INFERRING OPTIMAL SPECIES TREES UNDER GENE DUPLICATION AND LOSS

M. S. BAYZID, S. MIRARAB and T. WARNOW*

Department of Computer Science, The University of Texas at Austin, Austin, Texas 78712, USA

**E-mail: tandy@cs.utexas.edu*

www.cs.utexas.edu/users/tandy

Species tree estimation from multiple markers is complicated by the fact that gene trees can differ from each other (and from the true species tree) due to several biological processes, one of which is gene duplication and loss. Local search heuristics for two NP-hard optimization problems - minimize gene duplications (MGD) and minimize gene duplications and losses (MGDL) - are popular techniques for estimating species trees in the presence of gene duplication and loss. In this paper, we present an alternative approach to solving MGD and MGDL from rooted gene trees. First, we characterize each tree in terms of its "subtree-bipartitions" (a concept we introduce). Then we show that the MGD species tree is defined by a maximum weight clique in a vertex-weighted graph that can be computed from the subtree-bipartitions of the input gene trees, and the MGDL species tree is defined by a minimum weight clique in a similarly constructed graph. We also show that these optimal cliques can be found in polynomial time in the number of vertices of the graph using a dynamic programming algorithm (similar to that of Hallett and Lagergren[1]), because of the special structure of the graphs. Finally, we show that a constrained version of these problems, where the subtree-bipartitions of the species tree are drawn from the subtree-bipartitions of the input gene trees, can be solved in time that is polynomial in the number of gene trees and taxa. We have implemented our dynamic programming algorithm in a publicly available software tool, available at `http://www.cs.utexas.edu/users/phylo/software/dynadup/`.

Keywords: Gene Duplication and Loss; Incomplete Lineage Sorting; Clique.

1. Introduction

The estimation of species trees typically proceeds by concatenating multiple sequence alignments together for many genes and then estimating a tree on the resultant "super-matrix". These "combined analyses" require that all sequences be orthologous (hence each taxon should appear in each gene sequence alignment at most once), and assume that the true trees for the different genes are topologically identical. These two conditions can easily fail to hold when gene duplication and loss occurs, even when valiant efforts are made to estimate orthology. Thus, the estimation of species trees from gene trees that can differ due to gene duplication and loss,[2–6] especially when these gene trees contain more than a single copy of each taxon, requires more care.

Two of the most popular approaches for species tree estimation in the presence of gene duplication and loss are methods, such as iGTP[7] and DupTree,[8] that employ local search techniques to "solve" the NP-hard optimization problems MGD (Minimize Gene Duplication) and MGDL (Minimize Gene Duplication and Loss). For example, analyses based upon MGD and MGDL have been used in estimating species trees for snakes,[9] vertebrates,[10,11] *Drosophia*,[12] and plants.[13] These local search strategies are effective for relatively small numbers of taxa, but their utility for very large numbers of taxa has not been explored. In addition to local

search techniques, exact solutions[14,15] and fixed-parameter tractable algorithms[1,16] have been proposed for addressing MGD and MGDL; however, to date these approaches have not been used as widely as the heuristic searches.

In this paper we will present a new approach for MGD and MGDL that does not use local search techniques or branch-and-bound techniques, but instead uses dynamic programming to produce an optimal solution within a user-specified subspace of the set of candidate species trees. Thus, by letting that subspace be all possible species trees we obtain a globally optimal solution for MGD or MGDL, while constraining the set allows us to obtain good (even if not globally optimal) solutions in polynomial time. While our dynamic programming approach is similar to that of Hallet and Lagergren,[1] our clique-based formulation of the problem is new, and many of our theoretical results are not explicitly proven in Hallett and Lagergren.[1]

The algorithmic technique we present is also related to the approach used in Than and Nakhleh[17] (see also Yu, Warnow, and Nakhleh[18]) for the MDC (Minimize Deep Coalescence) problem,[5] an optimization problem for species tree estimation in the presence of incomplete lineage sorting. In these papers, the optimal solution for MDC is characterized graph-theoretically, as follows. First, every binary rooted tree on n taxa can be represented by its set of "clusters", where a cluster is the set of taxa that appear below a node in the tree. Furthermore, two clusters are said to be "compatible" if and only if they can co-exist in a tree (equivalently, two clusters are compatible if and only if they are pairwise disjoint or one contains the other). To solve MDC, each possible cluster is represented by a node in a graph, and edges exist between pairs of nodes whose clusters are compatible. It is known that whenever a set of clusters is given that are all pairwise compatible, then a rooted tree exists with precisely that set of clusters. Thus, a set of $n-1$ pairwise compatible clusters, where n is the number of species, defines a binary rooted species tree for that set of clusters.

Than and Nakhleh[17] showed that it is possible to weight the nodes in the graph so that the total weight of any $(n-1)$-clique is the MDC score for the species tree defined by that clique, so that solving the MDC problem is equivalent to finding a minimum weight $n-1$ clique.

