July 25, 2012

Dear Sir/Madam,

Enclosed is a paper entitled "Inferring Optimal Species Trees under Gene Duplication and Loss" for submission to Pacific Symposium on Biocomputing, 2013. Below are our responses to your submission requirements.

- The email address of the corresponding author:
  tandy@cs.utexas.edu
- The specific PSB session that should review the paper:
  Phylogenomics and Population Genomics: Models, Algorithms, and Analytical Tools
- We give our assurance that this paper is our original unpublished work and it is not under consideration elsewhere. Also, we will not submit it elsewhere until we have received your decision.
- Finally, all co-authors concur with the contents of the paper.

Regards,

Tandy Warnow, Professor
Department of Computer Science
The Unviersity of Texas at Austin

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, 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.

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 each taxon appear in each gene sequence alignment only once, and that the true trees for the different genes be 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,[1–5] 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[6] and DupTree,[7] 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,[8] vertebrates,[9,10] *Drosophia*,[11] and plants.[12] 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 this paper we will present a new approach for MGD and MGDL that does not use local search 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.

The algorithmic technique we present is related to the approach used in Than and Nakhleh[13] (see also Yun, Warnow, and Nakhleh[14]) for the MDC (Minimize Deep Coalescence) problem,[4] 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. Each possible cluster (subset of the species set) is represented by a node in a graph, and edges exist between pairs of nodes whose clusters are compatible – meaning they are either pairwise disjoint or one contains the other. 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. Than and Nakhleh[13] 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 clique of size $n-1$.

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 min 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. 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,[13–15] 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 time that is polynomial in the number of vertices in the graph. 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. Finally, we discuss future directions in Section 5.

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. In the case of the MGD problem, $d(t_i, T) = Dup(t_i, T)$, which is the minimum number of duplications needed to explain the topological differences between T and t_i. For the MGDL problem, $d(t_i, T) = Duploss(t_i, T)$ is the minimum number of duplications and losses needed to explain the topological differences between T and t_i. Finally, the MDC problem, introduced by Maddison,[4] seeks the species tree that minimizes $MDC(t_i, T)$, the number of "extra lineages" in T implied by tree t_i. (Due to space constraints, we direct the reader to prior publications[4,13,14] for more on MDC.)

Each of these three costs ($Dup(t, T), Duploss(t, T)$ and $MDC(t, T)$) depends upon finding an "optimal embedding" of the gene tree t into the species tree T; this is also called an optimal "reconciliation". An embedding of a rooted gene tree t into a species tree T is defined by a mapping f from the nodes of the gene tree to the nodes of the species tree that has some natural properties: first, it maps leaves in the gene tree mapped to the unique leaf in the species tree with the same taxon label, and second it 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 and leaves of a tree T by $V(T)$ and $L(T)$, respectively, and the internal nodes by $V_{int}(T)$. (Since the gene trees can have more than a single copy of the taxa, we let $L(T)$ denote the set of taxa that appear at least once in T.) The root of T is denoted by $root(T)$. 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 for the clade rooted at v by $cluster_T(v)$. Finally, in a rooted tree T with clade B, the edge e that is incident with the root of B but whose other endpoint is not within the subtree below the root is called the *parent edge* of the cluster B.

Optimal reconciliation of gene trees and species trees: the MRCA mapping. 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 $\mathcal{M} : V(gt) \to V(ST)$ by $\mathcal{M}(v) = MRCA_{ST}(cluster_{gt}(v)))$. In other words, $\mathcal{M}$ associates each node u of gt to the MRCA in ST of the cluster below u. It is now known that the optimal embedding for each of the three criteria we discuss (MDC, MGD, or MGDL) 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.[4,5,13,16] 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, MGDL, and MDC 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*.[16–19]

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 an 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 on $L(gt)$, denoted by $\mathcal{R}_{ST}(L(gt))$ is the smallest subtree of ST containing $L(gt)$ as its leaf set. The homomorphic subtree $ST|_{L(gt)}$ of ST induced by $L(gt)$ is a tree obtained from $\mathcal{R}_{ST}(L(gt))$ by contracting all nodes with degree 2 except for the root of $\mathcal{R}_{ST}(L(gt))$. We first consider the MRCA mapping $\mathcal{M}$ from gt to $ST|_{L(gt)}$. Then, the number of gene losses for a given gene tree gt and species tree ST for a particular internal node g, denoted by $loss_g$, can be calculated as follows:[16–19]

$$loss_g = \begin{cases} d(\mathcal{M}(a), \mathcal{M}(g)) + 1 & \text{if } \mathcal{M}(a) \subsetneq \mathcal{M}(g) = \mathcal{M}(b), \\ d(\mathcal{M}(a), \mathcal{M}(g)) + d(\mathcal{M}(b), \mathcal{M}(g)) & \text{if } \mathcal{M}(a) \subsetneq \mathcal{M}(g) \supsetneq \mathcal{M}(b), \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Here $d(s, s')$ is the number of nodes on the path from s to s' excluding s and s'.

