad (1)$$

where δ, λ and τ, respectively, are the costs of a duplication, loss, and transfer. Speciation and deep coalescence represent co-divergence with binary nodes and polytomies, respectively, in the species tree and have zero cost. We refer to the cost of event $\varepsilon \in \{\mathcal{D}, \mathcal{T}, \mathcal{S}, \mathcal{C}\}$ as $\kappa(\varepsilon)$.

A rooted, binary gene tree T_G; a rooted, arbitrary species tree T_S; a mapping $M_L : L(V_G) \to L(V_S)$ from contemporary genes to the species from which they were sampled; and a set of permitted events are given as input. The reconciliation of T_G with T_S results in an annotated tree, $R_{GS} = (V_G, E_G)$, in which every internal node, g, is annotated with the species $s \in V_S$ that contained gene g, designated $M(g)$, and the event that caused the divergence at g, designated $\mathcal{E}(g)$. In addition, every $g \in V_G \setminus \{\rho_G\}$ is annotated with $\mathcal{L}(g)$, the genes lost on the edge from $P(g)$ to g. Each loss is labeled with the species in which the loss occurred. We say $(u,v) \in E_G$ is a transfer edge if $\mathcal{E}(u) = \mathcal{T}$ and $M(u) \not\leq_S M(v)$ and define $\Lambda(R_{GS}) \subset E_G$ to be the set of transfer edges in R_{GS}. If $(u,v) \in \Lambda(R_{GS})$, a transfer occurred from donor species $d = M(u)$ to recipient species $r = M(v)$.

Here, we present the DTLI event inference problem under the constraint that a deep coalescent is inferred at g iff each lineage descending from $M(g)$ inherits at most one descendant of g:

The DTLI Event Inference Problem

Input: A rooted non-binary species tree, T_S; a rooted, binary gene tree, T_G; the leaf mapping, M_L.

Output: All reconciliation histories R_{GS} that minimize π and satisfy the model constraints.

Algorithms for the DTLI event model must address several issues that do not arise when only a subset of the events is considered: (1) There may be more than one combination of duplications, transfers and losses that gives rise to the same pattern of tree incongruence (i.e., there may be more than one optimal solution, R_{GS}). (2) The value of $M(g)$ is not uniquely determined by the children of g and multiple possible values of $M(g)$ must be considered because transfers cause genes to jump to distant locations in the species tree. (3) An optimal reconciliation at the root may entail a suboptimal reconciliation at an internal node, g. Inferring a more costly event at g may change the values of $M(\cdot)$ in nodes ancestral to g such that the overall score is reduced. Therefore, the values of $M(g)$ and $\mathcal{E}(g)$ required for an optimal solution cannot be determined using only local information, and more than one optimal solution may result.

To accommodate these requirements, it is necessary to enumerate all possible assignments of $M(g)$ and $\mathcal{E}(g)$, for each node $g \in V_G$.

At each g, the associated information is stored in two tables, $\mathcal{K}_g$ and $\mathcal{H}_g$. For each candidate assignment $s \in V_S$, the score that minimizes the cost of reconciling $T_G(g)$ with $T_S(s)$, is stored in $\mathcal{K}_g[s]$. The associated events and other information needed to reconstruct the history at g are stored in $\mathcal{H}_g[s]$.

Optimal reconciliations are calculated by a two-pass algorithm. The first pass (Alg. 2.1.1), is a dynamic program that populates each $\mathcal{K}_g$ and $\mathcal{H}_g$ in a post-order traversal of T_G. It returns the optimal reconciliation score, the values of $M(\rho_g)$ and $\widehat{W}(\rho_g)$ corresponding to that score, and the number of optimal histories. The second pass (Alg. S1.3.1), is a traceback algorithm that reads information from each $\mathcal{K}_g$ to construct an optimal solution. Each optimal history is generated by traversing, in pre-order of T_G, each unique path that lead to the optimal label(s) in $\mathcal{K}_{\rho_G}$. Appropriate values of $M(g)$ and $\mathcal{E}(g)$ at each node g are selected from $\mathcal{K}_g$. Each candidate optimal history is then tested for temporal feasibility, as described in the next section. Only those histories that are temporally feasible are reported.

A key calculation in the dynamic program of `firstPass` is determination of the possible events at g for a given candidate species assignment, $M(g) = s$. These events, in turn, depend on $M(c_1) = s_1$ and $M(c_2) = s_2$, where $c_1, c_2 \in C(g)$. The basis for determining candidate events that are consistent with s, s_1 and s_2 is the following observation: If a duplication occurred at g, then the species that inherit the descendants of c_1 and the species that inherit the descendants of c_2 will not be disjoint.

We define a test, based on this observation, for distinguishing duplication from other events:

$$\varepsilon = \mathcal{D} \text{ iff } \widehat{N}(c_1) \cap \widehat{N}(c_2) \neq \emptyset, \tag{2}$$

where $\widehat{N}(g)$ is the set of species that vertically inherit descendants of $P(g)$. If $\widehat{N}(c_1)$ and $\widehat{N}(c_2)$ are disjoint, than one of the other three events ($\mathcal{S}$, $\mathcal{C}$, or $\mathcal{T}$) must have occurred. These events can be distinguished from one another using $\widehat{N}(g)$, $M(g)$, and $M(c_1)$ and $M(c_2)$, as seen in `costCalc` in Alg. 2.1.1. Note that Eq. 2 is different from the standard least common ancestor (*lca*) test; however, when $M(g) = s$ is binary, the descendants of s are partitioned into two sets, the left and right descendants of s, iff there is no duplication. Therefore, it is equivalent to *lca* reconciliation [29].

Because $\widehat{N}$ only consists of elements that were vertically inherited, we must exclude transfer edges in the calculation. For this purpose, we define

$$\mathcal{R}(g) = \{h \in L(T_G(g)) | \exists z \ni (P(z), z) \in \Lambda(R_{GS}) \wedge h \leq_G z <_G g, \}$$

the set of leaves of $T_G(g)$ that were acquired through HGT. Formally, we define $\widehat{N} : V_G \to V_S^+$ to be a mapping from V_G to sets of nodes in V_S, where V_S^+ is the powerset of V_S. $\widehat{N}(g)$ is the set of children of $M(P(g))$ such that $\widehat{N}(g) = \{M(g)\}$ if $M(P(g)) \in L(T_S)$; otherwise, $\widehat{N}(g) =$

$$\{x | x \in C(M(P(g))) \ni \exists\, y \in L(g) \setminus \mathcal{R}(g), x \geq_S M(y)\}. \tag{3}$$

One more piece of machinery is needed: in order to determine $\widehat{N}(g)$, we must know the children of $M(P(g))$, but we do not have that information until we visit $P(g)$. Therefore, we define a similar set mapping, $\widehat{W} : V_G \to V_S^+$, to aid in the calculation of $\widehat{N}$. $\widehat{W}(g)$ is the set of children of $M(g)$ that vertically inherit a descendant of g. Formally, if $M(g) \in L(T_S)$, $\widehat{W}(g) = \{M(g)\}$; otherwise, $\widehat{W}(g) =$

$$\{x | x \in C(M(g)) \ni \exists\, y \in L(g) \setminus \mathcal{R}(g), x \geq_S M(y)\}. \tag{4}$$

Alg. 2.1.1, traverses T_G in post-order calling `calcCost` at each $g \in V_G$. The challenge in the DTLI model is to determine the sets of species that inherit the descendants of c_1 and c_2 when $M(g) = s$ is a polytomy; i.e., how to calculate $\widehat{N}(c_1)$ and $\widehat{N}(c_2)$. When s is binary, the descendants of s are easily partitioned into two sets; when s is a polytomy, all possible ways to partition the descendants must be considered. Each child of g can be retained in any subset of the children of s, ranging from size 1 to $|C(s)| - 1$. Our DTLI algorithm addresses this by considering all ways of partitioning $C(s)$ into two non-empty subsets.