This problem formulation seems to be particularly expensive, since MaxClique is NP-hard and the graph has an exponential number of vertices, but Than and Nakhleh also showed that finding the minimum weight clique of size $n-1$ can be obtained in time that is polynomial in the number of nodes in the graph, using dynamic programming (DP). They also presented a "heuristic" version that only uses clusters that appear in the input gene trees, and so runs in polynomial time. This heuristic version produces highly accurate species trees,[17–19] suggesting that restricting the search space to clusters in the input trees is an effective strategy for MDC.

The approach we present here for optimizing MGD or MGDL builds on these ideas. We also build a graph, but the nodes of our graph correspond to "subtree-bipartitions", a generalization of clusters that we define in this paper. We show how to define weights on vertices in the graph so that the optimal solution to MGD is obtained by finding a minimum weight clique of size $n-1$, and we show how to find that clique using dynamic programming. This technique directly allows us to solve the constrained MGD problem, in which we constrain the species tree solution to have its subtree-bipartitions from a user-provided set; as with MDC, a DP algorithm solves this in polynomial time. We then show how to extend this to the MGDL

problem, using the same graph but with different weights on the edges.

The rest of the paper is organized as follows. In Section 2, we present the theoretical foundations and terminology. We present theory and algorithms for solving MGD in Section 3, and results for MGDL in Section 4.

2. Basics

2.1. *Prior Terminology and Theory*

We begin by defining the MGD, MGDL, and MDC problems. The input to each problem is the same: a set $\mathcal{G} = \{t_1, t_2, \ldots, t_k\}$ of rooted binary gene trees, with leaves drawn from the set $\mathcal{X}$ of n taxa, and we allow the gene trees to have multiple copies of the taxa, and even to miss some taxa. The output of each problem is a species tree T on $\mathcal{X}$ minimizing $\sum_i d(t_i, T)$, where $d(t_i, T)$ is defined differently for each problem.

The original definitions for these problems assumed that the gene tree t_i had at least one copy of each taxon, and so these definitions need to be modified in order to handle incomplete gene trees, which have no copies of some taxon.

Handling incomplete gene trees: Most of the literature has handled the case of incomplete gene trees t_i as follows. Let T' be the tree obtained by restricting T to the leaf set of t_i and then suppressing all non-root nodes of degree two (i.e., T' is the homeomorphic subtree of T defined on the leafset of t_i). Then, T' is used instead of T when computing the MDC, MGD, or MGDL score. We call this the *restriction*-based approach, and hence define the restriction-based optimization problems MGD_r, $MGDL_r$, and MDC_r. (See Bayzid and Warnow[20] for another approach for handling incomplete gene trees.)

Optimal Embeddings for $MGD_r, MDGL_r,$ ***and*** MDC_r**.**

An embedding of a rooted gene tree t into a species tree T is a mapping f from the nodes of the gene tree to the nodes of the species tree that has some natural properties: first, f maps leaves in the gene tree mapped to the unique leaf in the species tree with the same taxon label, and second, f maintains the order relationships in the gene tree. This second condition can be stated as follows: if v and w are nodes in the gene tree with v above w (meaning that v is on the path from w to the root of the gene tree), then $f(v)$ is above $f(w)$ within the species tree.

Let T be a rooted binary tree. We denote the set of vertices of a tree T by $V(T)$, the root by $root(T)$, the internal nodes by $V_{int}(T)$, and the set of taxa that appear at the leaves by $L(T)$. (Note that since T can have multiple copies of some taxa, it is possible for $|L(T)|$ to be smaller than the number of leaves in T.)

A *clade* in T is a subtree of T rooted at some node in T, and the set of leaves of the clade is called a *cluster*. We denote the cluster at v by $c_T(v)$; however, when the tree T is understood, we may also write $c(v)$. We denote the set of clusters of a tree T by $C(T)$.

The most recent common ancestor (MRCA) of a set A of leaves in T is denoted by $MRCA_T(A)$. Given a gene tree gt and a species tree ST, where $L(gt) \subseteq L(ST)$, we define $\mathcal{M} : V(gt) \rightarrow V(ST)$ by $\mathcal{M}(v) = MRCA_{ST}(c_{gt}(v)))$. In other words, $\mathcal{M}$ associates each node u of gt to the MRCA in ST of the cluster below u.

The optimal embedding for each of the three criteria we discuss (MDC_r, MGD_r, and

$MGDL_r$) is obtained using $\mathcal{M}$, even when the gene tree gt is incomplete (lacks some taxon) or contains more than one copy of some taxon.[5,6,17,21] Therefore, since the same reconciliation of a gene tree into a species tree optimizes all three criteria, we may refer to an "optimal reconciliation" without specifying the criterion. Also, for any given mapping, the calculation of the three scores can be performed in polynomial time. Therefore, given a set of rooted gene trees and a rooted species tree, we can calculate the $MGD_r, MGDL_r$, and MDC_r scores of the species tree in polynomial time.

Duplication nodes: For a rooted gene tree gt and a rooted species tree ST, where $L(gt) \subseteq L(ST)$, an internal node v in gt is called a *duplication node* if $\mathcal{M}(v) = \mathcal{M}(v')$ for some child v' of v, and otherwise v is a *speciation node*.[21–24]

Given a rooted, binary gene tree gt and a rooted, binary species tree ST such that $L(gt) \subseteq L(ST)$, $Dup(gt, ST)$ denotes the number of duplications needed to reconcile gt with ST under the $\mathcal{M}$ mapping. For a set $\mathcal{G}$ of rooted, binary gene trees, the notation $Dup(\mathcal{G}, ST)$ extends in the obvious way.