The number of gene losses 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, 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 pair $(cluster_T(l)|cluster_T(r))$, 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 $\mathcal{SBP}_T = \{\mathcal{SBP}_T(u) : u \in V_{int}(T)\}$.

Subtree-bipartition 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. Furthermore, a set $\mathcal{C}$ of $n-1$ clusters is pairwise compatible if and only if there exists a binary rooted tree whose set of clusters is exactly $\mathcal{C}$ (this is a well known and easily proven result in phylogenetics).

We use the fact that $(n-1)$-cliques in the compatibility graph define rooted binary trees to develop solutions for the MGD and MGDL 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) or minimum weight (for MGDL). As was done by Than and Nakhleh,[13] 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. Finally, will show that a restriction of the compatibility graph to a subset of the possible subtree-bipartitions allows us to obtain more efficient solutions, some of which are polynomial in the number of species and gene trees.

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 $cluster_{gt}(u) \subseteq cluster_{ST}(v)$. Let $w = \mathcal{M}(u)$. Hence, $cluster_{ST}(v) \cap cluster_{ST}(w) \neq \emptyset$, 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 $cluster_{gt}(u) \subseteq cluster_{ST}(v)$, it follows that v must lie above w (i.e. between w and the root of ST). But then $cluster_{ST}(w)$ 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 assumption that $v \neq w$. □

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. Let l and r be the two children of u in gt. Then $\mathcal{SBP}_{gt}(u) = (cluster(l)|cluster(r))$. 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, $cluster(l) \subseteq cluster(l')$ and $cluster(r) \subseteq cluster(r')$. 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 $\mathcal{SBP}_{ST}(\mathcal{M}(u)) = (p_1|p_2)$ (i.e., the subtree bipartition at v is (p_1, p_2)). Then at least one of the following holds (1) $cluster(l) \not\subset p_1$ and $cluster(l) \not\subset p_2$ or (2) $cluster(r) \not\subset p_1$ and $cluster(r) \not\subset p_2$.

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. □

Note therefore that this implies that for a given species tree ST and gene tree gt, the vertices of gt partition into duplication nodes, speciation nodes, and leaves, and that the partition is determined by the domination relation.

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 on rooted binary gene trees

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

Let $\mathcal{G}$ be a set of rooted, binary gene trees on the set $\mathcal{X}$ of n taxa. 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 $\mathcal{G}$ 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 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 $\mathcal{C}$. Let $gt_i \in \mathcal{G}$ be a gene tree. 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 or 0 to the weight of $\mathcal{C}$. Let w_i be the amount contributed by gt_i to the weight of $\mathcal{C}$ and let n_i be the number of leaves in gt_i. Note that n_i does not refer to $|L(gt_i)|$. $L(gt_i)$ is the set of taxa in gt_i that appear at least once in gt, whereas, n_i is the total number of leaves in gt_i (considering multiple copies of a taxon). Then

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

Furthermore, by Corollary 2.2 and because a rooted binary tree with n_i leaves has $n_i - 1$ internal nodes, the number $Dup(gt, ST)$ of duplication nodes in gt_i with respect to ST is $n_i - 1 - w_i$. Then,

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

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

3.2. *The Dynamic Programming Algorithm for MGD*

The graph-theoretic characterization of the optimal solution for MGD 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 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,[13] 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 where the species tree is only permitted to have subtree bipartitions from the input gene trees. Here we show how to find this optimal solution using dynamic programming.

Let the set $\mathcal{SBP}$ of subtree-bipartitions be provided. We will define the constrained MGD problem by limiting the 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. Here we 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 of at least two elements that appears in some gene tree, 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 cluster $cl(x)$ (where $cl(x) = p \cup q$ for $x = (p|q)$) to every subtree-bipartition $x \in \mathcal{SBP}$. We partially order the set $cl(\mathcal{SBP}) = \{cl(x) : x \in \mathcal{SBP}\}$, based upon containment. For every ordered pair $< x, y >$ of subtree-bipartitions in $\mathcal{SBP}$, we determine whether x is dominated by y; this defines a partially ordered set, since if x is dominated by y and y is dominated by z, then x is dominated by z. All this preprocessing can be computed in $O(n|\mathcal{SBP}|^2)$.