At each internal node g, the algorithm assesses all possible values for $M(g)$ and $\widehat{W}(g)$ by looping through all $(s_1, s_2) \in V_S \times V_S$ and all $(\widehat{W}_1, \widehat{W}_2) \in C(s_1)^+ \times C(s_2)^+$. Considering all power sets corresponds to considering all the ways to partition $C(s_1)$ and $C(s_2)$. The optimal event and child mapping under s and $\widehat{W}$ is determined by minimizing the cost of the candidate solution at g:

$$\kappa(\varepsilon) + \mathcal{K}_{c_1}[s_1][\widehat{W}_1] + \mathcal{K}_{c_2}[s_2][\widehat{W}_2] + \lambda \cdot (n_{\mathcal{L}}(c_1) + n_{\mathcal{L}}(c_2)) \tag{5}$$

where $n_{\mathcal{L}}(c_i)$, the number of losses on edge (g, c_i), is calculated using the loss heuristic in [29]. Note that for each s, the local cost and history tables are also indexed by all possible values of $\widehat{W}$, which are in $C(s)^+$.

2.2 Temporal Infeasibility

Since the donor and recipient species of any transfer must have co-existed, each transfer implies a temporal constraint. A reconciliation is *temporally feasible* if an ordering of species exists that satisfies the constraints of all inferred transfers. Because reconciliations inferred by Alg. 2.1.1 are not guaranteed to be feasible, each candidate optimal solution is tested for feasibility *post hoc*.

To determine whether a reconciliation R_{GS} is temporally feasible, we construct a directed timing graph $G_t = (V_t, E_t)$ that encodes all temporal constraints on species in T_S. Only species that are the donor, d, or recipient, r, of a transfer edge in $\Lambda(R_{GS})$ must be considered. Thus, the vertex set is defined as $V_t = \{v \in V_S | \exists (g,h) \in \Lambda(T_G) \ni v = M(g) \vee v = M(h)\}$.

The edges in E_t represent three types of temporal constraints:

1. If species s_i is an ancestor of species s_j in T_S, then s_i predates s_j: for every (s_i, s_j) in $V_t \times V_t$, add (s_i, s_j) to E_t iff $s_i \geq_S s_j$.
2. Let (g,h) and (g',h') be transfers in $\Lambda(R_{GS})$, such that $g \geq_G g'$. Then $d = M(g)$ and $r = M(h)$ must have occurred no later than both $d' = M(g')$ and $r' = M(h')$. We add $(P(d), d')$, $(P(d), r')$, $(P(r), d')$, and $(P(r), r')$ to E_t.
3. Given a transfer $(g,h) \in \Lambda(R_{GS})$, species $M(g)$ and $M(h)$ must be contemporaneous. Further, any species that predates $M(g)$ must also predate $M(h)$, and vice versa. For every $(s_i, s_j) \in V_t \times V_t$, add (s_i, s_j) to E_t iff $\exists s_k \in V_t$ such that $s_i \geq_S s_k$ and s_k and s_j are the donor and recipient, or vice versa, of some transfer $(g,h) \in \Lambda(R_{GS})$.

We test each candidate optimal history for temporal feasibility by verifying that the associated timing graph G_t is acyclic, using a modified topological sorting algorithm in $\Theta(|V_t| + |E_t|)$ [5]. Temporally infeasible histories are not reported. Note that it is not the case that if one optimal history is infeasible, all optimal histories are infeasible. Finding the optimal, temporally feasible

Algorithm 2.1.1 DTLI Reconciliation

```
Input: T_G; T_S; M_L
Output: K_g, H_g  ∀g ∈ V_G; π

firstPass(T_G, T_S, M_L) {
 1    for each g ∈ V_G \ L(V_G) in postorder {
 2      for each (s_1, s_2) ∈ V_S × V_S {
 3        for each (Ŵ_1, Ŵ_2) ∈ C(s_1)^+ × C(s_2)^+ {
 4          costCalc(g, s_1, s_2, Ŵ_1, Ŵ_2)
 5        }
 6      }
 7    }
 8    π ← min_{s∈V_S} {K_{ρ_G}[s]}
 9    {(s*, Ŵ*)} ← argmin_{s∈V_S, Ŵ∈C(s)^+} {K_{ρ_G}[s][Ŵ]}
10  }

costCalc(g, s_1, s_2, Ŵ_1, Ŵ_2) {
11    // consider M(g) = lca(s_1, s_2), Ŵ(g) = N̂_1 ∪ N̂_2
12    N̂_1 ← climb(lca(s_1, s_2), Ŵ_1); N̂_2 ← climb(lca(s_1, s_2), Ŵ_2)
13    if (N̂_1 ∩ N̂_2 ≠ ∅) { ε ← D }      // Duplication
14    else if (s_1 ⋦_S s_2) { ε ← S }    // Speciation
15    else { ε ← C }                     // Deep coalescence
16    table(g, lca(s_1, s_2), (N̂_1 ∪ N̂_2), ε, s_1, s_2, Ŵ_1, Ŵ_2, N̂_1, N̂_2)
17    if (s_1 ⋦_S s_2 ∨ (s_1 = s_2 ∧ Ŵ_1 ∩ Ŵ_2 = ∅)) {   // Transfer
18      // consider HGT s_1 to s_2, M(g) = s_1, Ŵ_S = Ŵ_1
19      table(g, s_1, Ŵ_1, T, s_1, s_2, Ŵ_1, Ŵ_2, Ŵ_1, Ŵ_2)
20      // consider HGT s_2 to s_1, M(g) = s_2, Ŵ_S = Ŵ_2
21      table(g, s_2, Ŵ_2, T, s_1, s_2, Ŵ_1, Ŵ_2, Ŵ_1, Ŵ_2)
22    }
23  }

climb(s, Ŵ) {
24    select x ∈ Ŵ at random
25    if (x = s ∨ P(x) = s) { return Ŵ }
26    while (P(x) ≠ s) {
27      x ← P(x); N̂ ← {x}
28    }
29    return N̂
30  }

table(g, s, Ŵ_S, ε, s_1, s_2, Ŵ_1, Ŵ_2, N̂_1, N̂_2) {
31    cost ← κ(ε) + K_{c_1}[s_1][Ŵ_1] + K_{c_2}[s_2][Ŵ_2] + λ·(n_L(c_1) + n_L(c_2))
32    if cost < K_g[s][Ŵ_S] {
33      K_g[s][Ŵ_S] ← cost
34      H_g[s][Ŵ_S] ← (ε, s_1, s_2, Ŵ_1, Ŵ_2, N̂_1, N̂_2)
35    } else if cost = K_g[s][Ŵ_S] {
36      enqueue (ε, s_1, s_2, Ŵ_1, Ŵ_2, N̂_1, N̂_2) to H_g[s][Ŵ_S]
37    }
38  }
```

reconciliation is NP-complete [28]; we leave the problem of obtaining an optimal, feasible solution when all candidate solutions have infeasible timing constraints for future work.

2.3 Complexity and Running Time

Our algorithm is fixed-parameter tractable with polynomial complexity when the size of the largest polytomy, k^*, is fixed. In practical data analyses, k^* is likely to be small. Recent genome-scale analyses of ILS have focused on species trees with $k^* = 3$ [e.g., 10, 22]. In general, event inference will not yield informative results when the species tree is highly unresolved.