Gene losses: Let gt be a rooted, binary gene tree and ST a rooted, binary species tree such that $L(gt) \subseteq L(ST)$. The restriction of ST to $L(gt)$, denoted by $\mathcal{R}_{ST}(L(gt))$, is the smallest subtree of ST containing $L(gt)$ as its leaf set. The homeomorphic subtree $ST|_{L(gt)}$ of ST induced by $L(gt)$ is a tree obtained from $\mathcal{R}_{ST}(L(gt))$ by suppressing all nodes of $\mathcal{R}_{ST}(L(gt))$ with indegree and outdegree 1. We denote by r and l the two children of an internal node u. Then the number of gene losses for a given gene tree gt and species tree ST for a particular internal node u (under the restriction-based analysis), denoted by $loss_u$, can be calculated as follows:[21–24]

$$loss_u = \begin{cases} d(\mathcal{M}(r), \mathcal{M}(u)) + 1 & \text{if } \mathcal{M}(r) \subsetneq \mathcal{M}(u) = \mathcal{M}(l), \\ d(\mathcal{M}(r), \mathcal{M}(u)) + d(\mathcal{M}(l), \mathcal{M}(u)) & \text{if } \mathcal{M}(r) \subsetneq \mathcal{M}(u) \supsetneq \mathcal{M}(l), \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Here $d(s, s')$ is the number of internal nodes in the path in $ST|_{L(gt)}$ from s to s'.

The number of gene losses (under the restriction-based analysis) is given by $loss(gt, ST) = \sum_{g \in V(gt)} loss_g$, while for a set $\mathcal{G}$ of rooted, binary gene trees, the number of losses is given by $loss(\mathcal{G}, ST) = \sum_{gt \in \mathcal{G}} loss(gt, ST)$. The number of duplications and losses (again, under the restriction-based analysis), denoted by $Duploss(\mathcal{G}, ST)$, is the sum of the number of duplication and losses, i.e., $Duploss(\mathcal{G}, ST) = Dup(\mathcal{G}, ST) + loss(\mathcal{G}, ST)$.

2.2. *New Data Structures*

Subtree-Bipartitions: Let T be a rooted binary tree and u an internal node in T. The *subtree-bipartition* of u, denoted by $\mathcal{SBP}_T(u)$, is the unordered pair $(c_T(l)|c_T(r))$, where l and r are the two children of u. Note that subtree-bipartitions are not defined for leaf nodes. The set of subtree-bipartitions of a tree T is denoted by $\mathcal{SBP}_T = \{\mathcal{SBP}_T(u) : u \in V_{int}(T)\}$.

Domination, containment, disjointness, and compatibility: Let $BP_i = (P_{i_1}|P_{i_2})$ and $BP_j = (P_{j_1}|P_{j_2})$ be two subtree-bipartitions. We say that BP_i is *dominated* by BP_j (and

conversely that BP_j *dominates* BP_i) if either of the following two conditions holds: (1) $P_{i_1} \subseteq P_{j_1}$ and $P_{i_2} \subseteq P_{j_2}$, or (2) $P_{i_1} \subseteq P_{j_2}$ and $P_{i_2} \subseteq P_{j_1}$. We say that BP_i *contains* BP_j if $P_{j_1} \cup P_{j_2} \subseteq P_{i_1}$ or $P_{j_1} \cup P_{j_2} \subseteq P_{i_2}$, and that BP_i and BP_j are *disjoint* if $[P_{i_1} \cup P_{i_2}] \cap [P_{j_1} \cup P_{j_2}] = \emptyset$. We say that two subtree bipartitions are *compatible* if one contains the other, or they are disjoint.

The Compatibility Graph $CG(\mathcal{G})$***:*** Let $\mathcal{G}$ be a set of rooted binary gene trees on the set $\mathcal{X}$ of n taxa. The *compatibility graph* $CG(\mathcal{G})$ has one vertex for each possible subtree-bipartition defined on $\mathcal{X}$, and there is an edge between two vertices if and only if the associated subtree-bipartitions are compatible.

Note that if two subtree-bipartitions are compatible, then their associated clusters (produced by unioning the two parts of the bipartition) are also either disjoint or one contains the other.

Observation 2.1. A set $\mathcal{C}$ of $n-1$ subtree bipartitions is compatible (meaning all pairs of clusters are compatible) if and only if there exists a binary rooted tree whose set of subtree bipartitions is exactly $\mathcal{C}$.

Proof. Follows from the definition of subtree bipartition compatibility, and the fact that a set of $n-1$ compatible clusters on n taxa defines a binary tree with that set of clusters. □

We use the fact that $(n-1)$-cliques in the compatibility graph define rooted binary trees to develop solutions for the MGD_r and $MGDL_r$ problems. To do this, we define weights on nodes in the compatibility graph to characterize the solutions to these problems as $(n-1)$-cliques with maximum weight (for MGD_r or minimum weight (for $MGDL_r$). As was done by Than and Nakhleh[17] for the MDC_c problem, we will present a dynamic programming algorithm that finds an optimal $(n-1)$-clique in time that is polynomial in the number of nodes in the compatibility graph.