We compute $value(A)$ in order, from the smallest sets to the largest set $\mathcal{X}$. Recall that for a subtree bipartition $(A_1|A_2)$, we define $W_{dom}(A_1|A_2)$ to be the number of subtree bipartitions of the gene trees that are dominated by $(A_1|A_2)$. We set $value(A)$ as follows. For sets A with two taxa, we set $value(A) = W_{dom}(a_1|a_2)$, where $A = \{a_1, a_2\}$. For sets A with more than two taxa, we set $value(A)$ as follows:

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

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 $\mathcal{SBP}$. At the end of the algorithm, we have computed $value(\mathcal{X})$ as well as sufficient information to construct the species tree having the minimum number of duplications. 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 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 $O(d^2n^3k^2)$, 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

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 $A' \not\subseteq B$. Finally, $k_B(gt)$ is the number of B-maximal clusters within gt.

Theorem 4.1. *(From Yun, Warnow, and Nakhleh[14]) Let gt be a rooted binary gene tree and ST a species tree on the same set of leaves. 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. Therefore, $MDC(gt, ST) = \sum(k_B(gt) - 1)$, where B ranges over the clusters of ST.*

Theorem 4.2. *(From Zhang[16]) Let gt be a rooted binary gene tree and ST a rooted binary species tree. Then $Duploss(gt, ST) = MDC(gt, ST)+3*Dup(gt, ST)+(|V(gt)|-|V(\mathcal{R}_{ST}(L(gt)))|)$.*

Let $\mathcal{G}$ be a set of rooted, binary gene trees on the set $\mathcal{X}$ of taxa. We construct the compatibility graph $CG(\mathcal{G})$. Let v be a vertex associated with the subtree-bipartition $(p|q)$, and recall that $W_{dom}(v)$ is the number of subtree-bipartitions dominated by $(p|q)$. We let $W_{xl}(v)$ be number of extra lineages for the subtree-bipartition $(p|q)$, which (by theorems established in Than and Nakhleh[13] and Yu, Warnow, and Nakhleh[14]) satisfies $W_{xl}(v) = \sum_{gt \in \mathcal{G}}(k_B(gt) - 1)$, where B is the cluster associated with the subtree-bipartition $(p|q)$, i.e., $B = p \cup q$. We weight the node v using $W_{MGDL}(v) = W_{xl}(v) - 3 * W_{dom}(v)$. We then seek a clique $\mathcal{C}$ of size $|\mathcal{X}| - 1$ so as to minimize $W_{MGDL}(\mathcal{C}) = \sum_{v \in \mathcal{C}} W_{MGDL}(v)$.

Theorem 4.3. *Let $\mathcal{G}$ be a set of binary rooted gene trees, 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.*

Proof. Recall that the subtree-bipartitions corresponding to a clique of size $n - 1$ (where $n = |\mathcal{X}|$) in $CG(\mathcal{G})$ defines a rooted, binary species tree. Let $\mathcal{C}$ be a clique of size $n - 1$ and ST be the tree defined by the subtree-bipartitions represented by the nodes in $\mathcal{C}$. Let $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$. 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., $\mathcal{SBP}_{dom}(gt, ST) = \{bp : bp \in \mathcal{SBP}_{gt}$ and $dominated(bp, ST) = 1\}$. Note that $|\mathcal{SBP}_{dom}(gt, T)|$, which is equal to $\sum_{v \in V_{int}(ST)} W_{dom}(v)$, is the number of speciation nodes in gt with respect to ST. Then

$$
\begin{aligned}
Duploss(\mathcal{G},T) &= \sum_{i=1}^{|\mathcal{G}|} Duploss(gt_i,T) \\
&= \sum_{i=1}^{|\mathcal{G}|}[XL(gt_i,T) + 3 * Dup(gt_i,T) - (|V(gt_i)| - |V(\mathcal{R}_{ST}(L(gt_i)))|)] \text{ (by Theorem. 4.2)} \\
&= \sum_{i=1}^{|\mathcal{G}|}[XL(gt_i,T) + 3 * Dup(gt_i,T)] - \sum_{i=1}^{|\mathcal{G}|}(|V(gt_i)| - |V(\mathcal{R}_{ST}(L(gt_i)))|) \\
&= \sum_{i=1}^{|\mathcal{G}|}[\sum_{B \in cluster_T}(k_B(gt_i)-1) + 3 * ((n_i-1) - |\mathcal{SBP}_{dom}(gt_i,T)|)] \\
&\quad - \sum_{i=1}^{|\mathcal{G}|}(|V(gt_i)| - |V(\mathcal{R}_{ST}(L(gt_i)))|) \text{ (by Theorem. 4.1, Corollary 2.2)} \\
&= \sum_{v \in \mathcal{C}} W_{xl}(v) - 3\sum_{v \in \mathcal{C}} W_{dom}(v) + \sum_{i=1}^{|\mathcal{G}|} 3(n_i-1) \\
&\quad - \sum_{i=1}^{|\mathcal{G}|}(2n_i-1) + \sum_{i=1}^{|\mathcal{G}|}|V(\mathcal{R}_{ST}(L(gt_i)))| \text{ (since } |V(gt_i)| = 2n_i - 1) \\
&= \sum_{v \in \mathcal{C}}(W_{xl}(v) - 3W_{dom}(v)) + 3\sum_{i=1}^{|\mathcal{G}|} n_i \ - 3k \\
&\quad -2\sum_{i=1}^{|\mathcal{G}|} n_i + k + \sum_{i=1}^{|\mathcal{G}|}|V(\mathcal{R}_{ST}(L(gt_i)))| \text{ (since } |\mathcal{G}| = k) \\
&= \sum_{v \in \mathcal{C}} W_{MGDL}(v) + \sum_{i=1}^{|\mathcal{G}|} n_i - 2k + \sum_{i=1}^{|\mathcal{G}|}|V(\mathcal{R}_{ST}(L(gt_i)))| \\
&= W_{MGDL}(\mathcal{C}) + N - 2k + \sum_{i=1}^{|\mathcal{G}|}|V(\mathcal{R}_{ST}(L(gt_i)))| \text{ (since } \sum_{i=1}^{|\mathcal{G}|} n_i = N)
\end{aligned}
$$