THEOREM 2.1. *Given a binary gene tree T_G and a non-binary species tree T_S,* `firstPass` *takes $O(|V_G|(|V_S| + n_k 2^{k^*})^2(h_S + k^*))$ time.*

PROOF. `firstPass` visits each $g \in V_G$ in post order. At each g, `costCalc` is called once for every $(s_1, s_2) \in V_S \times V_S$ and $(\widehat{W}_1, \widehat{W}_2) \in C(s_1)^+ \times C(s_2)^+$, resulting in a total of $O(|V_G|(|\cup_{s \in V_S} C(s)^+|)^2)$ calls to `costCalc`. Because $|C(s)^+| = 2^{|C(s)|}$ is $O(1)$ when s is binary, $|\cup_{s \in V_S} C(s)^+|$ is bounded above by $|V_S| - n_k + n_k 2^{k^*}$, and the number of calls to `costCalc` is $O(|V_G|(|V_S| + n_k 2^{k^*})^2)$. We precalculate $lca(s_1, s_2)$ and test whether $s_1 \not\gtrdot s_2$, for all species pairs, in $O(|V_S|^2)$ time. Therefore, the complexity of `costCalc` is dominated by the calculations of $\widehat{N}$ for l and r, $\widehat{N}(l) \cup \widehat{N}(r)$, and $\widehat{N}(l) \cap \widehat{N}(r)$. These values can be computed in $O(h_S)$, $O(\log(k^*))$ and $O(k^*)$ time, respectively. Thus, each call to `costCalc` has complexity $O(h_S + k^*)$. Once the post order traversal is completed, we extract the minimum score in $\mathcal{K}_{\rho_G}$, and all values of $M(\rho_G)$ and $\widehat{W}(\rho_G)$ corresponding to that score. Since $|\mathcal{K}_{\rho_G}| = |\cup_{s \in V_S} C(s)^+|$, a linear search accomplishes this in $O(|V_S| + n_k 2^{k^*})$ time. Thus, the total complexity is $O(|V_G|(|V_S| + n_k 2^{k^*})^2(h_S + k^*))$. □

THEOREM 2.2. `secondPass` *returns each optimal reconciliation in $O(|V_G|(h_S + k^*))$.*

PROOF. `secondPass` starts from the $M(\rho_G)$ and $\widehat{W}(\rho_G)$ found in `firstPass`. It then constructs an optimal solution by visiting each subsequent $g \in V_G$, assigning mappings and events by looking up values in $\mathcal{H}_g$ in constant time. Losses are inferred in $O(k^* + h_S)$ time [see 29]. Thus, the complexity for returning each optimal history is $O(|V_G|(h_S + k^*))$. □

When T_S is binary, `firstPass` is completed in $O(h_S|V_G||V_S|^2)$ time, and `secondPass` reports each optimal solution in $O(h_S|V_G|)$ time.

Our NOTUNG implementation is efficient in practice. We measured the time required to reconcile 1128 cyanobacterial gene trees with a species tree of size $|V_S| \leq 21$ for all the parameter settings given in Table 1. To assess the effect of polytomy size, we also collapsed edges in the species tree to create a polytomy ranging in size from 2 to 6. The maximum average running time observed on a single AMD Opteron 2.3ghz, 64-bit processor was $\sim .05$ seconds per solution.

3 EMPIRICAL RESULTS

To assess the importance of a four-event model, we implemented our DTLI algorithm in NOTUNG 2.7 and applied it to two phylogenetic data sets in which ILS, HGT, and hybridization have been studied [2, 30]. Because a number of algorithms and software packages do not include losses in the optimization criterion, we sought to assess the impact of this modelling choice. Therefore, we also implemented and applied models excluding losses in the optimization criterion (DT and DTI) models. Except where stated, the trends reported here were observed consistently in both data sets.

The data sets analyzed contain 1128 cyanobacterial gene trees sampled from 11 species (Figs. 2 and S1), and 106 yeast gene trees sampled from 15 species (Fig. S2), respectively. Each gene tree has at most one gene copy per species. In order to assess the impact of our ILS model, for each data set we compared the performance of

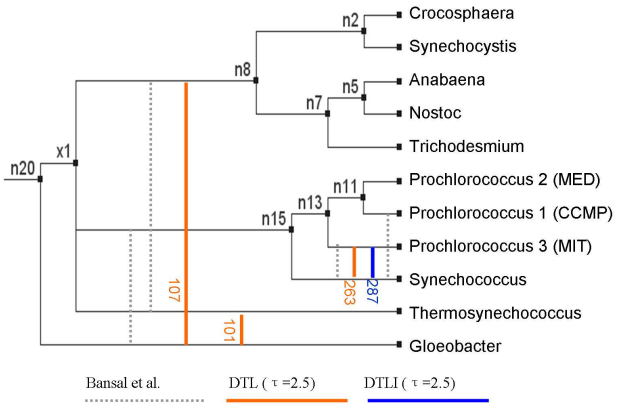


Fig. 2: Predicted transfer highways using the DTL and DTLI models with $\delta = 3$, $\tau = 2.5$ and $\lambda = 2$. The internal edge n16-n18 was collapsed for the DTLI model. Predicted highways with transfer counts greater than 1.5 standard deviations above the mean are shown, with the total number of transfers labeled. Highways predicted by Bansal et al. [2] are shown as dashed lines.

Table 1. Event counts for the cyanobacteria data set, with $\delta = 3$ and $\lambda = 2$.

Model	τ	n_D	n_T	n_L	n_C	Infeasible	Degenerate
DT	2.5	7	1798	1560	0	84	6
DT	6	1648	191	6096	0	0	0
DT	10	2066	0	7520	0	0	0
DTI	2.5	6	1521	1468	559	3	67
DTI	6	1425	133	5133	595	0	0
DTI	10	1691	0	5921	636	0	0
DTL	2.5	0	2121	781	0	42	13
DTL	6	73	1740	1516	0	82	50
DTL	10	1324	480	4797	0	83	40
DTLI	2.5	0	1783	895	409	92	16
DTLI	6	82	1458	1456	542	90	109
DTLI	10	1122	405	4093	602	4	53

Event counts from 314 gene trees with temporally infeasible or conflicting degenerate solutions in any model were removed; the number of trees not considered for each model and setting is given in the last two columns, respectively.

our algorithm on a binary and a non-binary species tree. The non-binary species tree was created by removing one edge resulting in a single polytomy of size 3. In each case, the selected edge was short and associated with substantial gene tree incongruence. Each polytomy was chosen as a reflection of an area of the species tree where ILS may be occurring. In both cases, the selected edge was one that is reportedly difficult to resolve [2, 25, 30].

We reconciled each tree using each of the four models (DT, DTI, DTL, and DTLI), with $\tau \in \{2.5, 6, 10\}$, $\delta = 3$, and $\lambda = 2$ (when considered). We tabulated (1) the number of events of each type, (2) the gene and (3) species lineages in which they occurred, (4) the donor and recipient of each transfer, and (5) the number of temporally infeasible reconciliations (Table 1 for cyanobacteria; Table S3 for yeast). Trees that had no temporally feasible solution for at least one set of parameter values, were eliminated from analysis under all models and values of τ. For each setting, gene trees were rooted with NOTUNG's rooting optimization algorithm using event parsimony. If a tree had multiple optimal solutions (one or more optimal roots or reconciliations for a specified root), it was only retained if all solutions yielded the same counts for each event.

Our observations highlight the extent to which model choice and degeneracy affect biological inferences. Approximately 10% of trees were removed because they are potentially misleading due to temporal infeasibility. Hallett et al. [12] reported no temporal infeasibility for the application of their DT algorithm to a simulated data set. Our results suggest that infeasible cases can be more prevalent in real data.

In addition, approximately 20% of trees had conflicting optimal solution, suggesting that inferences based on a single, randomly selected optimal solution could lead to conclusions that are not, in fact, supported by the data. This result highlights the importance of taking multiple solutions into account when performing tree reconciliation.

When the models with and without ILS are compared, we observed a substantial decrease in the combined number of duplications and transfers, ranging from 15-18% in cyanobacteria and 11-14% in yeast. We also observed considerable decreases in the number of losses, as high as 20% in the case of DT vs DTI. These differences indicate the extent to which ignoring ILS can lead to overestimation of other events.

Recently, great interest has been focused on *highways* of HGT (i.e., pairs of species with very active genetic exchange, relative to HGT in other species) [i.e., 2, 3]. We considered evidence of HGT highways in our cyanobacterial data, where a highway is an outlier in the total number of transfers, in both directions, between a pair of species. With the DTL model, we observe traffic (Fig. 2, red lines) similar to the HGT highways reported by Bansal et al. [2] (dotted lines), for the same data set. However, when events were inferred with the DTLI model, the elevated transfer rates in the Gloeobacter group disappeared, resulting a single highway (blue line). These results demonstrate that use of a complete event model is crucial for accurate inference.

In general, including losses in the optimization criterion resulted in (1) a dramatic decrease in the number of losses, and (2) a change in the ratio of the number of duplications to transfers. This likely occurs because duplications and losses are coupled. When losses are included in the optimization, their cost may prevent the model from over-inferring duplications. This suggests that for any application where accurate reconstruction of event histories matters, including losses in the optimization criterion is crucial.