2.3. *Theorems*

All results here are for rooted binary gene trees and species trees. We assume that the species tree has exactly one copy of each taxon in $\mathcal{X}$, but that the gene trees can have any number (including zero) of each taxon in $\mathcal{X}$. The total number of taxa in $\mathcal{X}$ is n.

Lemma 2.1. *Let gt be a rooted binary gene tree, ST a rooted binary species tree, and u an internal node of gt. Suppose the subtree-bipartition for u is dominated by the subtree-bipartition of v in ST. Then* $\mathcal{M}(u) = v$.

Proof. Since $\mathcal{SBP}_{gt}(u)$ is dominated by $\mathcal{SBP}_{ST}(v)$, it follows that $c_{gt}(u) \subseteq c_{ST}(v)$. Let $w = \mathcal{M}(u)$. Hence, $c_{ST}(v) \cap c_{ST}(w) \neq \emptyset$, and so v and w are comparable (that is, either they are identical or one lies above the other in ST). Suppose by way of contradiction that $v \neq w$. Since $c_{gt}(u) \subseteq c_{ST}(v)$, it follows that v must lie above w. But then $c_{ST}(w)$ is a subset of the cluster of one of v's children, and so disjoint from the cluster for the other child. Hence, $\mathcal{SBP}_{gt}(u)$ is not dominated by $\mathcal{SBP}_{ST}(v)$, contradicting the initial assumption. □

The following corollary is then obvious:

Corollary 2.1. *Let gt be a rooted binary gene tree and ST a rooted binary species tree. Then every subtree-bipartition of gt is dominated by at most one subtree-bipartition in ST.*

Theorem 2.1. *Let ST be a rooted, binary species tree, gt be a rooted binary gene tree, and u an internal node in gt. Then the subtree-bipartition of u in gt is dominated by a subtree-bipartition in ST if and only if u is a speciation node.*

Proof. Suppose u is a node in gt such that its subtree-bipartition is dominated by a subtree bipartition in ST. Let l and r be the two children of u in gt. Then $\mathcal{SBP}_{gt}(u) = (c(l)|c(r))$. Let v be a node in ST such that $\mathcal{SBP}_{gt}(u)$ is dominated by $\mathcal{SBP}_{ST}(v)$. Let l' and r' be the children of v. Then, without loss of generality, $c(l) \subseteq c(l')$ and $c(r) \subseteq c(r')$. Therefore, under the MRCA mapping, l and r will be mapped to a node in the subtree rooted at l' and r', respectively. Moreover, by Lemma 2.1 $\mathcal{M}(u) = v$. Therefore, $\mathcal{M}(l) \neq \mathcal{M}(u)$, and $\mathcal{M}(r) \neq \mathcal{M}(u)$. Hence u is not a duplication node.

Next, assume that $\mathcal{SBP}_{gt}(u)$ is not dominated by any subtree-bipartition of ST, and let $\mathcal{SBP}_{ST}(\mathcal{M}(u)) = (p_1|p_2)$. Then at least one of the following holds (1) $c(l) \not\subset p_1$ and $c(l) \not\subset p_2$ or (2) $c(r) \not\subset p_1$ and $c(r) \not\subset p_2$. Without loss of generality, suppose (1) holds. Then l cannot map to a node strictly below v. However, it is also equally obvious that l cannot map to a node strictly above v, since $\mathcal{M}(u) = v$ and l is a child of u. Hence, it must be that $\mathcal{M}(l) = u$. But in this case, u is a duplication node. □

We now define some functions:

- $dominated(bp, ST) \in \{0, 1\}$, with $dominated(bp, ST) = 1$ if bp is dominated by a subtree-bipartition in $\mathcal{SBP}_{ST}$, and 0 otherwise.
- $dom(bp, bp') = 1$ if bp is dominated by bp' and 0 otherwise.

Corollary 2.2. *Let gt be a rooted binary gene tree and ST a rooted binary species tree. Then*

$$Dup(gt, ST) = |V_{int}(gt)| - \sum_{u \in V_{int}(gt)} dominated(\mathcal{SBP}_{gt}(u), ST).$$

Proof. Follows directly from Theorem 2.1. □

3. Algorithms for MGD_r on rooted binary gene trees

3.1. *Graph-theoretic characterization of optimal solution to MGD_r*

Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of rooted, binary gene trees on the set $\mathcal{X}$ of n taxa, and let n_i be the number of leaves in tree gt_i. Note that n_i does not refer to $|L(gt_i)|$, since $L(gt_i)$ is the set of taxa in $\mathcal{X}$ that appear at least once in gt_i, whereas n_i is the total number of leaves in gt_i. Since gt_i can have multiple copies of a taxon, n_i can be larger than $|L(gt_i)|$.