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*

We now show how to use dynamic programming to find the optimal solution for MGDL without having to explicitly search for the optimal clique. As we did for MGDL, 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 $\mathcal{SBP}$. 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. In particular, we can set $\mathcal{SBP} = \cup_i \mathcal{SBP}_{gt_i}$, where i ranges over $1, \ldots, k$ for a given set $\mathcal{G} = \{gt_1, gt_2, \ldots, gt_k\}$ of gene trees.

For a cluster A, we compute $value(A)$ as follows. For A with only two taxa, 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 duplication and loss is given by $value(\mathcal{X}) + 3N$, 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 4.4. *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 minimizing the total number of duplications and losses 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. Conclusions

We have presented theoretical results and polynomial time algorithms for exactly solving constrained versions of MGD and MGDL, analogous to the "heuristic" version of Phylonet-MDC. Because these algorithms do not use a local search technique they have the potential to provide better results than those obtained by iGTP and DupTree, which rely upon local search, for datasets with many taxa. Not addressed in this paper is the case where the gene trees are unrooted. We have extended our algorithm to the case where the gene trees are single copy and unrooted, but the case where gene trees are multiple copy and unrooted is more challenging. Our software is implemented and available upon request from the authors (and will be made available in open source form if accepted).

6. 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. W. Fitch and E. Margoliash, *Science* **155**, 279 (1967).
2. R. D. M. Page, *Syst. Biol.* **43**, 58 (1994).
3. M. Goodman, J. Czelusniak, G. Moore, E. Romero-Herrera and G. Matsuda, *Syst. Zool.* **28**, 132 (1979).
4. W. P. Maddison, *Syst. Biol.* **46**, 523 (1997).
5. L. Zhang, *J. Comp. Biol.* **4**, 177 (1997).
6. R. Chaudhary, M. S. Bansal, A. Wehe, D. Fernández-Baca and O. Eulenstein, *BMC Bioinf.* , 574 (2010).
7. A. Wehe, M. S. Bansal, J. G. Burleigh and O. Eulenstein, *Am. J. Bot.* **24**, 1540 (2008).
8. J. B. Slowinski, A. Knight and A. P. Rooney, *Mol. Phylog. Evol.* **8**, 349 (1997).

9. R. D. M. Page, *Mol. Phylog. Evol.* **14**, 89 (2000).
10. R. D. M. Page and J. A. Cotton, Vertebrate phylogenomics: reconciled trees and gene duplications, in *Proc Pacific Symposium on Biocomputing*, 2002.
11. 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.
12. M. Sanderson and M. McMahon, *BMC Evol. Biol.* **7**, p. S3 (2007).
13. C. V. Than and L. Nakhleh, *PLoS Comp. Biol.* **5** (2009).
14. 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.
15. J. Yang and T. Warnow, *BMC Bioinf.* **12(Suppl 9)** (2011).
16. L. Zhang, *IEEE/ACM Trans. Comp. Biol. Bioinf.* **8**, 1685 (2011).
17. R. Guigo, I. Muchnik and T. Smith, *Mol. Phylog. Evol.* **6**, 189 (1996).
18. B. Ma, M. Li and L. Zhang, On reconstructing species trees from gene trees in terms of duplications and losses, in *Proc RECOMB*, 1998.
19. P. Gorecki, Reconciliation problems for duplication, loss and horizontal gene transfer, in *Proc RECOMB*, 2004.