4 DISCUSSION

This work presents the first reconciliation algorithm for the event inference problem under a model that captures the four major evolutionary processes driving tree incongruence: duplication, loss, transfer, and ILS. Our algorithm reconciles a binary gene tree with a non-binary species tree and is, to our knowledge, the first algorithm to allow non-binary species trees with a transfer model. Our algorithm outputs detailed event histories, describing the specific events inferred and the lineages in which they occurred.

When restricted to binary species trees, our algorithm reduces to an event inference algorithm for the DTL model that can infer all optimal solutions and does not require estimates of speciation times or otherwise restrict transfers to a limited set of species pairs.

Algorithms that capture duplication, transfer, and ILS in a single, integrated model are of increasing importance [7]. New sequencing

technologies are leading to rapid growth of whole genome data sets, in which there is evidence for both HGT and ILS. Our empirical analyses of two different data sets, representing both prokaryotic and eukaryotic data, indicate that use of a complete event model has substantial impact on the events inferred and, hence, the resulting biological conclusions. For example, it is possible that apparent HGT highways could be, at least in part, mis-interpretations of deep coalescence.

Our model is a compromise between current reconciliation models, which ignore ILS everywhere, and coalescent models which explicitly relate the probability of incongruence to the length and population size associated with every branch. Our model is more expressive than the former and more efficient and more widely applicable than the latter. A great strength of the multispecies coalescent is that it explicitly relates the probability of incongruence to effective population size and the time between species divergences. Estimates of these population parameters are only available for a limited set of well studied species. However, given a sufficiently large set of gene families, population parameters can be inferred directly from the data, but this is computationally demanding. For example, species tree inference from a set of 106 genes in 8 yeast species required 800 hours using Bayesian estimation on a coalescent model, where as a parsimony method inferred the identical tree in only a "fraction of a second" [27].

A parsimony model, on the other hand, does not take branch lengths into account, resulting in a potential reduction in accuracy. Future simulation studies are planned to characterize the accuracy of this approach. The benefits of this simpler model are that it can be applied to any set of taxa, not just species for which population parameters can be estimated, and it is not sensitive to overfitting. Because it is fast and general, it is highly suitable for processing large, genome-scale data sets.

The work presented here could profitably be generalized in several ways, including a model of transfers in which multiple genes are transferred in a single event; inference methods for data sets involving extinct or missing species; and ILS models that deviate from the assumption of a uniform gene tree distribution and take branch lengths and population size into account for data sets where such information is available. Another important area for future work is the selection of event costs and investigation of the robustness of results with respect to small changes in the costs used. Note that the problem of how to weight events also arises in coalescent models. For example, the coalescent-based DLI inference algorithm requires the user to supply duplication and transfer rates.

Funding: This work was supported by the National Science Foundation [BDI0641313]; Pittsburgh Supercomputing Center, Biomedical Computing Initiative and Computational Facilities Access grant [MCB000010P]; and a David and Lucille Packard Foundation fellowship. We thank H. Philippe for making his yeast trees available to us.

REFERENCES

[1]J. Andersson. Horizontal gene transfer between microbial eukaryotes. *Methods Mol Biol*, 532:473–487, 2009.

[2]M. Bansal, G. Banay, J. Gogarten, and R. Shamir. Detecting highways of horizontal gene transfer. *J Comput Biol*, 18:1087–1114, Sep 2011.

[3]R. Beiko, T. Harlow, and M. Ragan. Highways of gene sharing in prokaryotes. *Proc Natl Acad Sci U S A*, 102:14332–14337, Oct 2005.

[4]A. Berglund, P. Steffansson, M. Betts, and D. Liberles. Optimal gene trees from sequences and species trees using a soft interpretation of parsimony. *J Mol Evol*, 63(2):240–250, Aug 2006.

[5]T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[6]L. David and E. Alm. Rapid evolutionary innovation during an Archaean genetic expansion. *Nature*, 469:93–96, Jan 2011.

[7]J. Degnan and N. Rosenberg. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends Ecol Evol*, 24:332–340, Jun 2009.

[8]F. Delsuc, H. Brinkmann, and H. Philippe. Phylogenomics and the reconstruction of the tree of life. *Nat Rev Genet*, 6:361–375, May 2005.

[9]J. Doyon, V. Ranwez, V. Daubin, and V. Berry. Models, algorithms and programs for phylogeny reconciliation. *Brief Bioinform*, 12:392–400, Sep 2011.

[10]I. Ebersberger, P. Galgoczy, S. Taudien, S. Taenzer, M. Platzer, et al. Mapping human genetic ancestry. *Mol Biol Evol*, 24:2266–2276, Oct 2007.

[11]S. Edwards. Is a new and general theory of molecular systematics emerging? *Evolution*, 63:1–19, Jan 2009.

[12]M. Hallett, J. Lagergren, and A. Tofigh. Simultaneous identification of duplications and lateral transfers. In *RECOMB 2004: Proceedings of the Eigth International Conference on Research in Computational Biology*, ACM Press, pp. 347–356, New York, NY, USA, 2004. San Diego, California, USA, ACM.

[13]D. H. Huson and C. Scornavacca. A survey of combinatorial methods for phylogenetic networks. *Genome Biol Evol*, 3:23–35, Nov 2011.

[14]B. Ma, M. Li, and L. Zhang. From gene trees to species trees. *SIAM J. Comput.*, 30(3):729–752, 2000.

[15]W. Maddison. Gene trees in species trees. *Syst. Biol.*, 46(3):523–536, 1997.

[16]W. Maddison and L. Knowles. Inferring phylogeny despite incomplete lineage sorting. *Syst Biol*, 55(1):21–30, Feb 2006.

[17]W. Maddison and D. Maddison. Mesquite: A modular system for evolutionary analysis, version 2.75. http://mesquiteproject.org. 2011.

[18]M. Milinkovitch, R. Helaers, E. Depiereux, A. Tzika, and T. Gabaldón. 2x genomes–depth does matter. *Genome Biol*, 11:R16, Feb 2010.

[19]L. Nakhleh. Evolutionary phylogenetic networks: models and issues. In L Heath and N Ramakrishnan, editors, *The Problem Solving Handbook for Computational Biology and Bioinformatics*, pp. 125–158. Springer, 2010.

[20]L. Nakhleh, D. Ruths, and H. Innan. Gene trees, species trees, and species networks. In R. Guerra and D. Goldstein, editors, *Meta-analysis and Combining Information in Genetics and Genomics*, pp. 275–293. CRC Press, 2009.

[21]R. Page. GeneTree: comparing gene and species phylogenies using reconciled trees. *Bioinformatics*, 14(9):819–20, 1998.

[22]D. Pollard, V. Iyer, A. Moses, and M. Eisen. Widespread discordance of gene trees with species tree in *Drosophila*: evidence for incomplete lineage sorting. *PLoS Genet*, 2(10):e173, Oct 2006.

[23]M. Rasmussen and M. Kellis. Unified modeling of gene duplication, loss, and coalescence using a locus tree. *Genome Res*, Feb 2012.

[24]A. Rokas, B. Williams, N. King, and S. Carroll. Genome-scale approaches to resolving incongruence in molecular phylogenies. *Nature*, 425:798–804, Oct 2003.

[25]B. Schirrmeister, A. Antonelli, and H. Bagheri. The origin of multicellularity in cyanobacteria. *BMC Evolutionary Biology*, 11(1):45, 2011.

[26]M.H. Serres, A.R.W. Kerr, T.J. McCormack, and M. Riley. Evolution by leaps: gene duplication in bacteria. *Biol Direct*, 4:46, Nov 2009.

[27]C. Than and L. Nakhleh. Species tree inference by minimizing deep coalescences. *PLoS Comput Biol*, 5:e1000501, Sep 2009.

[28]A. Tofigh, M. Hallett, and J. Lagergren. Simultaneous identification of duplications and lateral gene transfers. *TCBB*, 8:517–535, Mar/Apr 2011.

[29]B. Vernot, M. Stolzer, A. Goldman, and D. Durand. Reconciliation with non-binary species trees. *J Comput Biol*, 15:981–1006, Oct 2008.

[30]Y. Yu, C. Than, J. Degnan, and L. Nakhleh. Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting. *Syst Biol*, Jan 2011.