We construct the *compatibility graph* $CG(\mathcal{G})$ with one vertex for each possible subtree-bipartition defined on $\mathcal{X}$, as described in the previous section. We set the weight of each node v, denoted by $W_{dom}(v)$, to be the total number of subtree-bipartitions of $\mathcal{G}$ that are dominated

by v. That is,

$$W_{dom}(v) = \sum_{gt \in \mathcal{G}} |\{bp : bp \in \mathcal{SBP}_{gt} \text{ and } dom(bp, v) = 1\}|.$$

We then find a clique $\mathcal{C}$ of size $n-1$ so as to maximize the weight $W_{dom}(\mathcal{C})$ of the clique $\mathcal{C}$, where $W_{dom}(\mathcal{C}) = \sum_{v \in \mathcal{C}} W_{dom}(v)$.

Theorem 3.1. *Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of binary, rooted gene trees on the n taxa in $\mathcal{X}$. Let $\mathcal{C}$ be an $(n-1)$-clique in $CG(\mathcal{G})$ maximizing $W_{dom}(\mathcal{C})$, and let ST be the species tree defined by the clique (so that $\mathcal{SBP}_{ST}$ corresponds to $\mathcal{C}$). Then ST is a binary species tree that optimizes MGD_r with respect to $\mathcal{G}$.*

Proof. Recall that any $(n-1)$-clique in the compatibility graph defines a rooted binary tree on $\mathcal{X}$. Let $\mathcal{C}$ be a clique of size $n-1$ and ST be the tree defined by $\mathcal{C}$. By Corollary 2.1, every subtree-bipartition in gt_i can be dominated by at most one node in $\mathcal{C}$. Therefore, each node of gt_i contributes either 1 (if the node is dominated) or 0 (if the node is not dominated) to the weight of $\mathcal{C}$. Let w_i be the amount contributed by gt_i to the weight of $\mathcal{C}$. Thus, w_i is the number of speciation nodes in gt_i with respect to the species tree corresponding to ST. Then

$$\sum_{v \in \mathcal{C}} W_{dom}(v) = \sum_{i=1}^{k} w_i = W_{dom}(\mathcal{C}).$$

Furthermore, by Corollary 2.2 and because a rooted binary tree with n_i leaves has $n_i - 1$ internal nodes, $Dup(gt_i, ST) = n_i - 1 - w_i$. Then,

$$Dup(\mathcal{G}, T) = \sum_{i=1}^{k} Dup(gt_i, ST) = \sum_{i=1}^{k} [n_i - 1 - w_i] = N - k - W_{dom}(\mathcal{C}),$$

where $\sum_{i=1}^{k} n_i = N$. Therefore, the clique with maximum weight defines a tree ST that minimizes $Dup(\mathcal{G}, ST)$.

□

3.2. *The Dynamic Programming Algorithm for MGD_r*

The graph-theoretic characterization of the optimal solution for MGD_r given in the previous section suggests an algorithm for finding the optimal solution, in which a max weight clique is sought in an exponentially large graph. However, we will show that this optimal solution can be found in time that is polynomial in the number of vertices in the graph, using dynamic programming. In addition, we will show that a constrained version of the MGD_r problem, in which the allowed subtree-bipartitions are given as input, can also be solved using the same basic dynamic programming algorithm. Finally, when the set of allowed subtree-bipartitions comes from the input set of gene trees, the result is an algorithm that runs in polynomial time.

The motivation to restrict the attention to a subset of the subtree-bipartitions comes from the observations made by Than and Nakhleh,[17] who noted that that clusters in the species tree that optimizes MDC tend to appear in at least one of the input gene trees. Therefore,

we consider a constrained search problem, where instead of considering all possible subtree-bipartitions, we only consider the subtree-bipartitions of the gene trees. When we do this, instead of constructing a compatibility graph with one node for each subtree bipartition, the compatibility graph will only have nodes for the (at most) $N - k$ subtree bipartitions in the input gene trees (where $N = \sum_{i=1}^{k} n_i$). A clique of size $n - 1$ with the maximum weight will define an optimal solution to the constrained version of MGD_r where the species tree is only permitted to have subtree bipartitions from the input gene trees.

Let $\mathcal{SBP}$ be any set of subtree-bipartitions, and let $\mathcal{CLS}$ be the set of associated clusters (i.e. $\mathcal{CLS} = \{p \cup q : (p|q) \in \mathcal{SBP}\}$. We will define the constrained MGD_r problem by limiting the solution space to those rooted, binary trees, all of whose subtree-bipartitions are in the set $\mathcal{SBP}$. Thus, by setting $\mathcal{SBP}$ to be the set of all possible subtree-bipartitions we obtain the globally optimal solution, but setting $\mathcal{SBP}$ to be a proper subset of the set of all subtree-bipartitions is also possible.

By Theorem 3.1, the binary species tree with a maximum total weight (as defined by summing up the weights of its subtree bipartitions) has a minimum number of duplications, because the duplication nodes are exactly those nodes whose subtree-bipartitions are not dominated by any subtree-bipartition in the species tree.

We now show how to calculate that optimal binary species tree directly, using dynamic programming. The DP algorithm computes a rooted, binary tree T_A for every cluster $A \in \mathcal{CLS}$, such that T_A maximizes the sum, over all gene trees t, of the number of subtree-bipartitions in t that are dominated by some subtree-bipartition in T_A. We denote this total number by $value(A)$.