[31]L. Zhang. From gene trees to species trees ii: Species tree inference by minimizing deep coalescence events. *IEEE/ACM Trans Comput Biol Bioinform*, Apr 2011.

[32]O. Zhaxybayeva and W. Doolittle. Lateral gene transfer. *Curr Biol*, 21:R242–R246, Apr 2011.

[33]O. Zhaxybayeva, W. Doolittle, R. Papke, and J. Gogarten. Intertwined evolutionary histories of marine *Synechococcus* and *Prochlorococcus marinus*. *Genome Biol Evol*, 1:325–339, Sep 2009.

[34]C. Zmasek and S. Eddy. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, 17(9):821–8, Sep 2001.

S1 SUPPLEMENTARY INFORMATION

S1.1 Empirical Input and Output

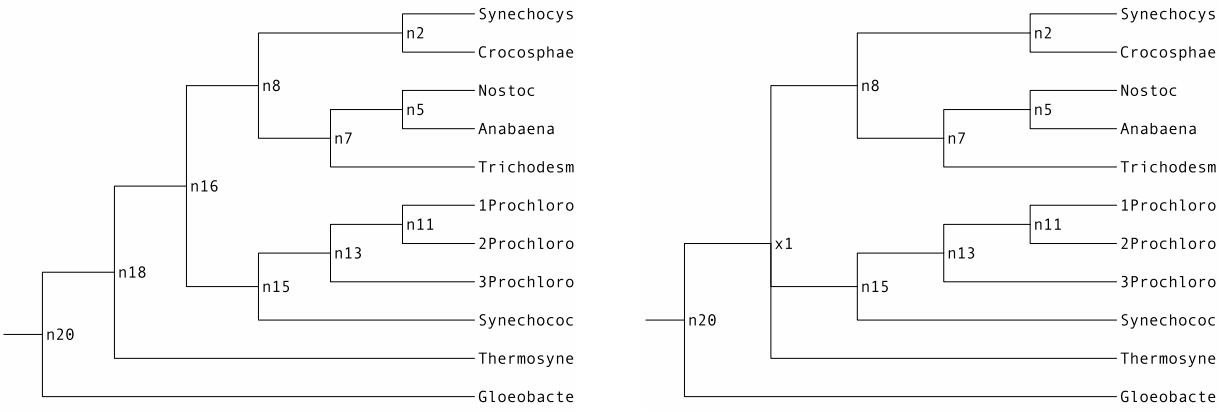
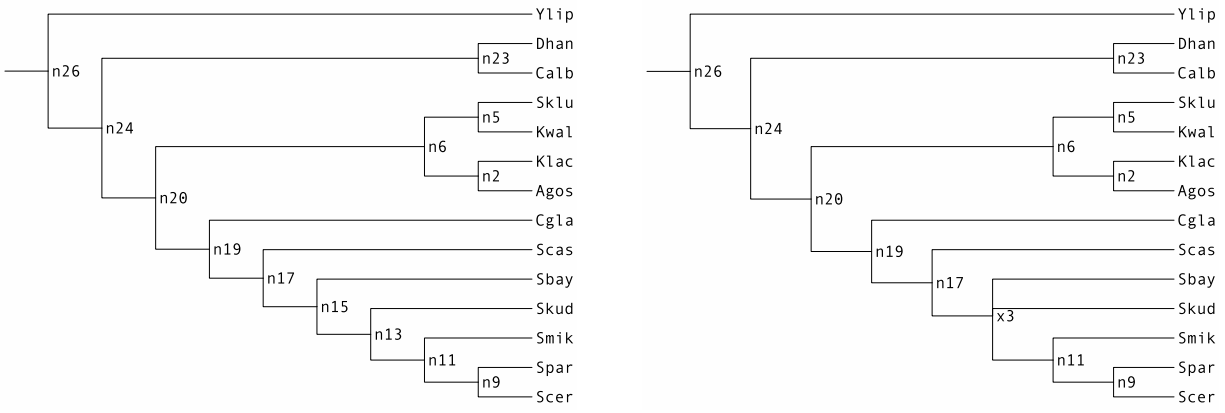


Fig. S1: The *(a)* binary and *(b)* non-binary species tree for the 11 cyanobacterial species. Only tree topologies, not branch lengths, are shown.

Table S1. Strain names for cyanobacterial species.

Short name	Long name
Synechocys	Synechocystis
Crocosphae	Crocosphaera
Nostoc	Nostoc
Anabaena	Anabaena
Trichodesm	Trichodesmium
1Prochloro	Prochlorococcus 1 (CCMP)
2Prochloro	Prochlorococcus 2 (MED)
3Prochloro	Prochlorococcus 3 (MIT)
Synechococ	Synechococcus
Thermosyne	Thermosynechococcus
Gloeobacte	Gloeobacter



(a) (b)

Fig. S2: The *(a)* binary and *(b)* non-binary species tree for the 15 yeast species. Only tree topologies, not branch lengths, are shown.

Table S2. Names for yeast species.

Short name	Long name
Ylip	Yarrowia lipolytica
Dhan	Debaryomyces hansenii
Calb	Candida albicans
Sklu	Saccharomyces kluyveri
Kwal	Kluyveromyces waltii
Klac	Kluyveromyces lactis
Agos	Ashbya gossypii
Cgla	Candida glabrata
Scas	Saccharomyces castellii
Sbay	Saccharomyces bayanus
Skud	Saccharomyces kudriavzevii
Smik	Saccharomyces mikatae
Spar	Saccharomyces paradoxus
Scer	Saccharomyces cerevisiae

Table S3. Event counts for the yeast data set, with $\delta = 3$ and $\lambda = 2$.

Model	τ	n_D	n_T	n_L	n_C	Infeasible	Degenerate
DT	2.5	1	207	192	n/a	3	1
DT	6	192	26	684	n/a	0	0
DT	10	245	0	841	n/a	0	0
DTI	2.5	8	172	180	67	4	11
DTI	6	162	25	568	69	0	0
DTI	10	213	0	720	72	0	0
DTL	2.5	0	233	138	n/a	4	1
DTL	6	6	203	192	n/a	3	1
DTL	10	155	53	563	n/a	0	11
DTLI	2.5	0	208	115	62	4	12
DTLI	6	10	172	172	66	2	13
DTLI	10	138	42	493	67	1	10

Event counts from 31 gene trees with temporally infeasible or conflicting degenerate solutions in at least one model were removed; the number of such trees for each model is shown in the last two columns, respectively.