We preprocess the data as follows. First, we compute the set $\mathcal{CLS}$, and order its elements based on size. We also calculate $\mathcal{SBP}_{\mathcal{G}} = \bigcup_{i=1}^{k} \mathcal{SBP}_{gt_i}$, i.e. the set of all subtree bipartitions in all gene trees, and we set $count(x)$ for $x \in \mathcal{SBP}_{\mathcal{G}}$ to be the number of times x appears in any of the gene trees. Recall that for a subtree bipartition x, we define $W_{dom}(x)$ to be the number of subtree bipartitions of the gene trees that are dominated by x. We define a partial order for elements of $\mathcal{SBP}$ and $\mathcal{SBP}_{\mathcal{G}}$ based upon subtree-bipartition size. For every ordered pair $< x, y >$ such that $x \in \mathcal{SBP}_{\mathcal{G}}$ and $y \in \mathcal{SBP}$, we determine whether x is dominated by y; if y dominates x then $W_{dom}(y)$ is incremented by $count(x)$. At the end of this step, $W_{dom}(y)$ is calculated correctly for every $y \in \mathcal{SBP}$. All this preprocessing can be computed in $O(n|\mathcal{SBP}|^2)$.

We compute $value(A)$ in order, from the smallest cluster to the largest cluster $\mathcal{X}$. We set $value(A)$ as follows. For any cluster A with two taxa, we set $value(A) = W_{dom}(a_1|a_2)$, where $A = \{a_1, a_2\}$. For a cluster A with more than two taxa, we set $value(A)$ as follows:

$$value(A) = \max\{value(A_1) + value(A - A_1) + W_{dom}(A_1|A - A_1) : (A_1|A - A_1) \in \mathcal{SBP}\}$$

If there is no $(A_1|A-A_1) \in \mathcal{SBP}$, we set its $value(A)$ to $-\infty$, signifying that A cannot be further resolved. At the end of the algorithm, if $\mathcal{SBP}$ includes at least one clique of size $n-1$, we have computed $value(\mathcal{X})$ as well as sufficient information to construct the species tree having the minimum number of duplications. If subtree bipartitions in $\mathcal{SBP}$ are not sufficient for building a fully resolved tree on $\mathcal{X}$, then $value(\mathcal{X})$ will be $-\infty$, and our algorithm returns FAIL. Note that for a specific cluster A, $value(A)$ can be computed in $O(|\mathcal{SBP}|)$ time, since at worst we

need to look at every subtree-bipartition in $\mathcal{SBP}$. In other words, we have proven the following:

Theorem 3.2. *Let $\mathcal{G}$ be a set of rooted binary gene trees, $\mathcal{SBP}$ a set of subtree-bipartitions. Then, if subtree bipartitions of $\mathcal{SBP}$ define at least one binary tree on $\mathcal{X}$, then the DP algorithm finds the species tree ST minimizing the total number of duplications subject to the constraint that $\mathcal{SBP}_{ST} \subseteq \mathcal{SBP}$ in $O(n|\mathcal{SBP}|^2)$ time. Therefore, if $\mathcal{SBP}$ is all possible subtree-bipartitions, we have an exact but exponential time algorithm. However, if $\mathcal{SBP}$ contains only those subtree-bipartitions from the input gene trees, then the DP algorithm finds the optimal constrained species tree in $O(d^2n^3k^2)$ (since the number of subtree-bipartitions $|\mathcal{SBP}|$ in $\mathcal{G}$ is $O(dkn)$), where n is the number of species, k is the number of gene trees, and d the maximum number of times that any taxon appears in any gene tree.*

4. Algorithms for $MGDL_r$

4.1. *Graph-Theoretic Characterization*

We begin with some additional terminology and theorems. For any cluster A in gt and a cluster B in ST, we say that A is B-maximal if (1) $A \subseteq B$, and (2) for any cluster A' in gt, if $A \subseteq A'$, then $A' \not\subseteq B$. We define $k_B(gt)$ to be the number of B-maximal clusters within gt, and Finally, in a rooted tree T with cluster G, the unique edge e that separates G from the rest of the leaves in T is called the *parent edge* of the cluster G.

Theorem 4.1. *(From Than and Nakhleh[17] and Yu, Warnow, and Nakhleh[18]) Let gt be a rooted binary gene tree and ST a species tree on the same set of taxa. Let B be a cluster in ST and let e be the parent edge of B in ST. Then $k_B(gt)$ is equal to the number of lineages on e in an optimal reconciliation of gt within ST with respect to MDC_c. Therefore, $MDC_c(gt, ST) = \sum(k_B(gt) - 1)$, where B ranges over the clusters of ST.*

Theorem 4.2. *Let gt be a rooted binary gene tree and ST a species tree on the same set of leaves. Then $MDC_r(gt, ST) = \sum(k_B(gt) - 1)$, where B ranges over the clusters of $ST|_{L(gt)}$.*

Proof. By definition, $MDC_r(gt, ST) = MDC_c(gt, ST|_{L(gt)})$. However, gt and $ST|_{L(gt)}$ have the same set of taxa. Therefore, by Theorem 4.1, $MDC_c(gt, ST|_{L(gt)}) = \sum(k_B(gt) - 1)$, as B ranges over the clusters of $ST|_{L(gt)}$. □