S1.2 Heatmaps Describing Transfers Between Species

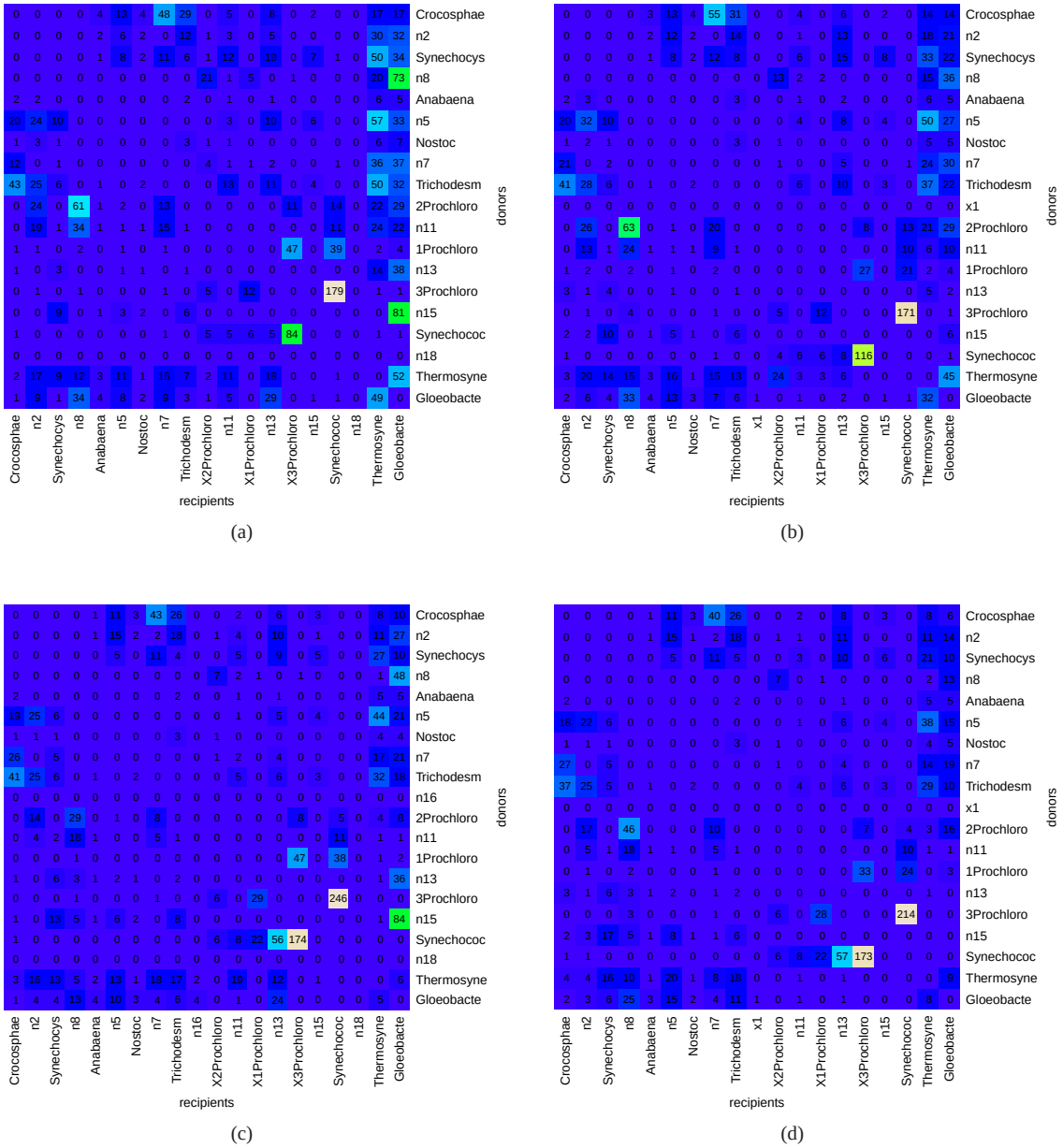
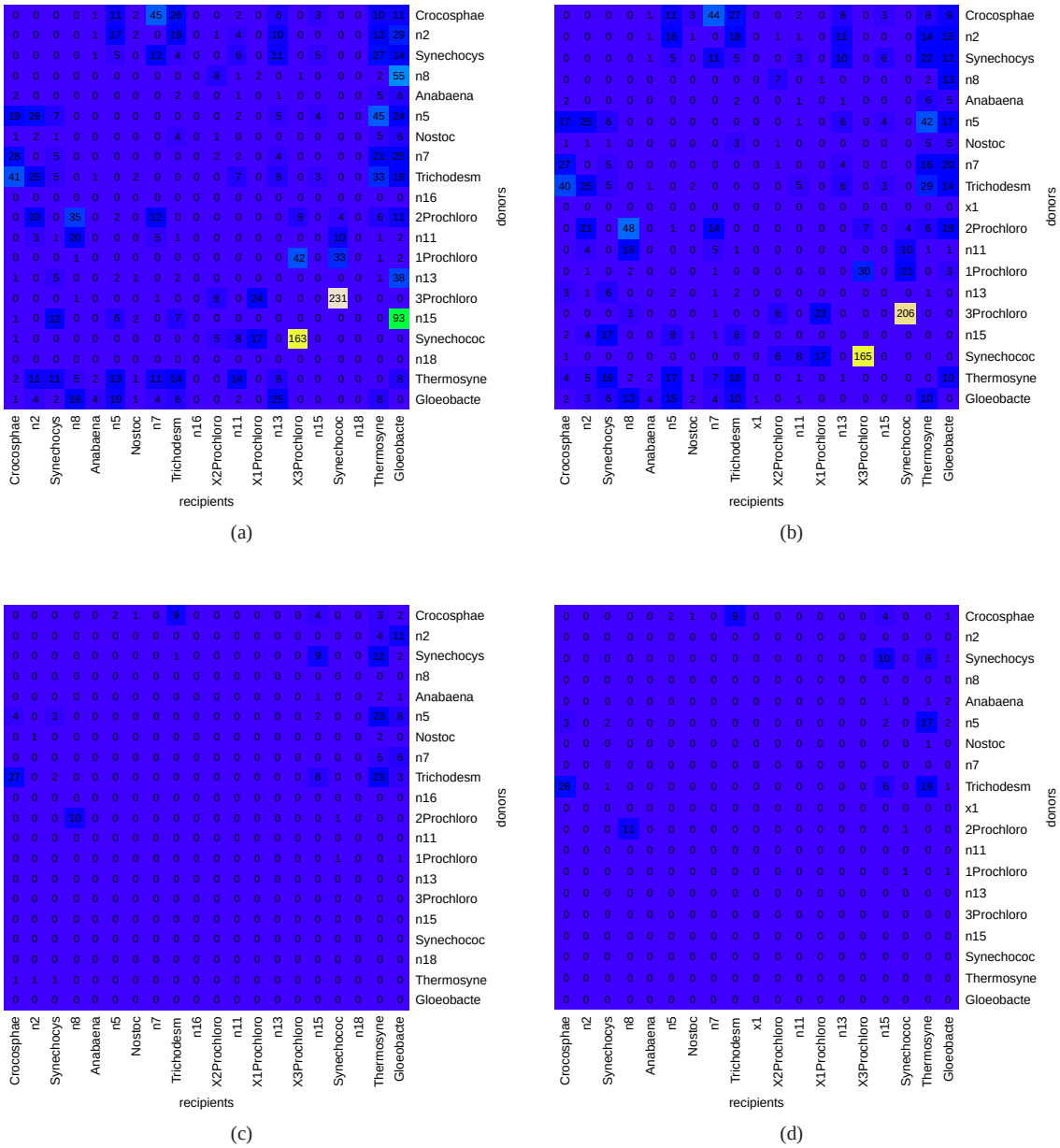


Fig. S3: Transfers in cyanobacteria, inferred with $\delta = 3$, $\lambda = 2$ and $\tau = 2.5$ under *(a)* the DTL-model, *(b)* DTLI-model, *(c)* the DT-model *(d)* DTI-model



(a) (b)

(c) (d)

Fig. S4: Transfers in cyanobacteria, inferred with $\delta = 3$, $\lambda = 2$ and $\tau = 6$ under *(a)* the DTL-model, *(b)* DTLI-model, *(c)* the DT-model *(d)* DTI-model

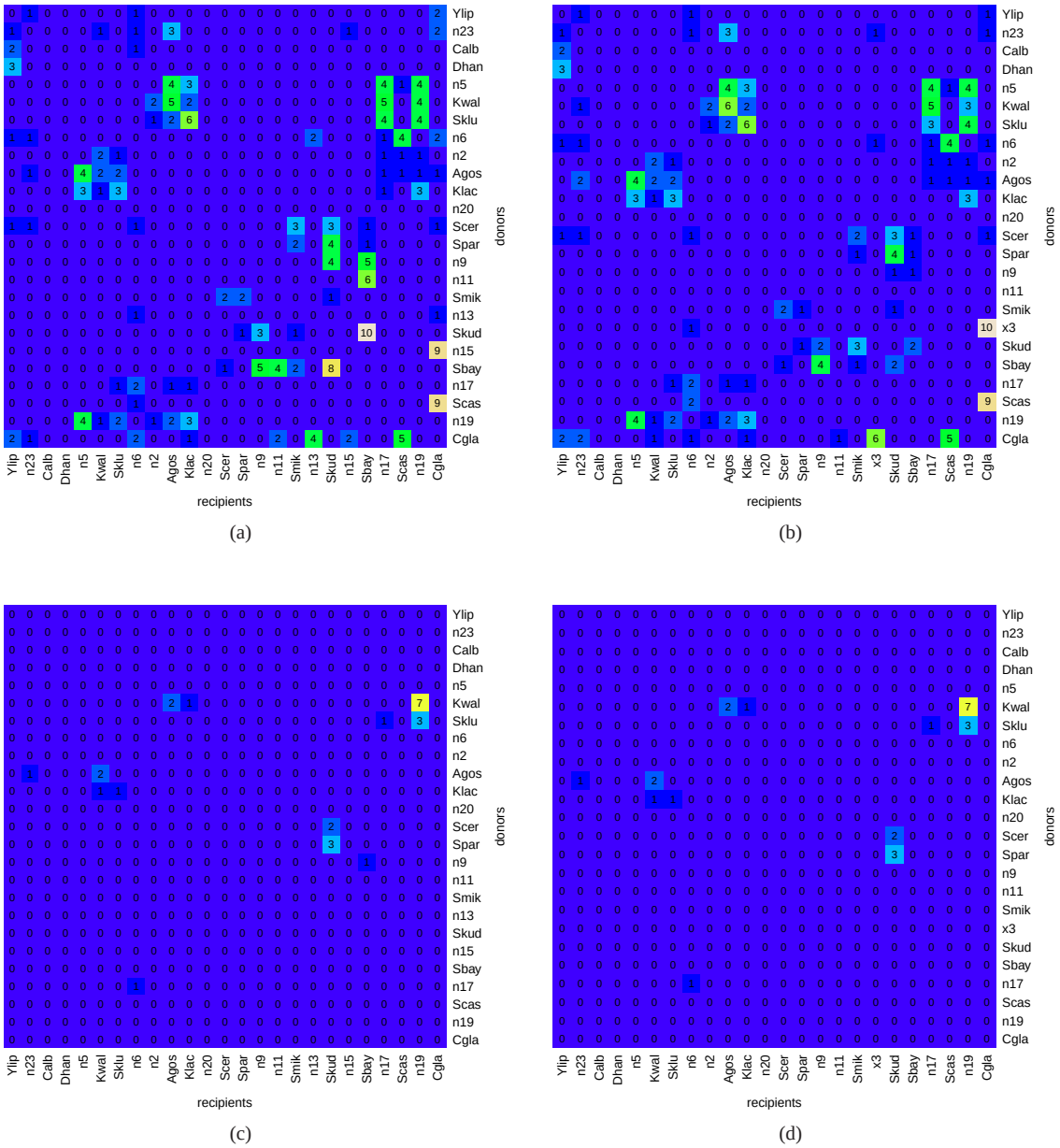
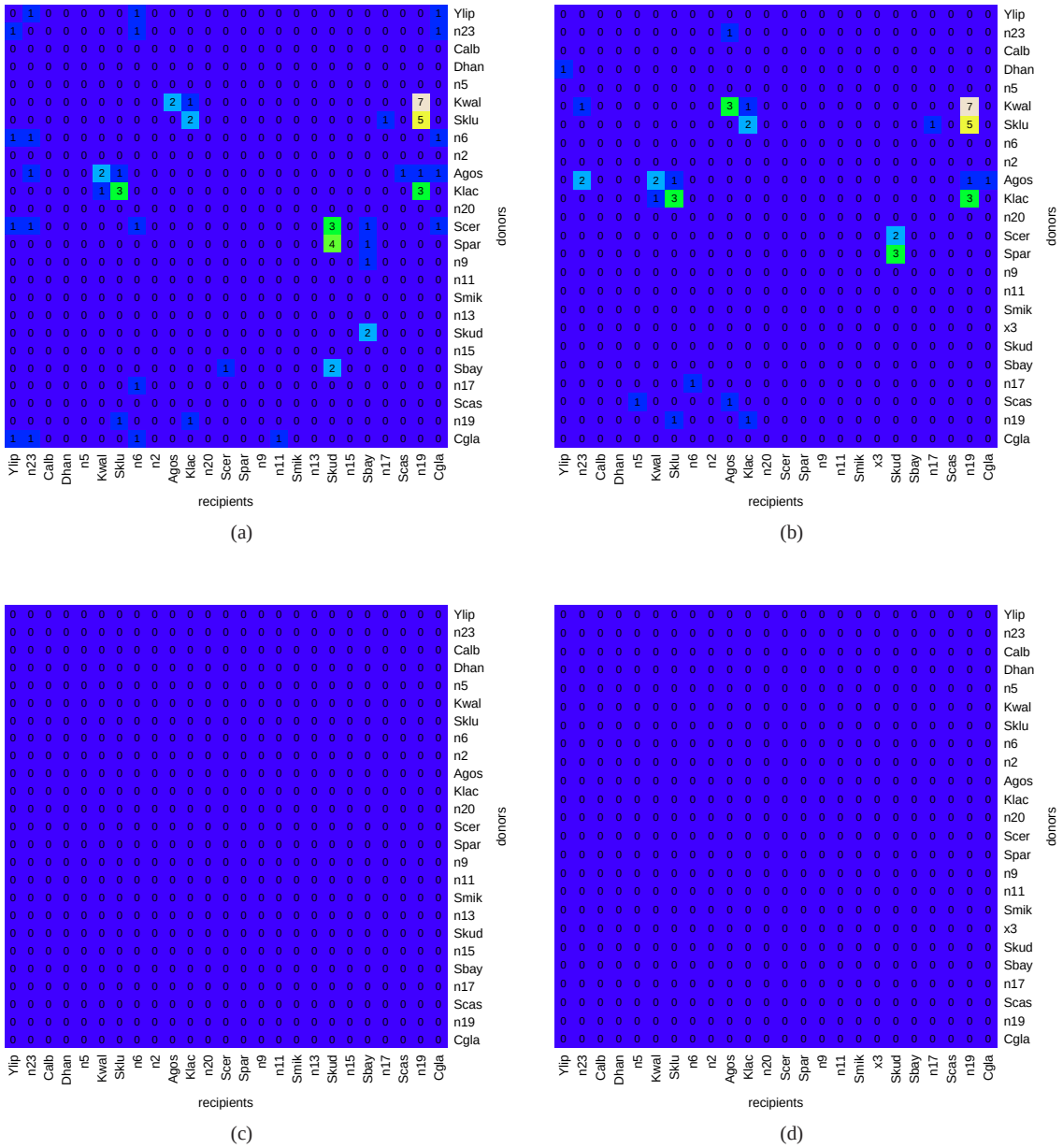


Fig. S5: Transfers in yeast, inferred with $\delta = 3$, $\lambda = 2$ and $\tau = 6$ under *(a)* the DTL-model, *(b)* DTLI-model, *(c)* the DT-model *(d)* DTI-model



(a) (b) (c) (d)

Fig. S6: Transfers in yeast, inferred with $\delta = 3$, $\lambda = 2$ and $\tau = 10$ under *(a)* the DTL-model, *(b)* DTLI-model, *(c)* the DT-model *(d)* DTI-model

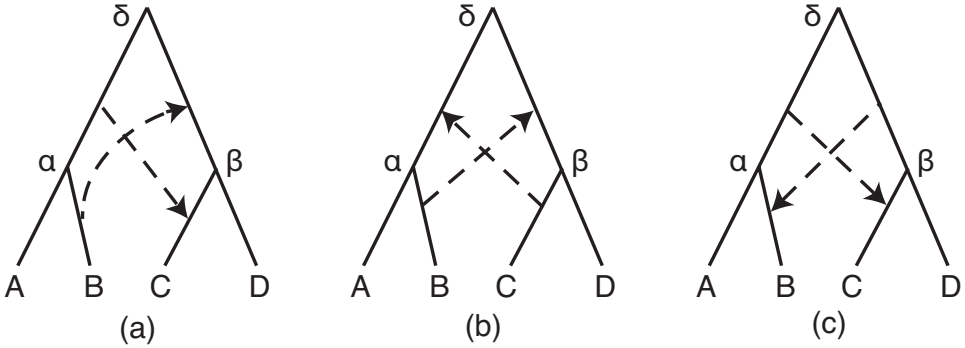


Fig. S7: *(a-c)* Three examples of temporally infeasible transfer pairs on a hypothetical species tree with four leaves. Dashed arrows correspond to inferred transfers.