Theorem 4.3. *(From Zhang[21]) Let gt be a rooted binary gene tree and ST a rooted binary species tree. Then, under the restriction-based analysis, $Duploss(gt, ST) = MDC_r(gt, ST) + 3 * Dup(gt, ST) + |V(gt)| - |V(\mathcal{R}_{ST}(L(gt)))|$.*

Let v be a vertex associated with the subtree-bipartition $(p|q)$, and let $B = p \cup q$ be the cluster associated with v. Define $W_{xl}(v, gt)$ to be 0 if $p \cap L(gt) = \emptyset$ or $q \cap L(gt) = \emptyset$, and otherwise to be $k_B(gt) - 1$. Set $W_{xl}(v) = \sum_{i=1}^{k} W_{xl}(v, gt_i)$. Then, for any species tree ST and set $\mathcal{G}$ of gene trees, $MDC_r(\mathcal{G}, ST) = \sum_{i=1}^{k} MDC_r(gt_i, ST) = \sum_{v \in \mathcal{C}} W_{xl}(v)$, where $\mathcal{C}$ is the clique in $CG(\mathcal{G})$ that corresponds to ST.

Theorem 4.4. *Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ be a set of binary rooted gene trees on set $\mathcal{X}$ of n species, and let $CG(\mathcal{G})$ be the compatibility graph with vertex weights defined by $W_{MGDL}(v) =$*

$W_{xl}(v) - 3W_{dom}(v)$. *The set of bipartitions in an* $(n-1)$*-clique of minimum weight in* $CG(\mathcal{G})$ *defines a binary species tree* ST *that optimizes* $MGDL_r$.

Proof. Let $\mathcal{C}$ be a clique of size $n-1$ and ST be the rooted binary tree defined by the subtree-bipartitions represented by the nodes in $\mathcal{C}$. Let $\mathcal{SBP}_{dom}(gt, ST)$ be the set of subtree-bipartitions in gt that are dominated by a subtree-bipartition in ST, i.e., $\mathcal{SBP}_{dom}(gt, ST) = \{bp : bp \in \mathcal{SBP}_{gt} \text{ and } dominated(bp, ST) = 1\}$. Note that $|\mathcal{SBP}_{dom}(gt, ST)|$ is the number of speciation nodes in gt with respect to ST. Therefore, the total number of speciation nodes in $\mathcal{G}$ is $\sum_{i=1}^{k} |\mathcal{SBP}_{dom}(gt_i, ST)| = \sum_{v \in V_{int}(ST)} W_{dom}(v)$. Let $N = \sum_{i=1}^{k} n_i$. Then,

$$
\begin{aligned}
Duploss(\mathcal{G}, ST) &= \sum_{i=1}^{k} Duploss(gt_i, ST) \\
&= \sum_{i=1}^{k} [MDC_r(gt_i, ST) + 3 * Dup(gt_i, ST) - (|V(gt_i)| - |V(\mathcal{R}_{ST}(L(gt_i)))|)] \text{ (by Theorem 4.3)} \\
&= \sum_{i=1}^{k} [MDC_r(gt_i, ST) + 3 * Dup(gt_i, ST)] - \sum_{i=1}^{k} (|V(gt_i)| - |V(\mathcal{R}_{ST}(L(gt_i)))|) \\
&= \sum_{i=1}^{k} [MDC_r(gt_i, ST) + 3 * ((n_i - 1) - |\mathcal{SBP}_{dom}(gt_i, ST)|)] \\
&\quad - \sum_{i=1}^{k} (|V(gt_i)| - |V(\mathcal{R}_{ST}(L(gt_i)))|) \text{ (by Corollary 2.2)} \\
&= \sum_{v \in \mathcal{C}} W_{xl}(v) + \sum_{i=1}^{k} 3(n_i - 1) - 3 \sum_{v \in \mathcal{C}} W_{dom}(v) \\
&\quad - \sum_{i=1}^{k} (2n_i - 1) + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i)))| \text{ (since } |V(gt_i)| = 2n_i - 1) \\
&= \sum_{v \in \mathcal{C}} (W_{xl}(v) - 3W_{dom}(v)) + 3 \sum_{i=1}^{k} n_i \ - 3k - 2 \sum_{i=1}^{k} n_i + k + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i)))| \\
&= \sum_{v \in \mathcal{C}} W_{MGDL}(v) + \sum_{i=1}^{k} n_i - 2k + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i)))| \\
&= W_{MGDL}(\mathcal{C}) + N - 2k + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i)))|
\end{aligned}
$$

Note that $|V(\mathcal{R}_{ST}(L(gt_i)))|$ does not depend on ST. Therefore, the clique $\mathcal{C}$ with minimum weight defines a tree ST that minimizes $Duploss(\mathcal{G}, ST)$.

□

4.2. *Dynamic Programming Approach for* $MGDL_r$

We now show how to use dynamic programming to find the optimal solution for $MGDL_r$ without having to explicitly search for the optimal clique. As we did for MGD_r, we generalize

the problem to allow the user to provide a set $\mathcal{SBP}$ of subtree-bipartitions, and the solution space is restricted to those rooted, binary trees, all of whose subtree-bipartitions are in the set $\mathcal{SBP}$.

We compute $value(A)$ for all clusters A with at least two species as follows. If $|A| = 2$, we set $value(A) = W(a_1|a_2)$, where $A = \{a_1, a_2\}$. For set A with more than two taxa, we set $value(A)$ as follows:

$$value(A) = \min\{value(A_1) + value(A - A_1) + W_{xl}(A_1|A - A_1) - 3W_{dom}(A_1|A - A_1) : (A_1|A - A_1) \in \mathcal{SBP}\}.$$

The optimal number of duplications and losses is given by $value(\mathcal{X}) + N - 2k + \sum_{i=1}^{k} |V(\mathcal{R}_{ST}(L(gt_i)))|$, where $N = \sum_{i=1}^{k} n_i$, and n_i is the number of leaves in gene tree gt_i. By backtracking, we can find the optimal set of compatible clusters and hence can construct the optimal tree. We now have the following theorem:

Theorem 4.5. *Let $\mathcal{G}$ be a set of k rooted binary gene trees on the set $\mathcal{X}$ of n taxa. Let $\mathcal{SBP}$ be an arbitrary set of subtree bipartitions on $\mathcal{X}$. Then the DP algorithm finds the species tree ST optimizing $MGDL_r$, subject to the constraint that $\mathcal{SBP}_{ST} \subseteq \mathcal{SBP}$, in $O(n|\mathcal{SBP}|^2)$ time. Therefore, for the case where $\mathcal{SBP}$ is the set of subtree-bipartitions from the k gene trees, the algorithm uses $O(d^2n^3k^2)$ time, where d is the maximum number of times any taxon appears in any gene tree.*

5. Acknowledgments

This research was supported by NSF (DEB 0733029 and DBI 1062335) to TW, NSERC (to SM), the Guggenheim Foundation (to TW), and the Fulbright Foundation (to MSB).

References

1. M. T. Hallett and J. Lagergren, New algorithms for the duplication-loss model, in *Proc RECOMB*, 2000.
2. W. Fitch and E. Margoliash, *Science* **155**, 279 (1967).
3. R. D. M. Page, *Syst. Biol.* **43**, 58 (1994).
4. M. Goodman, J. Czelusniak, G. Moore, E. Romero-Herrera and G. Matsuda, *Syst. Zool.* **28**, 132 (1979).
5. W. P. Maddison, *Syst. Biol.* **46**, 523 (1997).
6. L. Zhang, *J. Comp. Biol.* **4**, 177 (1997).
7. R. Chaudhary, M. S. Bansal, A. Wehe, D. Fernández-Baca and O. Eulenstein, *BMC Bioinf.* , 574 (2010).
8. A. Wehe, M. S. Bansal, J. G. Burleigh and O. Eulenstein, *Am. J. Bot.* **24**, 1540 (2008).
9. J. B. Slowinski, A. Knight and A. P. Rooney, *Mol. Phylog. Evol.* **8**, 349 (1997).
10. R. D. M. Page, *Mol. Phylog. Evol.* **14**, 89 (2000).
11. R. D. M. Page and J. A. Cotton, Vertebrate phylogenomics: reconciled trees and gene duplications, in *Proc Pacific Symposium on Biocomputing*, 2002.
12. J. Cotton and R. Page, *Tangled tales from multiple markers: reconciling conflict between phylogenies to build molecular supertrees*, in *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, ed. O. R. P. Bininda-Emonds 2004, pp. 107–125.
13. M. Sanderson and M. McMahon, *BMC Evol. Biol.* **7**, p. S3 (2007).

14. J. P. Doyon and C. Chauve, *Software Tools and Algorithms for Biological Systems (book series, Advances in Experimental Medicine and Biology)* , 287 (2011).
15. W.-C. Chang, G. J. Burleigh, D. F. Fernndez-Baca and O. Eulenstein, *BMC Bioinf.* **12**, p. S14 (2011).
16. U. Stege, Gene trees and species trees: The gene-duplication problem is fixed-parameter tractable, in *Proc International Workshop on Algorithms and Data Structures (WADS)*, 1999.
17. C. V. Than and L. Nakhleh, *PLoS Comp. Biol.* **5** (2009).
18. Y. Yu, T. Warnow and L. Nakhleh, Algorithms for MDC-based multi-locus phylogeny inference: Beyond rooted binary gene trees on single alleles, in *Proc RECOMB*, 2011.
19. J. Yang and T. Warnow, *BMC Bioinf.* **12(Suppl 9)** (2011).
20. M. S. Bayzid and T. Warnow, *J. Comp. Biol.* **19**, 591 (2012).
21. L. Zhang, *IEEE/ACM Trans. Comp. Biol. Bioinf.* **8**, 1685 (2011).
22. R. Guigo, I. Muchnik and T. Smith, *Mol. Phylog. Evol.* **6**, 189 (1996).
23. B. Ma, M. Li and L. Zhang, On reconstructing species trees from gene trees in terms of duplications and losses, in *Proc RECOMB*, 1998.
24. P. Gorecki, Reconciliation problems for duplication, loss and horizontal gene transfer, in *Proc RECOMB*, 2004.