S1.3 Traceback Algorithm, Multiple Optimal Solutions, and Temporal Feasibility

Our event inference algorithm infers a most parsimonious event history in three passes. Pass 1, described in Alg. 2.1.1 in the main text, populates the cost and event tables for each node ($\mathcal{K}_g$ and $\mathcal{H}_g$, respectively, $\forall g \in T_G$) in a post-order traversal of the gene tree. Pass 1 returns the minimum event cost at the root, ρ_G, and the set of all pairs

$$\{(s^*, \widehat{W}^*)\} = \underset{s \in V_S; \widehat{W} \in C(s)^+}{\operatorname{argmin}} \{\mathcal{K}_{\rho_G}[s, \widehat{W}]\} \tag{6}$$

that give that minimum cost. Recall that each s^* is a node in the species tree that, when assigned to ρ_G, results in an optimal score and that $\widehat{W}^*$ is the set of children of s^* that inherent a descendant of ρ_G in this event history.

In Pass 2, the traceback algorithm (Alg. S1.3.1) uses the information in the tables populated in Pass 1 to generate all candidate optimal reconciliations. These minimum cost histories are only *candidate* optimal reconciliations because they are not guaranteed to be temporally feasible. To obtain temporally feasible, optimal reconciliations, the candidates will be checked for temporal inconsistencies in Pass 3.

The set of all candidate optimal reconciliations, $\{R_{GS}\}$, is generated by recursively enumerating event histories in a series of pre-order traversals of T_G. At ρ_G, there may be more than one $(s^*, \widehat{W}^*)$ pair that yields a minimum cost event history. For a given $(s^*, \widehat{W}^*)$, descendants of ρ_G may also have more than one optimal $(s, \widehat{W})$ pair. For internal nodes other than ρ_G, we say a $(s, \widehat{W})$ pair is optimal if it leads to a minimum cost history at ρ_G; $(s, \widehat{W})$ may not result in the lowest cost at g. The complete set of minimum cost histories for a given $(s^*, \widehat{W}^*)$ pair is generated by considering all ways of combining an optimal history for the right child of ρ_G and an optimal for the left child of ρ_G. This strategy is applied recursively during the pre-order traversal. The final set of optimal candidate reconciliations is the union over all $(s^*, \widehat{W}^*)$ pairs in $\mathcal{K}_{\rho_G}$.

In `secondPass`, all $(s^*, \widehat{W}^*)$ pairs associated with ρ_G are enqueued into `solutionQueue`, which stores event histories that have not yet been explored. The enumeration then proceeds by calling `traceback` for an entry in `solutionQueue`. This process begins at the root with an unlabeled copy of the gene tree, $R_{GS} = T_G$. The nodes of R_{GS} are labeled with the species mapping, event, and losses as it is passed down the tree during the traversal, resulting in a fully labeled candidate reconciliation when the traversal is complete. When `traceback` is called on node g, it receives a partial reconciliation $\mathcal{R}_{GS}$ from its parent. In `traceback`, $\mathcal{R}_{GS}$ is augmented by initializing the mapping $M(g)$, the event $\mathcal{E}(g)$, and the losses at g. The species in which losses occurred are inferred using the heuristic approach described in [29]. This procedure is exact when the corresponding species node, $M(g)$, has less than four children. After the labeling steps at g are complete, an optimal event history for $T_G(g)$ is obtained by recursive calls to the children of g. If there is more than one $(s, \widehat{W})$ pair at g that will yield an optimal event history at ρ_G, the remaining pairs are enqueued into `solutionQueue` for later processing.

In Pass 3, each candidate optimal reconciliation is tested for temporal feasibility. As described in Section 2.2 of the main text, a graph is constructed that encodes all temporal constraints imposed by the species tree and the set of inferred transfers. If the graph is acyclic, then the event history contains no temporal constraint violations and is a legitimate reconciliation. The constraints we propose in Section 2.2 are appropriate for inferring detailed event histories and for counting the number of optimal solutions and can be used with all four event models (DT, DTI, DTL, and DTLI). Our approach builds on a feasibility checking scheme proposed by Tofigh et al. [28] for the less restrictive DT model, but imposes additional constraints. For example, all three scenarios in Fig. S7 are temporally infeasible and all three violate the constraints introduced in Section 2.2. Only Fig. S7b violates the constraints introduced in [28].

Tofigh's constraints are appropriate for applications where only the *number* of duplications and transfers is inferred (e.g., for scoring trees for the tree inference problem), but the specific events are not of interest. Under the DT model, there exist certain event histories that have a cycle, but for which a temporally feasible reconciliation can be identified that has the same number of duplications and transfers, but more losses. Since losses incur no cost under the DT model, this feasible reconciliation has the same cost as the original, infeasible history. For example, the infeasible histories in Figs. S7a and c can be converted into feasible histories by lifting the transfer recipient on (β,*C*) so that it enters the edge (δ,β), above the other transfer. Note that this operation generates a new history that is temporal feasibility but incurs an additional loss in species *D*. For the purposes of inferring only the optimal reconciliation cost, it is not necessary to construct the feasible history, but simply to verify that it exists.

Therefore, the constraints proposed by Tofigh et al. do not rule out infeasible histories for which there exists a feasible history with the same DT cost. This is sufficient if only the cost is of interest, but insufficient for applications where the goal is to infer the donor and recipient species of specific transfers. Similarly, the Tofigh constraints are not appropriate for counting the number of optimal solutions, because the set of optimal solutions reported under these constraints will include infeasible solutions like those in Figs. S7a and c, as well as their feasible counterparts. This will lead to an overestimate of the number of valid, optimal reconciliations.

Algorithm S1.3.1 DTLI Reconciliation Traceback

```
Input: T_G; T_S; K_g, H_g ∀g ∈ V_G; s*, Ŵ*
Output: R_GS, the set of all optimal R_GS

secondPass(T_G, T_S, K_g, H_g ∀g ∈ V_G, s*, Ŵ*) {
 1   R_GS ← { }
 2   for each (s*, Ŵ*) {
 3     enqueue (T_G, ρ_G, s*, Ŵ*, ∅) to solutionQueue
 4   }
 5   while solutionQueue is not empty {
 6     dequeue (R_GS, g, s, Ŵ, N̂) from solutionQueue
 7     R*_GS = traceback(R_GS, g, s, Ŵ, N̂)
 8     add R*_GS to R_GS
 9   }
10 }

traceback(R_GS, g, s, Ŵ, N̂) {
11   M(g) ← s; Ŵ(g) ← Ŵ; N̂(g) ← N̂
12   dequeue {ε, s_1, s_2, Ŵ_1, Ŵ_2, N̂_1, N̂_2} from H_g[s][Ŵ]
13   E(g) ← ε
14   if (g ≠ ρ_G) {
15     L(g) ← inferLosses(E(P(g)), s, M(P(g)), Ŵ(g), Ŵ(P(g)), N̂(g))
16   }
17   if (g ∉ L(R_GS)) {
18     R_GS = traceback(R_GS, l(g), s_1, Ŵ_1, N̂_1); R_GS = traceback(R_GS, r(g), s_2, Ŵ_2, N̂_2)
19     while K_g[s][Ŵ(g)] is not empty {
20       dequeue {ε, s_1, s_2, Ŵ_1, Ŵ_2, N̂_1, N̂_2} from H_g[s][Ŵ(g)]
21       enqueue (R_GS, l(g), s_1, Ŵ_1, N̂_1) to solutionQueue
22       enqueue (R_GS, r(g), s_2, Ŵ_2, N̂_2) to solutionQueue
23     }
24   }
25   return reconciliation R_GS
26 }

inferLosses(E(P(g)), M(g), M(P(g)), Ŵ(g), Ŵ(P(g)), N̂(g)) {
27   losses= ∅
28   if (M(g) ∉ L(T_S) ∧ M(g) ≠ M(P(g))) { // polytomy loss
29     losses+ = C(M(g)) \ Ŵ(g)
30   }
31   select x ∈ Ŵ(g) at random
32   while (P(x) ≠ M(P(g))) { // skipped species losses
33     if (P(x) ≠ M(g)) { losses+ = C(P(x)) \ x }
34     x = P(x) // climb
35   }
36   if (E(P(g)) = D ∧ Ŵ(P(g)) \ N̂(g) ≠ ∅) { // dup loss
37     losses+ = Ŵ(P(g)) \ N̂(g)
38   }
39   return
40 }
```